NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CLASSIFICATION OF DIGITAL MODULATION SCHEMES USING LINEAR AND NONLINEAR CLASSIFIERS

by

Nathan P. Geisinger

March 2010

Thesis Advisor:	Monique P. Fargues
Thesis Co-Advisors:	Roberto Cristi
	Ralph C. Robertson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2010	**3. REPORT TYPE AND DATES COVERED** Engineer's Thesis	
4. TITLE AND SUBTITLE Classification of Digital Modulation Schemes Using Linear and Nonlinear Classifiers			**5. FUNDING NUMBERS**
6. AUTHOR(S) Nathan P. Geisinger			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The potential benefits of automated detection of digital modulation types have made it a continuing topic of research for many years. Commercial systems could be made more interoperable and military sensors could send demodulated products for analysis, to name just two. Noisy channels and multipath fading environments continue to make this a challenging problem. This thesis applies classification algorithms that have been used in other applications. Nine different digital modulation schemes are considered. The criteria for selecting higher-ordered moments and cumulants as features for discrimination are discussed. An overview of the classification algorithms considered is provided, as well as the statistical models for noisy channels. Results show that the scheme proposed here works well in AWGN channels and in moderate fading conditions.

14. SUBJECT TERMS Blind Modulation Classification, Cumulants, Principal Component Analysis, Linear Discriminant Analysis, Kernel-based functions			**15. NUMBER OF PAGES** 241
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CLASSIFICATION OF DIGITAL MODULATION SCHEMES USING LINEAR AND NONLINEAR CLASSIFIERS

Nathan P. Geisinger
Lieutenant, United States Navy
B.S., Electrical and Computer Engineering, Carnegie Mellon University, 2000

Submitted in partial fulfillment of the
requirements for the degrees of

ELECTRICAL ENGINEER

and

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2010

Author: Nathan P. Geisinger

Approved by: Monique P. Fargues
Thesis Advisor

Roberto Cristi
Thesis Co-Advisor

Ralph C. Robertson
Thesis Co-Advisor

Ralph C. Robertson
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The potential benefits of automated detection of digital modulation types have made it a continuing topic of research for many years. Commercial systems could be made more interoperable and military sensors could send demodulated products for analysis, to name just two. Noisy channels and multipath fading environments continue to make this a challenging problem. This thesis applies classification algorithms that have been used in other applications. Nine different digital modulation schemes are considered. The criteria for selecting higher-ordered moments and cumulants as features for discrimination are discussed. An overview of the classification algorithms considered is provided, as well as the statistical models for noisy channels. Results show that the scheme proposed here works well in AWGN channels and in moderate fading conditions.

.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. OBJECTIVES....1
B. BACKGROUND....1
C. ORGANIZATION....2
D. SIMULATION SOFTWARE....3

II. DIGITAL MODULATION AND WIRELESS CHANNELS....5
A. THE NEED FOR MODULATION....5
B. DIGITAL MODULATION SCHEMES....6
1. Introduction....6
2. Phase Shift Keying....6
3. Quadrature Amplitude Modulation....8
4. Frequency-Shift Keying....9
C. SOURCES OF SIGNAL DEGRADATION....9
1. Introduction....9
2. Additive White Gaussian Noise....9
3. Fading Channels....12
D. CONCLUSION....14

III. SIGNAL FEATURES....15
A. INTRODUCTION....15
B. SPECTRUM BASED CLASSIFICATION OF FSK SCHEMES....15
C. MOMENTS....16
D. CUMULANTS....17
E. FEATURE SELECTION....17
F. CONCLUSION....28

IV. LINEAR AND NONLINEAR CLASSIFIERS....29
A. INTRODUCTION....29
B. LINEAR CLASSIFICATION SCHEMES....29
1. Principal Component Analysis....29
2. Linear Discriminant Analysis....30
C. NONLINEAR CLASSIFICATION SCHEMES....32
1. The Kernel "Trick"....32
2. Nonlinear Component Analysis....33
3. General Discriminant Analysis....33
D. CONCLUSION....34

V. IMPLEMENTATION AND RESULTS....35
A. SIGNAL GENERATION AND CORRUPTION....35
B. CLASSIFIER IMPLEMENTATION....35
C. LINEAR VERSUS NONLINEAR CLASSIFIERS....37
1. PCA....38
2. LDA....39

3. KPCA 40
4. GDA 41
D. PERFORMANCE OF THE PCA AND LDA CLASSIFIERS IN AWGN AND FADING CONDITIONS 42
1. LDA 42
2. PCA 50
VI. CONCLUSIONS 59
APPENDIX A: CUMULANTS EXPRESSED AS FUNCTIONS OF EQUAL AND LOWER ORDER MOMENTS 61
APPENDIX B: BEHAVIOR OF MOMENTS AND CUMULANTS WITH DECREASING SNR 63
A. AWGN ONLY 63
B. AWGN PLUS SLOW, FREQUENCY-FLAT RAYLEIGH FADING 75
C. AWGN PLUS SLOW, FREQUENCY-FLAT RICEAN FADING 88
D. AWGN PLUS SLOW, FREQUENCY-SELECTIVE RAYLEIGH FADING 102
E. AWGN PLUS SLOW, FREQUENCY-SELECTIVE RICEAN FADING 115
F. AWGN PLUS FAST, FREQUENCY-FLAT RAYLEIGH FADING 129
G. AWGN PLUS FAST, FREQUENCY-FLAT RICEAN FADING 142
H. AWGN PLUS FAST, FREQUENCY-SELECTIVE RAYLEIGH FADING 156
I. AWGN PLUS FAST, FREQUENCY-SELECTIVE RICEAN FADING 169
APPENDIX C: EFFECT OF CONSTELLATION ROTATION ON MOMENTS AND CUMULANTS FOR PSK AND QAM SIGNALS 183
A. MOMENTS 183
1. PSK Signals 183
2. QAM Signals 183
B. CUMULANTS 184
1. $C_{x,4,0}$ 184
2. $C_{x,3,1}$ 185
3. $C_{x,2,2}$ 185
4. $C_{x,6,0}$ 185
5. $C_{x,5,1}$ 185
6. $C_{x,4,2}$ 185
7. $C_{x,3,3}$ 186
8. $C_{x,8,0}$ 186
9. $C_{x,7,1}$, $C_{x,6,2}$, $C_{x,5,3}$, and $C_{x,4,4}$ 186
APPENDIX D: MATLAB CODE 187

A. LINEAR CLASSIFIERS 187
1. Description: 187
2. PCA.m 187
3. LDA.m 188
4. LINEARTESTER.m 190
B. NONLINEAR CLASSIFIERS 194
1. Description: 194
2. KPCA.m 195
3. GDA.m 196
4. NONLINTESTER.m 198
C. SUPPORTING FUNCTIONS 203
1. Description 203
2. PROFGEN.m 204
3. TRAINPROFS.m 206
4. KERNEL.m 208
5. MDIST.m 208
LIST OF REFERENCES 211
INITIAL DISTRIBUTION LIST 213

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. QPSK Constellation. 7
Figure 2. 16-QAM Constellation 8
Figure 3. Power Spectral Density of AWGN. 10
Figure 4. Effect of AWGN on 16-QAM (SNR = 20dB). 11
Figure 5. Effect of Rayleigh Fading on 16-QAM (SNR = 20dB). 13
Figure 6. Single-Sided Spectrum of BFSK. 16
Figure 7. $C_{x,4,0}$ (AWGN only). 23
Figure 8. $C_{x,5,1}$ (AWGN only). 24
Figure 9. $C_{x,8,0}$ (AWGN only) 24
Figure 10. $C_{x,4,0}$ (slow, flat Rayleigh fading). 25
Figure 11. $C_{x,5,1}$ (slow, flat Rayleigh fading). 25
Figure 12. $C_{x,8,0}$ (slow, flat Rayleigh fading) 26
Figure 13. $C_{x,4,0}$ (fast, frequency-selective Rayleigh fading). 26
Figure 14. $C_{x,5,1}$ (fast, frequency-selective Rayleigh fading). 27
Figure 15. $C_{x,8,0}$ (fast, frequency-selective Rayleigh fading). 27
Figure 16. Classification Flow Chart for Test Signals 37
Figure 17. $E_{x,2,0}$ in AWGN. 63
Figure 18. $E_{x,1,1}$ in AWGN. 63
Figure 19. $E_{x,4,0}$ in AWGN. 64
Figure 20. $E_{x,3,1}$ in AWGN. 64
Figure 21. $E_{x,2,2}$ in AWGN. 65
Figure 22. $E_{x,6,0}$ in AWGN. 65
Figure 23. $E_{x,5,1}$ in AWGN. 66
Figure 24. $E_{x,4,2}$ in AWGN. 66
Figure 25. $E_{x,3,3}$ in AWGN. 67
Figure 26. $E_{x,8,0}$ in AWGN. 67
Figure 27. $E_{x,7,1}$ in AWGN. 68
Figure 28. $E_{x,6,2}$ in AWGN. 68
Figure 29. $E_{x,5,3}$ in AWGN. 69
Figure 30. $E_{x,4,4}$ in AWGN. 69
Figure 31. $C_{x,4,0}$ in AWGN. 70
Figure 32. $C_{x,3,1}$ in AWGN. 70

Figure 33. $C_{x,2,2}$ in AWGN.....71
Figure 34. $C_{x,6,0}$ in AWGN.....71
Figure 35. $C_{x,5,1}$ in AWGN.....72
Figure 36. $C_{x,4,2}$ in AWGN.....72
Figure 37. $C_{x,3,3}$ in AWGN.....73
Figure 38. $C_{x,8,0}$ in AWGN.....73
Figure 39. $C_{x,7,1}$ in AWGN.....74
Figure 40. $C_{x,6,2}$ in AWGN.....74
Figure 41. $C_{x,5,3}$ in AWGN.....75
Figure 42. $C_{x,4,4}$ in AWGN.....75
Figure 43. $E_{x,2,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....76
Figure 44. $E_{x,1,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....76
Figure 45. $E_{x,4,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....77
Figure 46. $E_{x,3,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....77
Figure 47. $E_{x,2,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....78
Figure 48. $E_{x,6,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....78
Figure 49. $E_{x,5,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....79
Figure 50. $E_{x,4,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....79
Figure 51. $E_{x,3,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....80
Figure 52. $E_{x,8,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....80
Figure 53. $E_{x,7,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....81
Figure 54. $E_{x,6,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....81
Figure 55. $E_{x,5,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....82
Figure 56. $E_{x,4,4}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....82
Figure 57. $C_{x,4,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....83
Figure 58. $C_{x,3,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....83
Figure 59. $C_{x,2,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....84
Figure 60. $C_{x,6,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....84
Figure 61. $C_{x,5,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....85
Figure 62. $C_{x,4,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....85
Figure 63. $C_{x,3,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....86
Figure 64. $C_{x,8,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....86
Figure 65. $C_{x,7,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....87

Figure 66. $C_{x,6,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....87
Figure 67. $C_{x,5,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....88
Figure 68. $C_{x,4,4}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.....88
Figure 69. $E_{x,2,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....89
Figure 70. $E_{x,1,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....90
Figure 71. $E_{x,4,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....90
Figure 72. $E_{x,3,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 91
Figure 73. $E_{x,2,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....91
Figure 74. $E_{x,6,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....92
Figure 75. $E_{x,5,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 92
Figure 76. $E_{x,4,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....93
Figure 77. $E_{x,3,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 93
Figure 78. $E_{x,8,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....94
Figure 79. $E_{x,7,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 94
Figure 80. $E_{x,6,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....95
Figure 81. $E_{x,5,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....95
Figure 82. $E_{x,4,4}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....96
Figure 83. $C_{x,4,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....96
Figure 84. $C_{x,3,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 97
Figure 85. $C_{x,2,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....97
Figure 86. $C_{x,6,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....98
Figure 87. $C_{x,5,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 98
Figure 88. $C_{x,4,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....99
Figure 89. $C_{x,3,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 99
Figure 90. $C_{x,8,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....100
Figure 91. $C_{x,7,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading. 100
Figure 92. $C_{x,6,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....101
Figure 93. $C_{x,5,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....101
Figure 94. $C_{x,4,4}$ in AWGN and Slow, Frequency-Flat Ricean Fading.....102
Figure 95. $E_{x,2,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 103
Figure 96. $E_{x,1,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....103
Figure 97. $E_{x,4,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 104
Figure 98. $E_{x,3,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....104

Figure 99. $E_{x,2,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 105
Figure 100. $E_{x,6,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 105
Figure 101. $E_{x,5,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....106
Figure 102. $E_{x,4,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 106
Figure 103. $E_{x,3,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....107
Figure 104. $E_{x,8,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 107
Figure 105. $E_{x,7,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....108
Figure 106. $E_{x,6,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 108
Figure 107. $E_{x,5,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 109
Figure 108. $E_{x,4,4}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 109
Figure 109. $C_{x,4,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 110
Figure 110. $C_{x,3,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....110
Figure 111. $C_{x,2,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 111
Figure 112. $C_{x,6,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 111
Figure 113. $C_{x,5,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....112
Figure 114. $C_{x,4,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 112
Figure 115. $C_{x,3,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....113
Figure 116. $C_{x,8,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 113
Figure 117. $C_{x,7,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.....114
Figure 118. $C_{x,6,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 114
Figure 119. $C_{x,5,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 115
Figure 120. $C_{x,4,4}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading. 115
Figure 121. $E_{x,2,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....116
Figure 122. $E_{x,1,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading. 117
Figure 123. $E_{x,4,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....117
Figure 124. $E_{x,3,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading. 118
Figure 125. $E_{x,2,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....118
Figure 126. $E_{x,6,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....119
Figure 127. $E_{x,5,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading. 119
Figure 128. $E_{x,4,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....120
Figure 129. $E_{x,3,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading. 120
Figure 130. $E_{x,8,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....121
Figure 131. $E_{x,7,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading. 121

Figure 132. $E_{x,6,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....122
Figure 133. $E_{x,5,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....122
Figure 134. $E_{x,4,4}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....123
Figure 135. $C_{x,4,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....123
Figure 136. $C_{x,3,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....124
Figure 137. $C_{x,2,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....124
Figure 138. $C_{x,6,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....125
Figure 139. $C_{x,5,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....125
Figure 140. $C_{x,4,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....126
Figure 141. $C_{x,3,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....126
Figure 142. $C_{x,8,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....127
Figure 143. $C_{x,7,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....127
Figure 144. $C_{x,6,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....128
Figure 145. $C_{x,5,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....128
Figure 146. $C_{x,4,4}$ in AWGN and Slow, Frequency-Selective Ricean Fading.....129
Figure 147. $E_{x,2,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 130
Figure 148. $E_{x,1,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 130
Figure 149. $E_{x,4,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 131
Figure 150. $E_{x,3,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....131
Figure 151. $E_{x,2,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 132
Figure 152. $E_{x,6,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 132
Figure 153. $E_{x,5,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....133
Figure 154. $E_{x,4,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 133
Figure 155. $E_{x,3,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....134
Figure 156. $E_{x,8,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 134
Figure 157. $E_{x,7,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....135
Figure 158. $E_{x,6,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 135
Figure 159. $E_{x,5,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 136
Figure 160. $E_{x,4,4}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 136
Figure 161. $C_{x,4,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 137
Figure 162. $C_{x,3,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....137
Figure 163. $C_{x,2,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 138
Figure 164. $C_{x,6,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 138

Figure 165. $C_{x,5,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....139
Figure 166. $C_{x,4,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 139
Figure 167. $C_{x,3,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....140
Figure 168. $C_{x,8,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 140
Figure 169. $C_{x,7,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.....141
Figure 170. $C_{x,6,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 141
Figure 171. $C_{x,5,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 142
Figure 172. $C_{x,4,4}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading. 142
Figure 173. $E_{x,2,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....143
Figure 174. $E_{x,1,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 144
Figure 175. $E_{x,4,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....144
Figure 176. $E_{x,3,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 145
Figure 177. $E_{x,2,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....145
Figure 178. $E_{x,6,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....146
Figure 179. $E_{x,5,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 146
Figure 180. $E_{x,4,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....147
Figure 181. $E_{x,3,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 147
Figure 182. $E_{x,8,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....148
Figure 183. $E_{x,7,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 148
Figure 184. $E_{x,6,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....149
Figure 185. $E_{x,5,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....149
Figure 186. $E_{x,4,4}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....150
Figure 187. $C_{x,4,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....150
Figure 188. $C_{x,3,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 151
Figure 189. $C_{x,2,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....151
Figure 190. $C_{x,6,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....152
Figure 191. $C_{x,5,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 152
Figure 192. $C_{x,4,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....153
Figure 193. $C_{x,3,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 153
Figure 194. $C_{x,8,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....154
Figure 195. $C_{x,7,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading. 154
Figure 196. $C_{x,6,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....155
Figure 197. $C_{x,5,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....155

Figure 198. $C_{x,4,4}$ in AWGN and Fast, Frequency-Flat Ricean Fading.....156
Figure 199. $E_{x,2,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....157
Figure 200. $E_{x,1,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 157
Figure 201. $E_{x,4,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....158
Figure 202. $E_{x,3,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 158
Figure 203. $E_{x,2,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....159
Figure 204. $E_{x,6,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....159
Figure 205. $E_{x,5,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 160
Figure 206. $E_{x,4,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....160
Figure 207. $E_{x,3,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 161
Figure 208. $E_{x,8,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....161
Figure 209. $E_{x,7,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 162
Figure 210. $E_{x,6,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....162
Figure 211. $E_{x,5,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....163
Figure 212. $E_{x,4,4}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....163
Figure 213. $C_{x,4,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....164
Figure 214. $C_{x,3,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 164
Figure 215. $C_{x,2,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....165
Figure 216. $C_{x,6,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....165
Figure 217. $C_{x,5,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 166
Figure 218. $C_{x,4,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....166
Figure 219. $C_{x,3,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 167
Figure 220. $C_{x,8,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....167
Figure 221. $C_{x,7,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading. 168
Figure 222. $C_{x,6,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....168
Figure 223. $C_{x,5,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....169
Figure 224. $C_{x,4,4}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.....169
Figure 225. $E_{x,2,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 170
Figure 226. $E_{x,1,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.....170
Figure 227. $E_{x,4,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 171
Figure 228. $E_{x,3,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 171
Figure 229. $E_{x,2,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 172
Figure 230. $E_{x,6,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 172

Figure 231. $E_{x,5,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 173
Figure 232. $E_{x,4,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 173
Figure 233. $E_{x,3,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 174
Figure 234. $E_{x,8,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 174
Figure 235. $E_{x,7,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 175
Figure 236. $E_{x,6,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 175
Figure 237. $E_{x,5,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 176
Figure 238. $E_{x,4,4}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 176
Figure 239. $C_{x,4,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 177
Figure 240. $C_{x,3,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 177
Figure 241. $C_{x,2,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 178
Figure 242. $C_{x,6,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 178
Figure 243. $C_{x,5,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 179
Figure 244. $C_{x,4,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 179
Figure 245. $C_{x,3,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 180
Figure 246. $C_{x,8,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 180
Figure 247. $C_{x,7,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 181
Figure 248. $C_{x,6,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 181
Figure 249. $C_{x,5,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 182
Figure 250. $C_{x,4,4}$ in AWGN and Fast, Frequency-Selective Ricean Fading. 182

LIST OF TABLES

Table 1. Moments of Digitally Modulated Signals (SNR=∞). 19
Table 2. Cumulants of Digitally Modulated Signals (SNR=∞). 20
Table 3. Normalized Cumulants of Digitally Modulated Signals (SNR=∞). 21
Table 4. Confusion Matrix for PCA classifier in AWGN, SNR = 20 dB. 38
Table 5. Confusion Matrix for PCA classifier in AWGN, SNR = 5 dB. 39
Table 6. Confusion Matrix for LDA classifier in AWGN, SNR = 20 dB. 39
Table 7. Confusion Matrix for LDA classifier in AWGN, SNR = 5 dB. 40
Table 8. Confusion matrix for KPCA classifier in AWGN, SNR = 20 dB. 40
Table 9. Confusion matrix for KPCA classifier in AWGN, SNR = 5 dB. 41
Table 10. Confusion Matrix for GDA classifier in AWGN, SNR = 20 dB. 41
Table 11. Confusion Matrix for GDA classifier in AWGN, SNR = 5 dB. 42
Table 12. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 20 dB). 43
Table 13. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 5 dB). 43
Table 14. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 20 dB). 44
Table 15. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 5 dB). 44
Table 16. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 20 dB). 45
Table 17. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 5 dB). 45
Table 18. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 20 dB). 46
Table 19. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 5 dB). 46
Table 20. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 20 dB). 47
Table 21. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 5 dB). 47
Table 22. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 20 dB). 48
Table 23. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 5 dB). 48
Table 24. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 20 dB). 49
Table 25. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 5 dB). 49
Table 26. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 20 dB). 50

Table 27. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 5 dB). 50
Table 28. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 20 dB). 51
Table 29. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 5 dB). 51
Table 30. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 20 dB).....52
Table 31. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 5 dB).....52
Table 32. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 20 dB).....53
Table 33. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 5 dB).....53
Table 34. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 20 dB).....54
Table 35. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 5 dB).....54
Table 36. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 20 dB).....55
Table 37. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 5 dB).....55
Table 38. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 20 dB). 56
Table 39. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 5 dB). 56
Table 40. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 20 dB).....57
Table 41. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 5 dB).....57
Table 42. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 20 dB). 58
Table 43. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 5 dB). 58
Table 44. Cumulants Expressed In Terms of Moments.....61

EXECUTIVE SUMMARY

Blind modulation classification remains a challenging problem despite the numerous studies that have investigated it. This thesis applies higher-order moments and cumulants as features to several modulation classification algorithms.

An overview of digital communications is provided along with the statistical models for noisy and fading channels. The extraction of moments and cumulants is discussed at length, and the criteria for selecting them as features for discrimination are explained. Attempts are made to make the classifers robust to fading effects, first by investigating the effects of phase shifts on the statistics, and second by applying realistic power normalizations. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and their nonlinear, kernel-based counterparts are presented.

Principal Component Analysis works slightly better than Linear Discriminant Analysis in most channel conditions and equals the performance of kernel-based PCA with less computational overhead. Results show that this classifier works very well with signals that are corrupted only by additive white Gaussian noise (AWGN). It achieves a classification rate of 95.1% at an SNR of 20 dB and 89.0% at 5 dB. In moderate fading conditions, such as might be experienced by a mobile device being carried by a person walking, it performs nearly as well, with classification rates in a Rayleigh channel of 87.0% (20 dB) and 86.8% (5 dB), and 91.5% (20 dB) and 91.7% (5 dB) in a Ricean channel. Even PCA's performance degrades sharply, however, in more severe fading conditions.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
FFT	Fast Fourier Transform
FSK	Frequency Shift Keying
GDA	General Discriminant Analysis
KPCA	Kernel Principal Component Analysis
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
PDF	Probability Density Function
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SNR	Signal-to-Noise Ratio

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my parents for all the moral support that they provided in the course of this research, and I would especially like to thank my thesis advisor, Monique Fargues, who always had one more suggestion whenever I seemed to have hit a wall.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OBJECTIVES

Blind modulation classification in which the modulation type of a waveform is determined without a priori information has many potential applications. In the commercial sector, it would enable the interoperability of diverse communications devices. In the military domain, it would allow the demodulation of an intercepted signal to be performed in situ rather than after the signal has been sampled and forwarded on for processing. Software-defined radio would also benefit from this capability since the transmitter and receiver would not have to agree on a particular modulation scheme in advance.

Because of the obvious benefits of blind modulation classification, much has been published on this topic in the literature in recent years, as will be seen below. There is no widespread agreement, however, on which modulation schemes to consider. Even within the main families of schemes, there are endless variants. Some papers consider phase-shift keying (PSK) versus frequency shift keying (FSK). Others introduce quadrature amplitude modulation (QAM) or variants of PSK such as minimum-shift keying. This thesis is no different in its arbitrariness. Nine different modulation schemes are considered: 2, 4, and 8-PSK, 2, 4, and 8-FSK, and 16, 64, and 256-QAM. How the features used for classification were chosen and the classifier algorithms are discussed. Simulation results for signals in additive white Gaussian noise (AWGN) and signals in fading channels are also discussed.

B. BACKGROUND

Various methods have been proposed to accomplish blind modulation classification. Marchand proposes the use of higher-ordered moments and cumulants in [1]. In one way or another, these features are used in [2, 3, 4]. Hatzichristos uses them as inputs to a neural network classifier [2], whereas Young uses a simpler approach

involving threshold values [3]. Like, Chakravarthy, Ratazzi, and Wu propose a neural network-based classifier that uses cyclostationarity detection in its first tier and 4^{th} and 8^{th}-ordered cumulants thereafter [4].

Other methods involve the use of wavelet transforms. Ho, Prokopiw, and Chan propose a scheme that uses very few symbols (on the order of 100) but do not consider QAM, in which both the amplitude and the phase are modulated [5, 6]. Prakasam and Madheswaran extracts several statistics from the wavelet transform and places them in an elaborate decision tree [7].

A summary of many of these methods can be found in [8], in which four broad categories for all the methods to date are defined: algorithms based on instantaneous amplitude, phase, and frequency; algorithms based on the wavelet transform; algorithms based on cumulants; and algorithms based on cyclostationarity properties.

This thesis falls into the third category. Some of the same features – higher-ordered moments and cumulants - investigated in [2, 3] are examined, but a different set of classification algorithms is investigated. Specifically, the classification rates of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), as well as their nonlinear (kernel-based) counterparts, are determined and compared. These schemes are well-known and have been applied successfully to face-recognition applications in [9, 10].

C. ORGANIZATION

The theory of digital communications systems, the nine modulation schemes considered in this work, and the mathematical models for introducing noise to wireless channels are reviewed in Chapter II. The features used for classification, namely moments and cumulants, and the reasons for choosing the particular moments and cumulants used in this work, are discussed in Chapter III. The linear and nonlinear classification schemes that use the moments and cumulants as inputs are introduced in Chapter IV. The software implementation of the classifiers and the simulation results are discussed in Chapter V. Finally, conclusions and suggestions for further work are presented in Chapter VI.

D. SIMULATION SOFTWARE

All simulations were conducted in MATLAB version 7.6. Various functions from the Communications Toolbox were used. The code developed for this thesis is listed in the appendices. Note that this work implicitly assumes that some preprocessing has already occurred to down-convert the received signal as the functions in the Communications Toolbox used in this work simulate digital signals at baseband.

THIS PAGE INTENTIONALLY LEFT BLANK

II. DIGITAL MODULATION AND WIRELESS CHANNELS

The reader is introduced to the digital modulation schemes considered in this thesis in this chapter. Additionally, the mathematical models used to represent modulated signals corrupted during transmission are presented.

A. THE NEED FOR MODULATION

The purpose of any digital communication system is to transmit data from an information source to an information sink. At a minimum, a baseband waveform must be constructed from the symbols making up the information. This often amounts to assigning two different voltage levels to the binary symbols that represent the information. Construction of this baseband waveform is called pulse code modulation.

Some media, such as coaxial cable, can propagate baseband waveforms with no further manipulation required. For wireless transmission, on the other hand, baseband waveforms are impractical for several reasons [11]. First, the antenna size necessary to transmit an electromagnetic wave is inversely proportional to the wave's frequency. Therefore, shifting the baseband waveform to a higher frequency allows for a smaller antenna. For many mobile devices, this results in a carrier frequency on the order of 1 GHz.

Second, multiple transmitters operating at the same frequency generally interfere with each other. By assigning each transmitter a unique portion of the electromagnetic spectrum, multiple transmitters can operate without interference. This is called frequency-division multiple access.

The translation of a baseband waveform to a higher frequency is accomplished by manipulating the features of a sinusoidal wave at that frequency. This process is called bandpass modulation.

B. DIGITAL MODULATION SCHEMES

1. Introduction

There are three features of a sinusoidal wave that the designer can manipulate: amplitude, frequency, and phase. A given modulation scheme works on one or more of these. This work will consider three fundamental schemes: PSK, FSK and QAM.

2. Phase Shift Keying

As the name implies, PSK works by assigning a unique phase to each symbol present in the baseband waveform. Thus, binary phase-shift keying (BPSK) in which each symbol represents one bit uses two different phases, 4-PSK uses four different phases, and so on. The resulting waveforms can be represented mathematically by [12]:

$$s_i(t) = \left\{ \sqrt{\frac{2E_S}{T_S}} \cos\left(2\pi f_c t + \frac{2\pi}{M} i \right) \quad i = 0,1,...,M-1, \right. \tag{2.1}$$

where M is the number of bits per symbol, f_c is the carrier frequency, E_s is the energy per symbol, and T_s is the symbol interval.

The case where $M = 4$ illustrates a result that is widely exploited in communications systems: orthogonal signaling. The four phases are $0, \pi/2, \pi$, and $3\pi/2$. Note, however, that

$$\cos\left(2\pi f_c t + \frac{\pi}{2} \right) = \sin\left(2\pi f_c t \right) \text{ and } \cos\left(2\pi f_c t + \frac{3\pi}{2} \right) = -\sin\left(2\pi f_c t \right).$$

This is important because the sine and cosine waves are orthogonal signals, meaning that they are uncorrelated in time over a symbol duration. Even when their frequencies are the same, they can be transmitted by the same antenna and recovered by the same receiver without mutual interference. When two signals are orthogonal, it is common to represent them as two perpendicular vectors that form a basis in R^2. This leads to a conceptualization of the modulated symbols as points in a constellation, as shown in Figure 1.

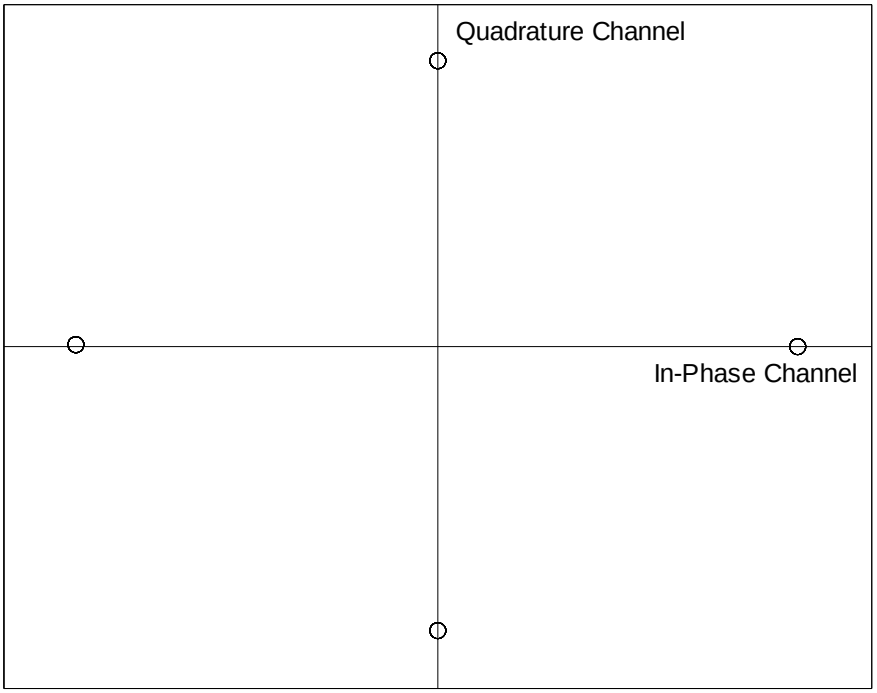

Figure 1. QPSK Constellation.

Thus, any modulated waveform can be constructed as the sum of two sinusoids. The first, represented by the horizontal axis, is called the in-phase (I) channel. The second, represented by the vertical axis, is exactly out of phase with the first and is called the quadrature (Q) channel. For this reason, 4-PSK is usually referred to as quadrature phase-shift keying (QPSK). This result will be particularly important in the discussion of QAM.

PSK schemes are popular for their symbol error performance and bandwidth efficiency. This comes at the price, however, of increased receiver complexity. In order to recover the baseband signal, the receiver must generate a sinusoid that is phase-synchronized with the received signal (except for one variant called differential phase-shift keying, which is not considered in this work). Modulation schemes that impose this requirement on the receiver are said to be coherently detected.

PSK signals sometimes have an additional phase offset that rotates the constellation by a uniform amount, typically $\pi/4$ for BPSK and $\pi/8$ for QPSK. As will be seen in Chapter III, this can affect the features used for classification.

3. Quadrature Amplitude Modulation

Whereas PSK only uses phase differences to distinguish between symbols, QAM modulates both the phase and the amplitude of the carrier. Thus, it can be thought of as a generalization of PSK where all symbols need not have equal energy. Like PSK, the mathematical expression for an *M*-ary QAM signal can be decomposed into orthogonal sinusoids [12]:

$$s_i(t) = \sqrt{\frac{2E_0}{T_s}} a_i \cos(2\pi f_c t) - \sqrt{\frac{2E_0}{T_s}} b_i \sin(2\pi f_c t) \quad i = 0,1,...,M-1. \tag{2.2}$$

Figure 2 shows the symbol constellation obtained for a 16-QAM signal.

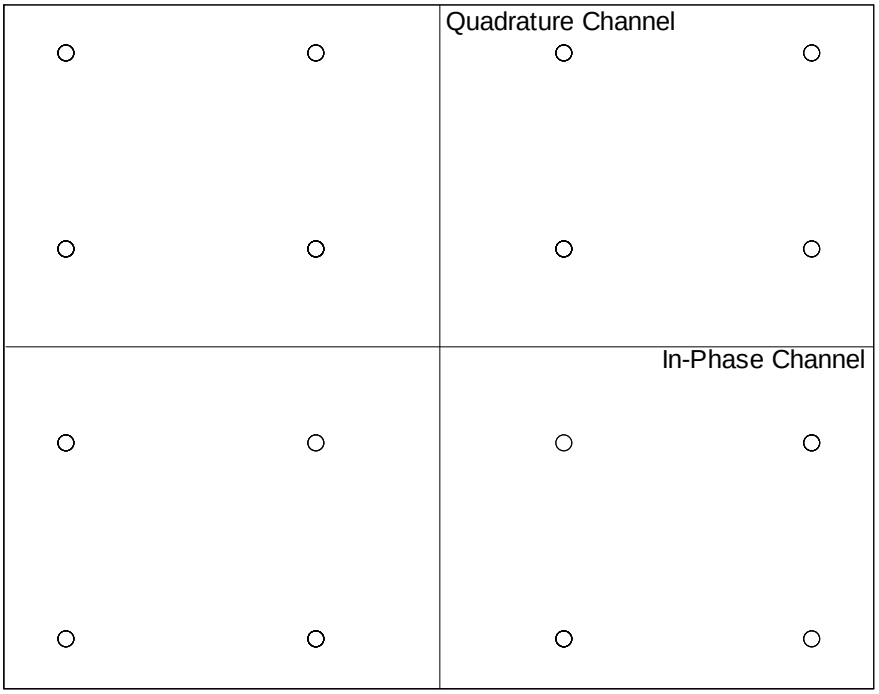

Figure 2. 16-QAM Constellation.

Although this work only considers QAM schemes that have square-shaped constellations (i.e., an equal number of symbols occupying each quadrant), this is not a general requirement for QAM. QAM signals tend to be bandwidth efficient and, like PSK, must be coherently detected.

4. Frequency-Shift Keying

FSK signals are unlike PSK and QAM in that each of the *M* symbols in an *M*-ary scheme has its own signaling frequency [12]:

$$s_i(t) = \sqrt{\frac{2E_s}{T_s}} \cos(2\pi f_i t) \quad i = 1, 2, ..., M. \tag{2.3}$$

FSK uses bandwidth inefficiently but can be detected noncoherently, which reduces receiver complexity. It cannot be decomposed and visualized as I- and Q-channels, since it uses multiple signaling frequencies.

C. SOURCES OF SIGNAL DEGRADATION

1. Introduction

In the process of transmission and reception, there are several ways in which a signal can undergo degradation, making it harder for the receiver to demodulate it correctly—or making it harder to determine its modulation type. A radio frequency (RF) signal always experiences a decrease in power due to the distance from the transmitter, called path loss. In addition, this work considers two types of degradation: AWGN and multipath fading.

2. Additive White Gaussian Noise

Any receiver will have a certain amount of thermal energy associated with its antenna and other circuitry. Additionally, interfering signals will be received by the antenna [11]. This interference can be modeled as a Gaussian random process that adds to the modulated signal. For a signal propagating in free space, this model is sufficient to

describe the degradation of the signal. It is common to express this process in terms of its power spectral density, $N_0/2$, which is a constant as a function of frequency.

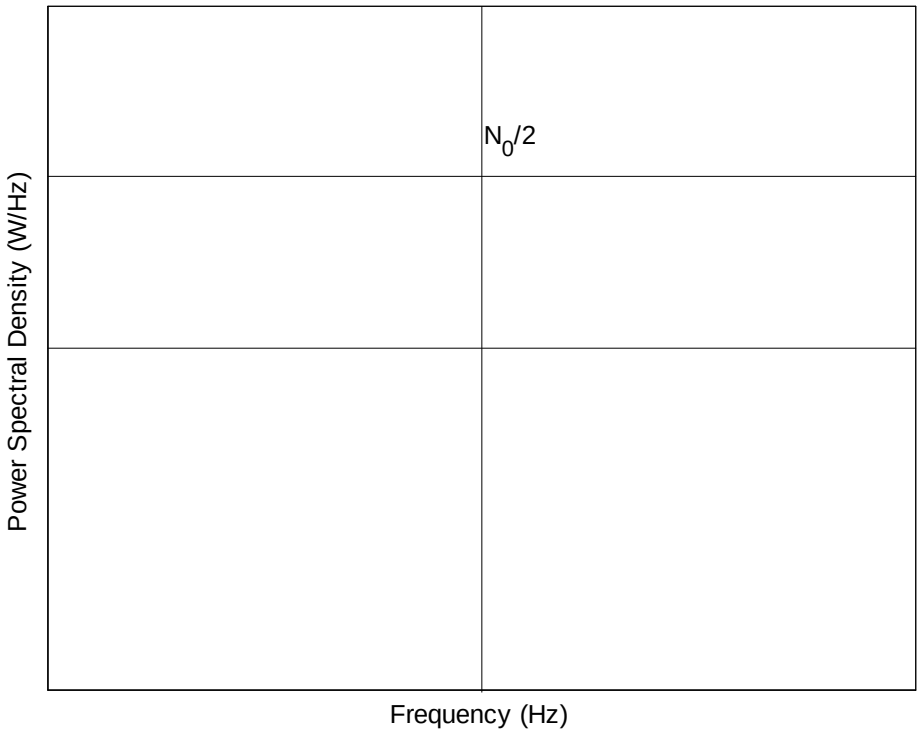

Figure 3. Power Spectral Density of AWGN.

The effect of AWGN on a signal is to introduce uncertainty as to which symbols are being transmitted. The variance in the thermal noise will shift each symbol around its intended value. For example, the constellation of a 16-QAM signal with 10,000 symbols in the presence of AWGN is shown in Figure 4.

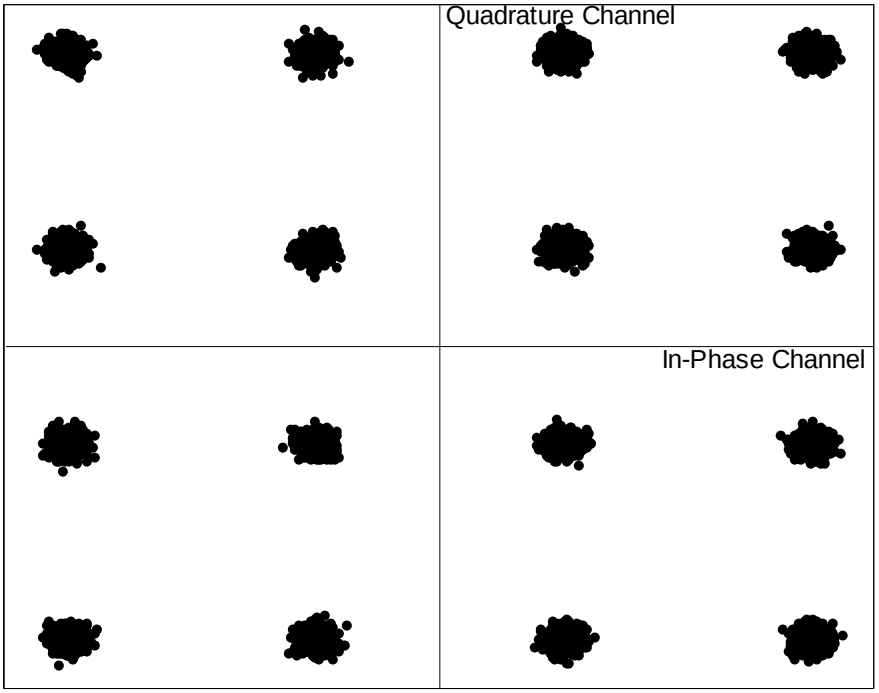

Figure 4. Effect of AWGN on 16-QAM (SNR = 20dB).

This degradation can be quantified by defining the signal-to-noise ratio (SNR), which is simply the ratio of the signal power to the noise power. For the purpose of error analysis, it is necessary to modify this quantity to account for symbol time and receiver bandwidth. This results in the quantity E_s/N_0, which denotes the ratio of symbol energy-to-noise energy. Note that much of the literature in communications theory expresses SNR as E_b/N_0, the ratio of bit energy to noise energy. This is because the usual measure of error performance is bit error ratio, and expressing it in terms of E_s/N_0 allows for an apples-to-apples comparison between *M*-ary systems. In order to keep the bit energy consistent, however, an *M*-ary system must have *M* times as much symbol energy as a binary system ($E_s = ME_b$), and, therefore, *M* times as much power, assuming equal

symbol rates. Because the goal of this work is to distinguish between modulation types, the symbol energy is kept consistent and E_s/N_0 is used as the measure of SNR.

3. Fading Channels

While AWGN alone adequately models the degradation of a signal in free space, many signals undergo more complicated propagation on top of AWGN. The presence of physical obstructions such as buildings causes multipath propagation losses. As the name implies, this means that the signal arrives at the receiver via more than one propagation path. Depending on the relative lengths of the paths, this can result in either constructive or destructive interference. Furthermore, relative motion between the transmitter and receiver causes Doppler shift.

Multipath propagation results in both large scale and small scale fading. Large scale fading is similar to the path loss that a signal experiences in free space in that it is proportional to $1/r^n$, where r is the distance between the transmitter and receiver. In free space $n = 2$, whereas in a fading channel, n depends on the particular objects reflecting the signal. Also, this relationship only determines the mean path loss due to large scale fading. The actual path loss can be modeled as a random variable with a log-normal variation around this mean value [11].

Small scale fading refers to finer variations in the relative position of the transmitter and receiver that can cause multipath copies of the signal to interfere with each other. When a line-of-sight component is present among the reflected paths, the received amplitude can be modeled as a random variable with a Ricean probability density function (PDF) [11],

$$p(r_0) = \frac{r_0}{\sigma^2} \exp\left[-\frac{r_0^2 + A^2}{2\sigma^2} \right] I_0\left(\frac{r_0 A}{\sigma^2} \right), \tag{2.4}$$

where r_0 is the distance between transmitter and receiver, A is the magnitude of the line-of-sight component, σ is the mean amplitude of the reflected components, and I_0 is a modified Bessel function of the first kind. When no line-of-sight component is available, this PDF simplifies to the Rayleigh PDF:

$$p(r_0) = \frac{r_0}{\sigma^2} \exp\left[-\frac{r_0^2}{2\sigma^2} \right]. \tag{2.5}$$

Finally, in mobile applications the Doppler effect causes the multiple received copies of the signal to alternate between constructive and destructive interference. When this effect is slow compared to a symbol interval, it is called slow fading. Similarly, when the effect occurs within a symbol interval, it is called fast fading [13].

Fortunately, the MATLAB Communications Toolbox includes functions that account for all these effects. A 16-QAM signal in slow Rayleigh fading is shown in Figure 5. The overall effect is to rotate the constellation.

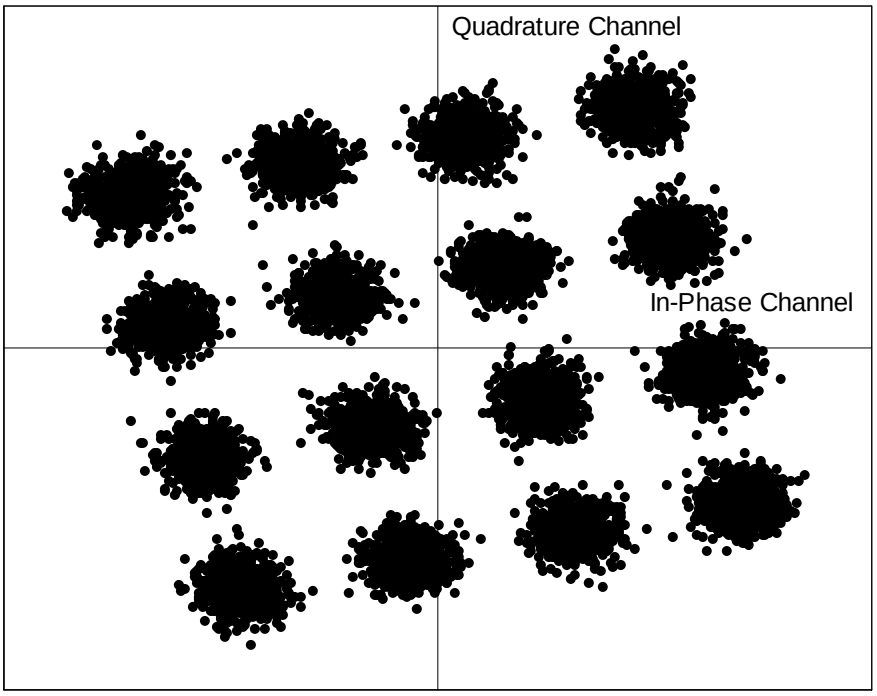

Figure 5. Effect of Rayleigh Fading on 16-QAM (SNR = 20dB).

D. CONCLUSION

In this chapter, the reader was introduced to some of the digital signaling techniques commonly in use and the statistical methods that model their corruption by white noise and multipath fading, which both make the task of classifying the signals harder. The features used as a basis for discrimination will be discussed in Chapter III.

III. SIGNAL FEATURES

In order to implement the classifier, features must be extracted from the signals to provide a basis for distinguishing between the modulation types. In this chapter, the features that will be used in classification are presented and their behavior under AWGN and multipath fading examined.

A. INTRODUCTION

For FSK signals, the most effective approach is to exploit their unique spectral characteristics. For PSK and QAM, a number of different discriminating features have been proposed in the literature. The use of higher ordered moments and cumulants was proposed in [14] and explored in [2, 3]. Wavelet transforms were examined in [5, 6, 7]. More recently, [4] presented an approach based on cyclostationarity concepts.

B. SPECTRUM BASED CLASSIFICATION OF FSK SCHEMES

The spectrum of an FSK signal is sufficient to distinguish it from PSK and QAM signals and to separate different M-ary FSK schemes. A simple way to automate this examination of the spectrum is presented in [15]. First, the fast Fourier transform (FFT) of the signal is computed. Next, transform peak values are compared to each other in order to determine whether the signal is FSK and if so, what type.

Consider the single-sided spectrum of the BFSK signal shown in Figure 6. There are clearly two peaks in this spectrum. If the ratio of the second highest value to the third highest value in the spectrum is computed, the result is a number larger than one. Likewise, a 4-FSK signal will have a large ratio of the fourth to the fifth highest value and an 8-FSK signal of the eighth to the ninth highest value. For non-FSK signals, these ratios will all be approximately unity.

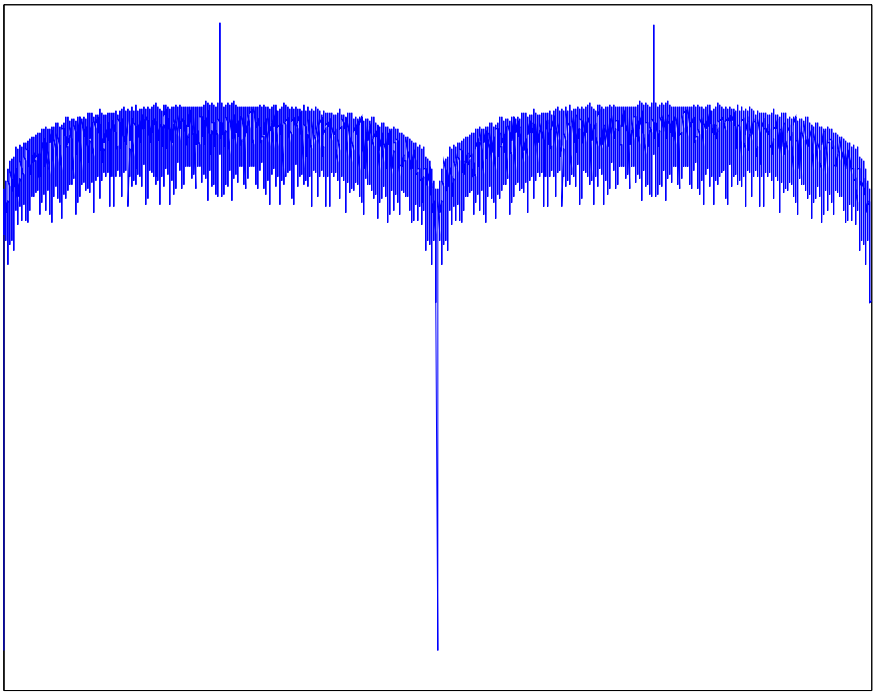

Figure 6. Single-Sided Spectrum of BFSK.

C. MOMENTS

Statistical moments are nothing more than the expected value of a random variable raised to the power indicated by the order of the moment. A first order moment is the mean of the variable. In many applications, the second order moment is a measurement of power. For complex random variables, a conjugate term is often included (note that the signals generated in this work are simulated by their complex envelope). In general, the moment of a random variable is defined as

$$E_{x,a,b} = E\left[x^a (\overline{x})^b \right], \quad a,b \in \mathbb{Z}, \tag{3.1}$$

where x is the variable and $\overline{x}$ is its complex conjugate. The order of this moment is $a+b$. For our purposes, x is a vector of samples of a digitally modulated signal. Rather than explicitly calculating statistical moments, they are estimated by raising each element in the vector to the indicated power and computing the numerical mean. In this work the

moments are normalized by the measure of signal power appropriate for the order of the moment. For example, a second order moment is normalized by the signal power itself, a fourth order moment is normalized by the power squared, and so on. Additionally, all moments are central moments as the signal mean is zero.

D. CUMULANTS

In order to define the cumulants of a random variable x, we first define its characteristic function and second characteristic function, respectively, as [16]:

$$\Phi(s) = \int_{\infty}^{-\infty} f(x) e^{sx} dx \tag{3.2}$$

and

$$\Psi(s) = \ln \Phi(s). \tag{3.3}$$

The n^{th}-order cumulant of x is defined as the n^{th} derivative of the second characteristic function evaluated at $s = 0$:

$$\lambda_n = \frac{d^n \Psi(0)}{ds^n}. \tag{3.4}$$

Note, that x may be complex-valued. In order to be consistent with the notation above for moments, cumulants will be represented by $C_{x,a,b}$.

Fortunately, cumulants can also be expressed as functions of equal and lower ordered moments, making it simple to compute them numerically. These expressions were computed up to the eighth order in [1] and are presented in Appendix A.

E. FEATURE SELECTION

Some points should be made about the application of moments and cumulants to the problem at hand. First, they will have imaginary components. It was asserted in [1] and [2] that the imaginary components would be zero for the signals considered here. This may be the case if there is no rotation of the signal constellation. As mentioned in Chapter II, however, fading channels have the effect of rotating the entire constellation; also, some implementations of BPSK and QPSK may include an arbitrary phase offset.

To illustrate the effect of a phase rotation, $E_{x,4,0}$ is calculated below for a QPSK signal with an arbitrary phase shift of ϕ. The expression used is the baseband (or complex-envelope) expression for QPSK. Since QPSK signals have four possible values, $d(k)$ is a uniformly distributed random integer between 0 and 3.

$$\begin{aligned} E\left[\left(\exp\left(j\left(\frac{2\pi}{4}d(k)+\phi\right)\right)\right)^4\right] &= E\left[\exp\left(j\left(\frac{8\pi}{4}d(k)+4\phi\right)\right)\right] \\ &= \exp(j4\phi)\sum_{k=0}^{3}\frac{1}{4}\exp(j2\pi k) \\ &= \exp(j4\phi)\frac{1}{4}[1+1+1+1] \\ &= \exp(j4\phi). \end{aligned} \tag{3.5}$$

Clearly, the phase rotation may introduce an imaginary component into the statistic that varies with the phase shift ϕ. Section A of Appendix C contains a general proof that, for PSK and QAM signals, an arbitrary phase shift of the constellation affects the phase of the moments but not their magnitudes. In section B, this result is shown to hold for the cumulants $C_{x,2,0}, C_{x,1,1}, C_{x,4,0}, C_{x,3,1}, C_{x,6,0}, C_{x,5,1}$, and $C_{x,8,0}$. Therefore, the magnitude of each of these statistics are used rather than the real component alone. In Tables 1 and 2 the magnitudes of the moments and cumulants computed for all nine modulation types using uncorrupted signals of 20,000 symbols each are presented. These were estimated by simulating the signals in MATLAB. Note that they have been normalized by the estimated received signal power.

The main purpose of presenting the moments and cumulants here is to verify that they are being calculated correctly, but it should already be clear that some of them will be more useful than others. For example, $C_{x,2,0}$ is useful for identifying BPSK, but confuses all the other modulation types. In contrast, $C_{x,4,0}$ and $C_{x,5,1}$ have different values for all PSK and QAM schemes, although the 64- and 256-QAM values are very close together. As will be seen, adjustments are necessary in order to make them more robust to noisy conditions. Accordingly, their behavior in noisy conditions is investigated before deciding which ones to use as classification features.

Table 1. Moments of Digitally Modulated Signals (SNR=∞).

	BPSK	**QPSK**	**8PSK**	**BFSK**	**4FSK**	**8FSK**	**16QAM**	**64QAM**	**256QAM**
$E_{x,2,0}$	1.000	0.006	0.006	0.000	0.000	0.000	0.008	0.007	0.008
$E_{x,1,1}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$E_{x,4,0}$	1.000	1.000	0.006	0.000	0.000	0.000	0.681	0.618	0.606
$E_{x,3,1}$	1.000	0.006	0.006	0.000	0.000	0.000	0.011	0.012	0.013
$E_{x,2,2}$	1.000	1.000	1.000	1.000	1.000	1.000	1.320	1.381	1.396
$E_{x,6,0}$	1.000	0.006	0.006	0.000	0.000	0.000	0.018	0.025	0.025
$E_{x,5,1}$	1.000	1.000	0.006	0.000	0.000	0.000	1.321	1.296	1.291
$E_{x,4,2}$	1.000	0.006	0.006	0.000	0.000	0.000	0.019	0.022	0.025
$E_{x,3,3}$	1.000	1.000	1.000	1.000	1.000	1.000	1.960	2.225	2.293
$E_{x,8,0}$	1.001	1.000	1.000	0.000	0.000	0.000	2.204	1.907	1.828
$E_{x,7,1}$	1.001	0.006	0.006	0.000	0.000	0.000	0.031	0.050	0.054
$E_{x,6,2}$	1.001	1.000	0.006	0.000	0.000	0.000	2.485	2.755	2.815
$E_{x,5,3}$	1.001	0.006	0.006	0.000	0.000	0.000	0.032	0.047	0.055
$E_{x,4,4}$	1.001	1.001	1.000	1.000	1.000	1.000	3.124	3.961	4.194

Table 2. Cumulants of Digitally Modulated Signals (SNR=∞).

	BPSK	**QPSK**	**8PSK**	**BFSK**	**4FSK**	**8FSK**	**16QAM**	**64QAM**	**256QAM**
$C_{x,2,0}$	1.000	0.006	0.006	0.000	0.000	0.000	0.008	0.007	0.008
$C_{x,1,1}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$C_{x,4,0}$	2.000	1.000	0.006	0.000	0.000	0.000	0.681	0.618	0.606
$C_{x,3,1}$	2.000	0.012	0.012	0.000	0.000	0.000	0.012	0.011	0.011
$C_{x,2,2}$	2.000	1.000	1.000	1.000	1.000	1.000	0.680	0.619	0.604
$C_{x,6,0}$	15.999	0.087	0.006	0.000	0.000	0.000	0.068	0.054	0.052
$C_{x,5,1}$	15.999	3.998	0.024	0.000	0.000	0.000	2.084	1.794	1.738
$C_{x,4,2}$	15.999	0.062	0.065	0.000	0.000	0.000	0.060	0.050	0.048
$C_{x,3,3}$	15.999	4.000	4.000	4.000	4.000	4.000	2.080	1.798	1.733
$C_{x,8,0}$	243.962	33.971	0.998	0.000	0.000	0.000	14.023	11.465	11.009
$C_{x,7,1}$	243.957	1.087	0.012	0.000	0.000	0.000	0.870	0.715	0.737
$C_{x,6,2}$	243.962	45.984	0.280	0.000	0.000	0.000	29.864	27.043	26.478
$C_{x,5,3}$	243.962	0.396	0.528	0.000	0.000	0.000	0.273	0.328	0.345
$C_{x,4,4}$	243.962	17.999	16.992	17.000	17.000	17.000	17.371	24.098	25.728

One potential problem in using these statistics as they are is that the magnitude of the cumulants increases with their order. This characteristic could have the unintended consequence of weighting these larger statistics more heavily in the classification

scheme. In order to mitigate this effect, [17] proposed raising each cumulant to the power $2/n$, where n is the cumulant's order. A revised table of these normalized cumulants is presented in Table 3.

Table 3. Normalized Cumulants of Digitally Modulated Signals (SNR=∞).

	BPSK	**QPSK**	**8PSK**	**BFSK**	**4FSK**	**8FSK**	**16QAM**	**64QAM**	**256QAM**
$C_{x,2,0}$	1.000	0.007	0.006	0.000	0.000	0.000	0.007	0.007	0.008
$C_{x,1,1}$	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
$C_{x,4,0}$	1.414	1.000	0.070	0.000	0.000	0.000	0.824	0.787	0.778
$C_{x,3,1}$	1.414	0.107	0.103	0.000	0.000	0.000	0.098	0.100	0.099
$C_{x,2,2}$	1.414	1.000	1.000	1.000	1.000	1.000	0.824	0.786	0.777
$C_{x,6,0}$	2.520	0.424	0.173	0.004	0.004	0.004	0.368	0.369	0.372
$C_{x,5,1}$	2.520	1.587	0.265	0.006	0.006	0.006	1.276	1.216	1.201
$C_{x,4,2}$	2.520	0.379	0.384	0.006	0.006	0.007	0.351	0.356	0.350
$C_{x,3,3}$	2.520	1.587	1.587	1.587	1.587	1.587	1.276	1.215	1.200
$C_{x,8,0}$	3.952	2.414	0.999	0.018	0.018	0.018	1.933	1.842	1.820
$C_{x,7,1}$	3.952	0.982	0.313	0.018	0.018	0.018	0.895	0.897	0.905
$C_{x,6,2}$	3.952	2.604	0.676	0.040	0.039	0.040	2.337	2.282	2.268
$C_{x,5,3}$	3.952	0.763	0.818	0.036	0.036	0.037	0.693	0.719	0.768
$C_{x,4,4}$	3.952	2.060	2.030	2.031	2.031	2.031	2.042	2.218	2.254

Simulations showed that taking the magnitude of the statistics and normalizing the cumulants according to their order greatly improved discrimination power of the features considered when dealing with AWGN. However, simulations also showed that some of the statistics were very sensitive to the received signal power. In addition, the fading process affects the power of signals as well as their phase. A simple assumption will improve the classifier's performance in AWGN and allow it to deal better with faded signals. Suppose that a measurement of received power is available in the absence of any signals. This measurement will equal the power added to the signals by AWGN. Subtracting this from the received signal power will allow us to normalize the statistics by the "noise-free" power.

As stated previously, all FSK signals have nearly the same moments and cumulants and are handled by measuring their spectral characteristics. For the remaining six modulation types, the number of features clearly must be reduced in order to simplify the problem for the classifier. Thus, the challenge is to identify a smaller set of moments and/or cumulants that can distinguish between the modulation schemes and that are relatively invariant over a range of SNRs.

In order to accomplish this task, MATLAB was used to generate and corrupt the modulated signals with AWGN and then to extract their moments and cumulants for even orders from two to eight. These results are plotted for each statistic in Section A of Appendix B. Sections B through I show the same statistics under Rayleigh and Ricean fading conditions for all combinations of slow or fast and frequency-flat or frequency-selective channels. The exact parameters used in MATLAB are also presented.

Sections B and C show that the moments and cumulants do not change much under slow, frequency-flat fading conditions for either Rayleigh or Ricean channels, indicating that the classifier should work well in these regimes. In sections D and E, the frequency-selective nature of the channels begins to shift the values, although some of them are still stable with increasing AWGN.

In sections F and G, the values have shifted significantly due to fast fading. Furthermore, they are less stable with increasing AWGN. Since these results are for

frequency-flat channels, we can conclude that Doppler shift has a more significant effect on the statistics than path delay. In sections H and I, which show the effects of fast, frequency-selective fading, the statistics are somewhat more stable but still well off their original values.

Three statistics were selected as classification features: $C_{x,4,0}$, $C_{x,5,1}$, and $C_{x,8,0}$. All of them are invariant to phase shifts and provide good separation between the PSK schemes and some separation between the QAM schemes. Their plots are shown in Figures 7 to 9. While some of the moments (such as $E_{x,4,0}$ and $E_{x,8,0}$) appeared to meet the same criteria, including them tended to make the classifier perform worse with faded signals. The values for 64- and 256-QAM do not differ until the second or third decimal place, indicating that they will present a challenge for the classifier.

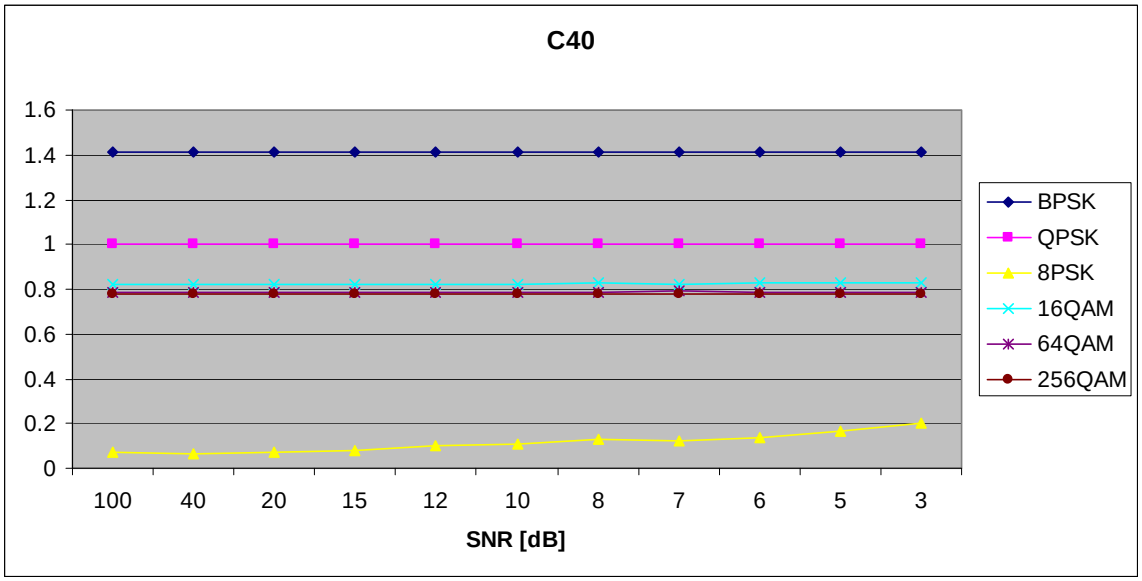

Figure 7. $C_{x,4,0}$ (AWGN only).

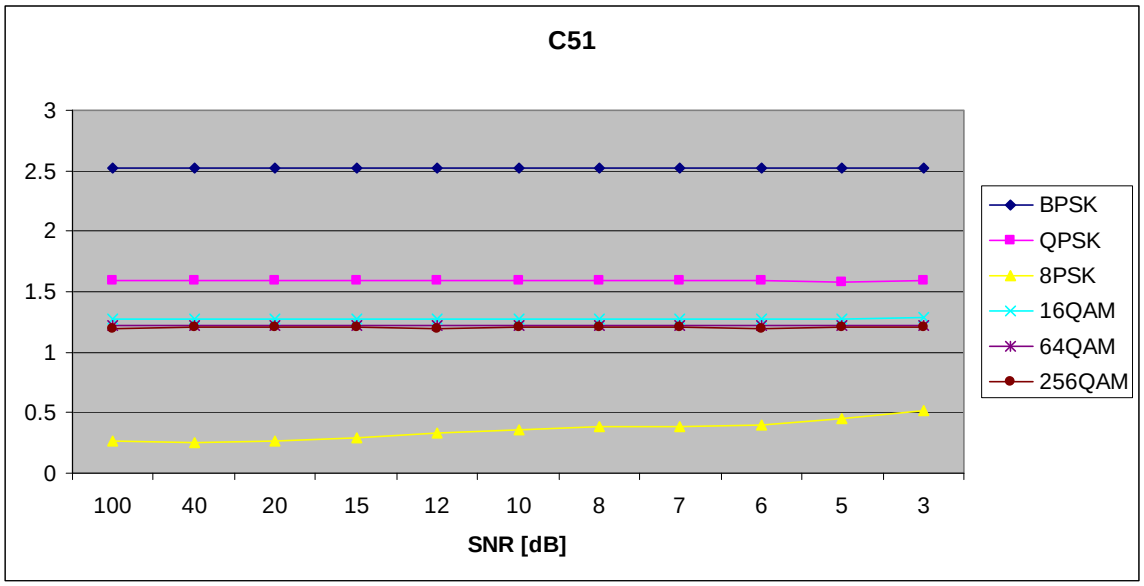

Figure 8. $C_{x,5,1}$ (AWGN only).

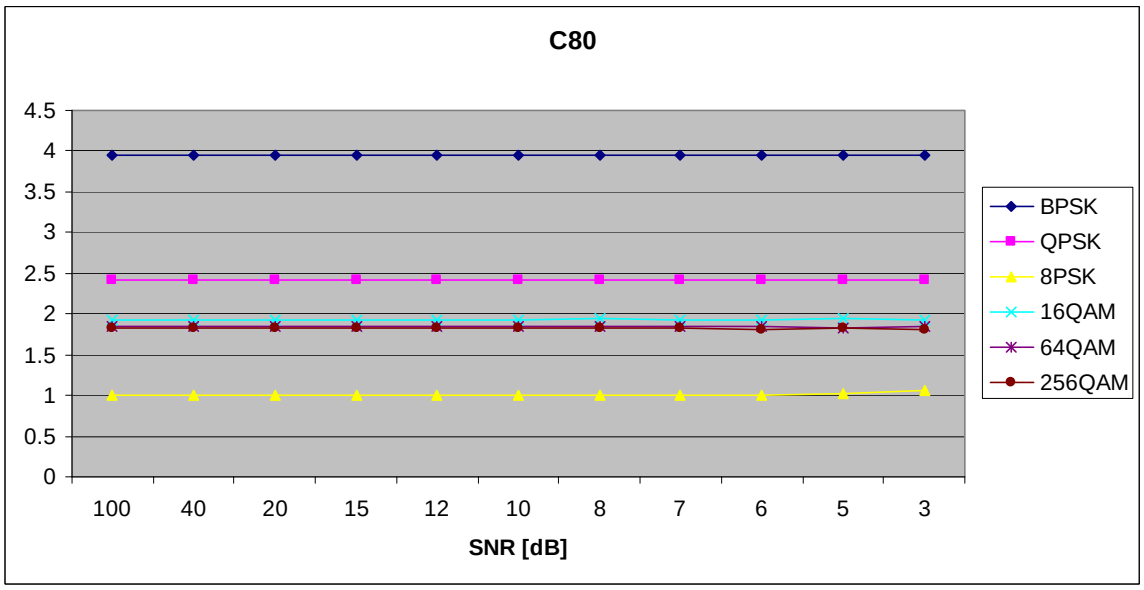

Figure 9. $C_{x,8,0}$ (AWGN only).

Figures 10 through 12 show the same cumulants for a slow, flat Rayleigh channel while Figures 13 through 15 show them for a fast, frequency selective channel. The values do not change much under moderate fading conditions, but under more severe fading they change significantly. Since any classifier is only as good as the features it works with, this indicates that severe fading conditions will be problematic.

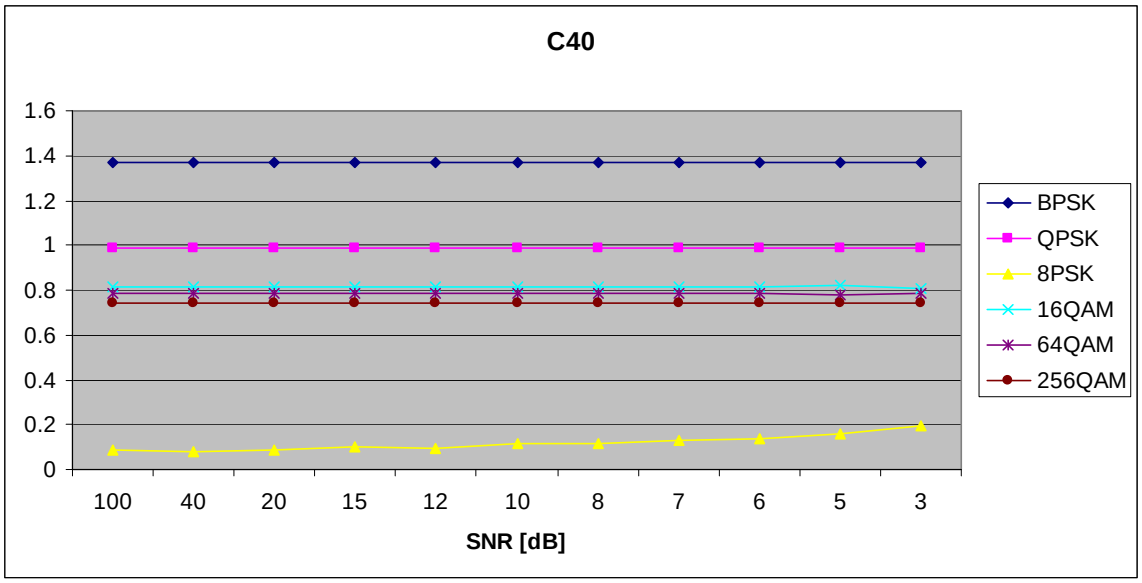

Figure 10. $C_{x,4,0}$ (slow, flat Rayleigh fading).

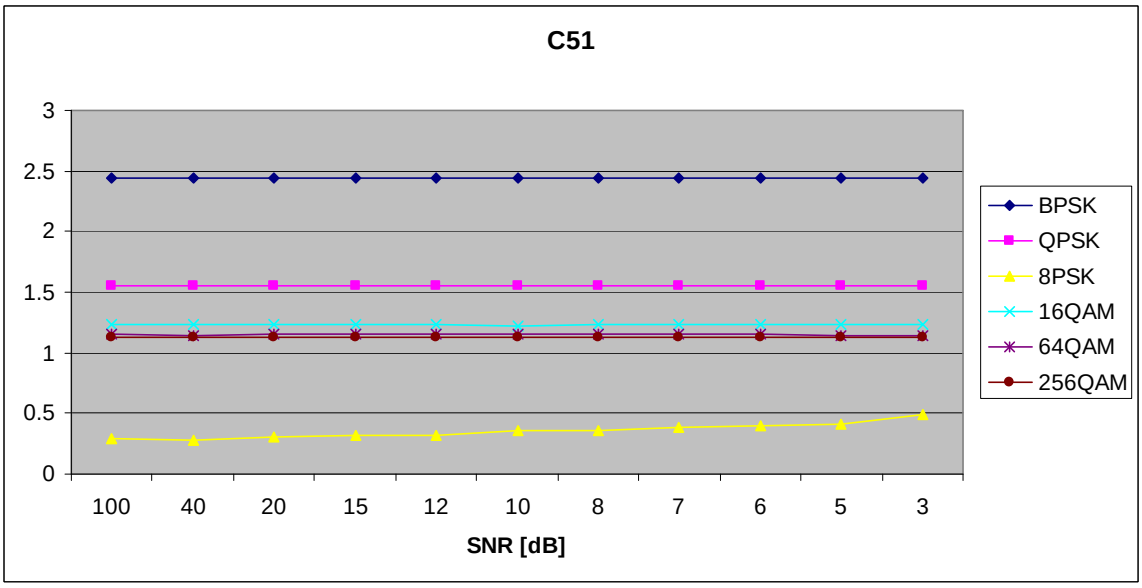

Figure 11. $C_{x,5,1}$ (slow, flat Rayleigh fading).

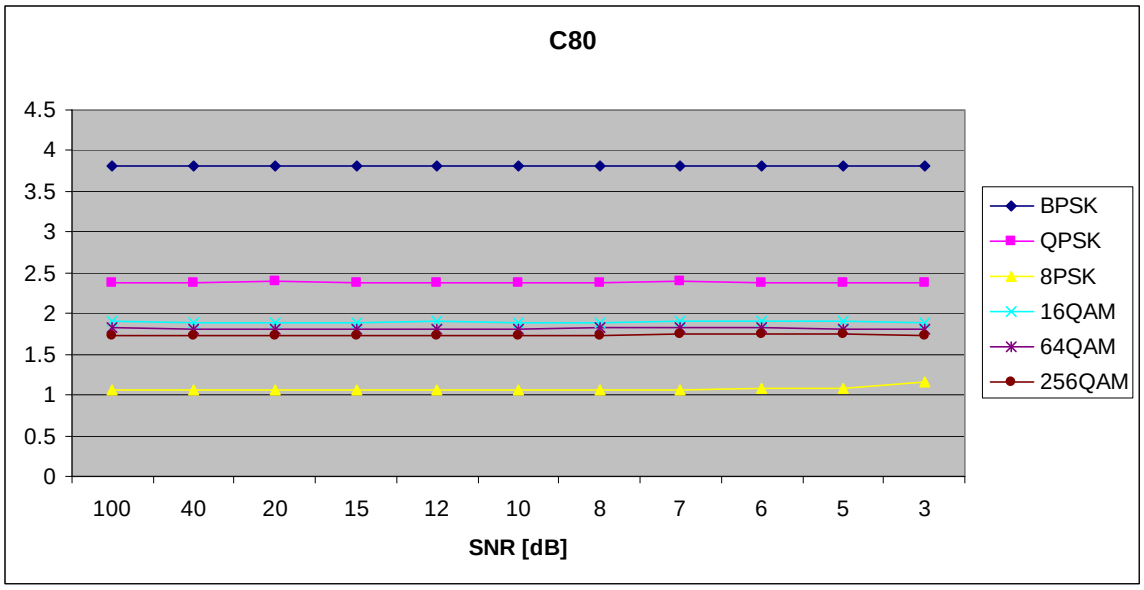

Figure 12. $C_{x,8,0}$ (slow, flat Rayleigh fading).

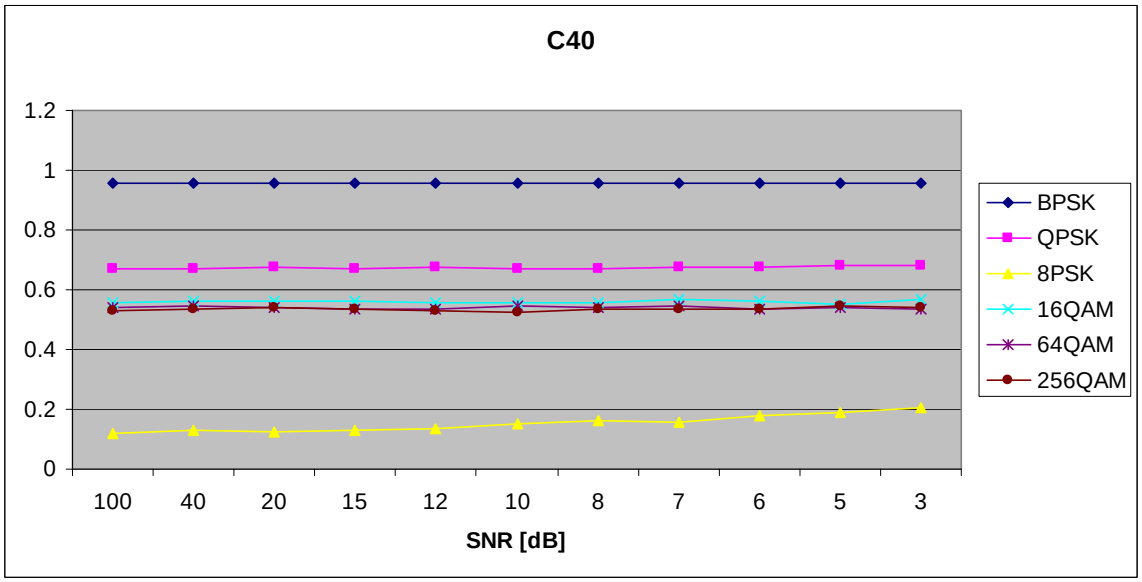

Figure 13. $C_{x,4,0}$ (fast, frequency-selective Rayleigh fading).

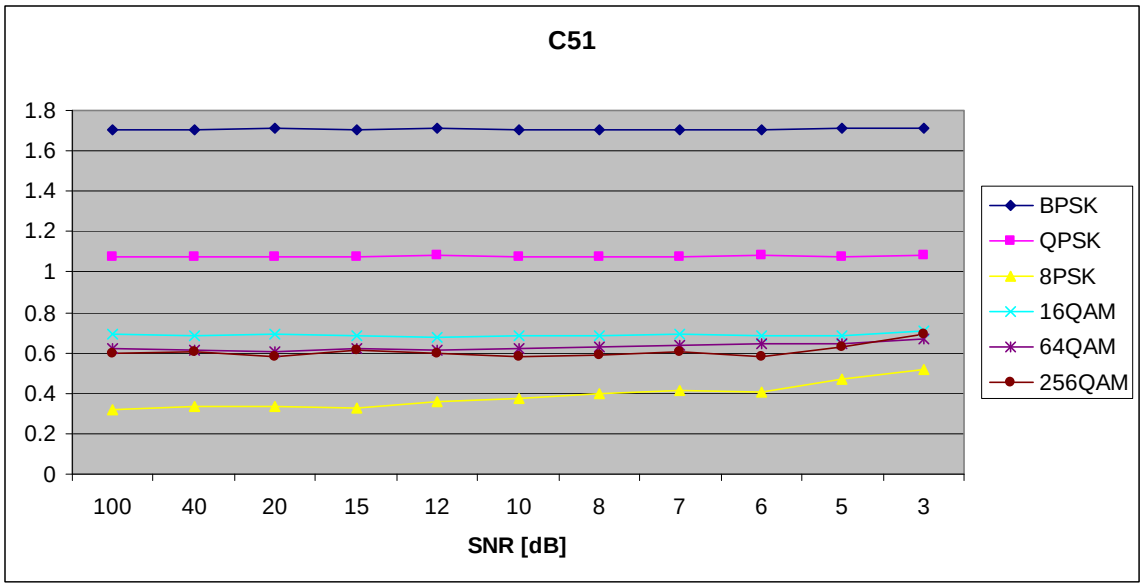

Figure 14. $C_{x,5,1}$ (fast, frequency-selective Rayleigh fading).

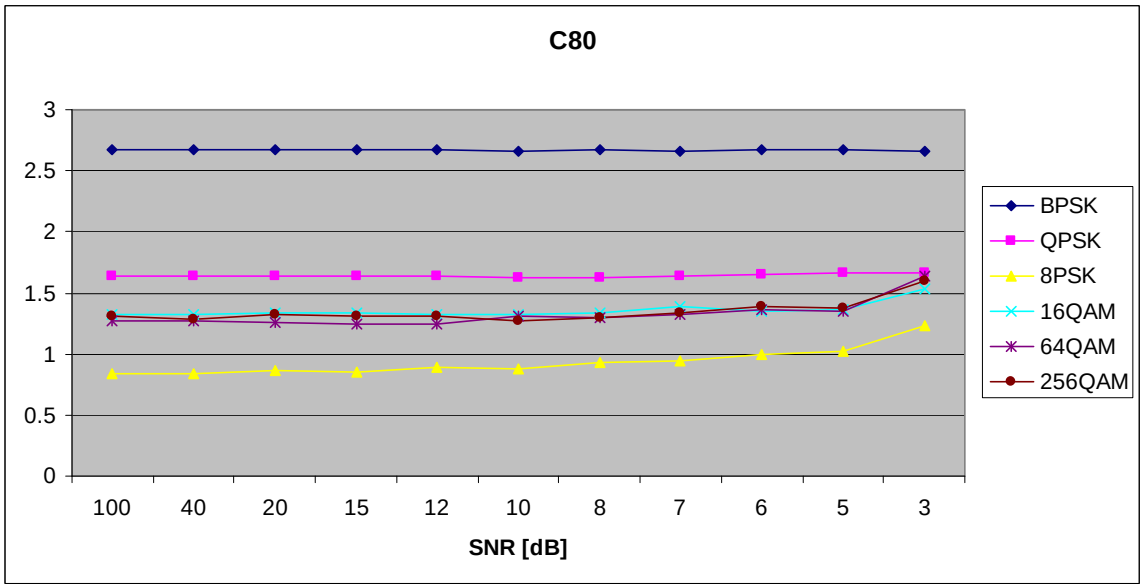

Figure 15. $C_{x,8,0}$ (fast, frequency-selective Rayleigh fading).

F. CONCLUSION

So far, we have seen the features that will form the basis for classification. These features will be grouped into 3-dimensional vectors for exploitation by different classifiers, which are the subject of Chapter IV.

IV. LINEAR AND NONLINEAR CLASSIFIERS

Now that discriminating features have been selected, a classification scheme must be identified to exploit them. This chapter discusses several classification methods.

A. INTRODUCTION

The use of higher-ordered moments and cumulants to classify modulation types was explored in [2] and [3] using a neural network and a classification tree, respectively. The purpose of this thesis is to investigate a different set of classification schemes, all of which rely on eigenvector decomposition in some way. This chapter will present two linear methods, PCA and LDA, and their nonlinear counterparts.

B. LINEAR CLASSIFICATION SCHEMES

Linear classification schemes work by projecting data into a feature space using a linear mapping and then comparing the result to a centroid for each class. If the data is linearly separable, these schemes work well. These algorithms attempt to find the best projection matrix.

1. Principal Component Analysis

PCA seeks the best representation of the data in a least-squares sense [18]. It does this by decomposing the data covariance matrix into its eigenvectors and choosing the most significant of them to form a projection matrix. The training data in this case is a matrix X of column vectors, denoted x_i, containing the chosen signal statistics. Assume that there are a total of P of these statistical profiles. Let N be the number of classes (i.e., modulation types) represented in the training set. Also let n be the number of training profiles in each class so that $P = nN$.

Define the mean profile as

$$m = \frac{1}{P}\sum_{i=1}^{P} x_i. \tag{4.1}$$

The first step is to subtract the mean profile from each of the *P* profiles since it is common to all of them and thus contains no information useful for classification:

$$\overline{x_i} = x_i - m. \tag{4.2}$$

These new centered training profiles form the new centered training matrix $\overline{X}$. Define the covariance matrix as

$$C = \overline{X}\,\overline{X}^T \tag{4.3}$$

and decompose it into its eigenvectors w_i. The eigenvectors corresponding to the top *k* eigenvalues form the projection matrix *W*. In this application, all of the possible eigenvectors were significant and were included in the projection matrix.

The columns of *W* define the feature space into which the statistical profiles will be projected. In order to determine the centroids for each class, first project the centered training matrix into this space:

$$T = W^T \overline{X} = \left[W^T \overline{x}_1 \,|\, W^T \overline{x}_2 \,|\, ... \,|\, W^T \overline{x}_P \right]. \tag{4.4}$$

The centroid for class *l* will be

$$g_l = \frac{1}{n} \sum_{i=1}^{n} W^T \overline{x}_i. \tag{4.5}$$

In order to use the classifier, compute the projection of a test profile *t* into the feature space,

$$t' = W^T \left(t - m \right), \tag{4.6}$$

and then choose the centroid having the smallest Euclidean distance to the point t'.

2. Linear Discriminant Analysis

In one sense, PCA is just a compression scheme in that it constructs the feature space by arbitrarily selecting the most significant eigenvectors as basis vectors. LDA, on the other hand, seeks a different projection matrix *W* that will maximize the separation between classes [18].

To explain this further, it is necessary to define the within-class scatter matrix S_W and the between-class scatter matrix S_B, defined as

$$S_W = \sum_{i=1}^{N} S_i, \tag{4.7}$$

where S_i is the covariance matrix of class *i*, (defined in the same way as Equation 3.13 above, except that it only contains profiles from class *i*) and

$$S_B = \sum_{i=1}^{N} n(m_i - m)(m_i - m)^T, \tag{4.8}$$

where m_i is the mean profile of class *i* and *m* is defined as in Equation 3.11. If these same matrices were calculated in the feature space formed by projecting the profiles using the projection matrix *W*, the results would be $W^T S_W W$ and $W^T S_B W$, respectively. The problem for LDA becomes finding the matrix W that maximizes the ratio of the determinants of these two matrices,

$$\frac{|W^T S_B W|}{|W^T S_W W|}. \tag{4.9}$$

From [18], finding the matrix W in the expression above is equivalent to solving the Eigen problem

$$S_B w_i = \lambda_i S_W w_i, \tag{4.10}$$

where w_i is the *i*th column of *W* for *i*=1 to the number of features. As with PCA, the columns are arranged in order of decreasing eigenvalues and an arbitrary number are kept.

Note, in some applications it is necessary to apply PCA before LDA; otherwise, S_W may be singular. This requirement is typical in applications where the dimension of the data is larger than the number of training observations. However, the dimension of the feature vector is three in our application, as only three representative features were selected, and LDA can be applied directly.

The class centroids and the projection of a given signal profile into the feature space are calculated in the same manner as in PCA.

C. NONLINEAR CLASSIFICATION SCHEMES

PCA and LDA work well with classes that are linearly separable. To visualize linear separability, consider a projected feature space which has three dimensions. If the classes are linearly separable, simple planes can be drawn as the decision boundaries between the classes. For data that is not linearly separable, it would be preferable to draw curved planes separating the classes. Conceptually, this can be achieved by a nonlinear mapping of the data into a higher dimensional space in which the classes are linearly separable. In the original space, the decision boundaries between them would appear curved.

1. The Kernel "Trick"

Carrying out an arbitrary nonlinear mapping to the higher-dimensional space may be computationally expensive. However, if the mapping function is chosen well, it may be encapsulated in a mechanism commonly known as the kernel "trick." Rather than operating on individual data points, kernel functions compute the dot product between two data points after projection into the higher dimensional space. The projection of an individual data point into the space is formed by computing its dot product with every other member of the space and arranging the dot products in a vector.

As an example, consider the following mapping function from two to three dimensions presented in [19]:

$$\left(x_1, x_2\right) \rightarrow \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right). \tag{4.11}$$

The goal is to find a function that will express the dot product between two points x and y in the higher dimensional space in terms of a function in the lower dimensional space. It can be shown that the function

$$k\left(x, y\right) = \left(x \cdot y\right)^2 \tag{4.12}$$

accomplishes this goal. Common kernel functions include the following:

polynomial:

$$k\left(x, y\right) = \left(a\left(x \cdot y\right) + b\right)^d \tag{4.13}$$

Gaussian:

$$k(x, y) = \exp\left(\frac{-(x-y)^2}{2\sigma^2}\right) \tag{4.14}$$

sigmoid:

$$k(x, y) = \tanh(\kappa(x \cdot y) + \theta). \tag{4.15}$$

2. Nonlinear Component Analysis

Nonlinear Component Analysis is the extension of the PCA method to higher dimensional spaces via the kernel trick; hence, it is sometimes known as kernel PCA (KPCA). The first step is to construct the $P \times P$ kernel matrix *K* by applying the kernel trick to each pair of points in $\overline{X}$ (this can usually be done with a matrix equation). Thus, *K* is the matrix of training data in the higher dimensional space. Once again, the projection matrix *W* is formed by using the eigen-decomposition of the matrix *K*. If dimension reduction is desired, only the most significant eigenvectors need to be kept. However, the matrix *K* must be reformed in order for its dimensions to match other matrices by applying the following steps:

$$K = W\Gamma W^T, \tag{4.16}$$

where Γ is the diagonal matrix of eigenvalues. At that point, the columns of *W* should be normalized by the magnitude of each column in the matrix $\left(W^T K W\right)^{1/2}$. The projected training data is calculated by

$$T = W^T K \tag{4.17}$$

and the class centroids are calculated in the same manner as with the linear PCA scheme. To project a test profile *t*, first translate it to the higher dimensional space by

$$t' = t^T \overline{X} \tag{4.18}$$

and then project it for classification, leading to

$$t'' = W^T t'. \tag{4.19}$$

3. General Discriminant Analysis

As was the case for their linear counterparts, GDA differs from KPCA by taking into account the scatter within classes. The kernel matrix *K* is formed as in the previous

section and decomposed into eigenvectors U and eigenvalues Γ. Next, the block diagonal matrix B is formed from N square matrices. The dimension of these matrices is $n \times n$ and each coefficient in the matrix is equal to $1/n$.

These results are used to find the eigenvectors β of the matrix $U^T BU$. The projection matrix is found by computing

$$W = U\Gamma^{-1}\beta \tag{4.20}$$

and normalizing by the matrix $\left(W^T KW\right)^{1/2}$ [20]. The class centroids and projected testing data are calculated in the same manner as the previous section.

D. CONCLUSION

Several well-known classification schemes were presented in this chapter. In the next chapter, their application to the features selected in Chapter III are presented.

V. IMPLEMENTATION AND RESULTS

The previous four chapters have laid the groundwork for classifying digital signals. In this chapter, the generation of the signals is explained, and the implementation of the classifier is discussed. Finally, we present simulation results. All MATLAB code developed for this thesis can be found in Appendix D.

A. SIGNAL GENERATION AND CORRUPTION

The MATLAB Communications Toolbox provides standard objects for generating digital signals. As noted in Chapter I, these objects simulate the signals at baseband. For PSK and QAM, this means that the output is the complex envelope of the signal with one sample per symbol. By definition, FSK signals cannot be represented as a single frequency, so they are generated with several samples per symbol, but the resulting spectrum is still centered at 0 Hz.

In terms of a physical system, this means that the digital signal has already been downconverted from its carrier frequency to baseband. In the case of PSK and QAM, it also means that the symbol rate has been obtained. Both of these assumptions are reasonable given an accurate spectrum measurement.

For fading channels, the Communications Toolbox also provides functions to generate filter objects for Rayleigh and Ricean channels. For either type of fading, the number of paths must be specified along with the gain and time delay for each one. Additionally, the Doppler shift must be specified. These parameters were varied to simulate slow and fast fading as well as frequency-flat and frequency-selective channels. Whether or not the channel involves fading, AWGN can be added to the signals using a standard function in MATLAB.

B. CLASSIFIER IMPLEMENTATION

Separate functions were developed to implement the four classifiers discussed in Chapter IV, but they all peform similar tasks. First, fifty training signals of 20,000

symbols are generated for each modulation type and estimates for the three selected cumulants computed for each one. Next, these parameters are passed to a training function that determines a projection matrix and class centroids.

In each trial of the testing phase, one thousand signals (also of 20,000 symbols) are generated and a modulation scheme is chosen at random for each one. One hundred trials are run for each combination of testing SNR and fading parameters in order to ensure that the results are statistically significant. The spectrum of each signal is analyzed to see if it matches one of the FSK types. To do this, an FFT is calculated on the first 4096 samples of the signal. The ratios of the second to the third peaks, the fourth to the fifth peaks, and the eighth to the ninth peaks are calculated and compared to threshold values estimated in a separate simulation. If the ratio of the second to the third peak is above the respective threshold, the signal is classified as BFSK. If not, similar comparisons are made for the other ratios to determine if the signal is 4-FSK or 8-FSK.

If none of the thresholds is crossed, the signal is not FSK. Its cumulants are calculated and projected into the new feature space using the projection matrix calculated by the training algorithm. The projected features are compared to the class centroids and the closest one is chosen as the modulation type. A confusion matrix is saved so that the classifier's performance can be analyzed for each modulation type. The procedure for classifying a test signal is shown in Figure 16.

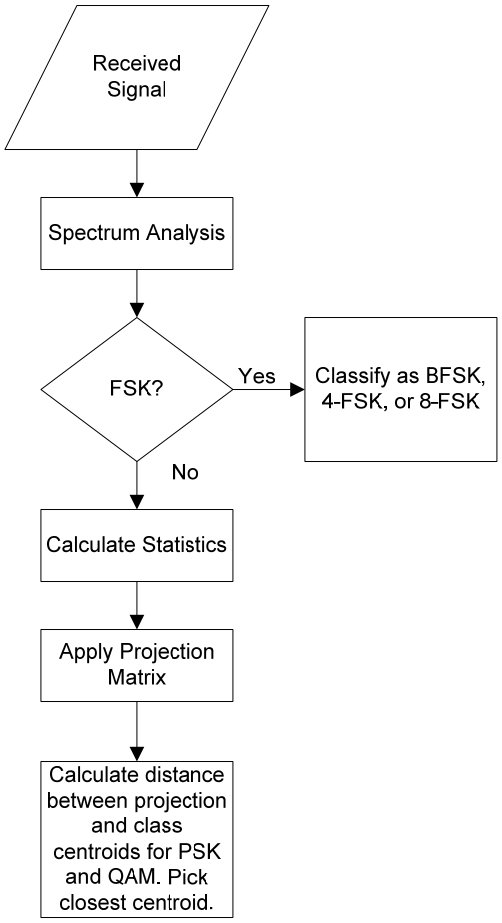

Figure 16. Classification Flow Chart for Test Signals

One question that must be addressed is the SNR at which the classifier is trained. A classifier trained at 20 dB will have better accuracy handling test signals at 20 dB than 5 dB and vice versa. However, after applying the noise-free power normalization discussed in Chapter III, these differences were so slight that it was found to be simpler to train the classifiers with uncorrupted data.

C. LINEAR VERSUS NONLINEAR CLASSIFIERS

The performance of the nonlinear classifiers was generally disappointing. As shown in Tables 4 through 11, for some kernels KPCA and GDA equaled the performance PCA and LDA, respectively. Note that in the case of KPCA, the best-performing scheme used a polynomial kernel of order one, which is just a more

complicated implementation of PCA. Given that nonlinear schemes involve significantly more computations, the linear schemes should be used.

The results that compare all four schemes are presented below as confusion matrices for signals in AWGN (more extensive results for the performance of PCA and LDA in fading channels is presented in section D). The overall performance and standard deviation are presented at the top of each table. Several different kernels were tried as well as different kernel parameters for each one. Only the kernel type leading to the best results is presented for KPCA and GDA schemes. PCA achieved the best overall classification rate, although section D shows that LDA performs better in fast Ricean channels.

1. PCA

PCA achieves the best overall performance and has the added benefit of being the simplest classifier to implement. At an SNR of 20 dB (shown in Table 4), the errors are primarily due to 64-QAM and 256-QAM being confused for each other. At 5 dB (shown in Table 5) there is more confusion between these two. Additionally, 16-QAM is sometimes mistaken for the other QAM schemes, and a significant number of 8-FSK signals are classified as 8-PSK

Table 4. Confusion Matrix for PCA classifier in AWGN, SNR = 20 dB.

		Classifier Output (Average Performance = 95.1%, Standard Deviation = 0.6%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11197	0	0	0	0	0	0	0	0	11197
	QFSK	0	11272	0	0	0	0	0	0	0	11272
	8FSK	0	0	11016	0	0	106	0	0	0	11122
	BPSK	0	0	0	11186	0	0	0	0	0	11186
	QPSK	0	0	0	0	10903	0	0	0	0	10903
	8PSK	0	0	0	0	0	10995	0	0	0	10995
	16QAM	0	0	0	0	0	0	11220	0	0	11220
	64QAM	0	0	0	0	0	0	5	9342	1640	10987
	256QAM	0	0	0	0	0	0	0	3166	7952	11118

Table 5. Confusion Matrix for PCA classifier in AWGN, SNR = 5 dB.

Actual Modulation Type \ Classifier Output (Average Performance = 89.0%, Standard Deviation = 1.1%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11053	0	0	0	0	0	0	0	0	11053
QFSK	0	10971	0	0	0	0	0	0	0	10971
8FSK	0	1	10865	0	0	452	0	0	0	11318
BPSK	0	0	0	11243	0	0	0	0	0	11243
QPSK	0	0	0	0	11137	0	0	0	0	11137
8PSK	0	0	0	0	0	10967	0	0	0	10967
16QAM	0	0	0	0	0	0	10394	792	14	11200
64QAM	0	0	0	0	0	0	1371	5826	3774	10971
256QAM	0	0	0	0	0	0	489	4154	6497	11140

2. LDA

LDA performs as well as PCA at 20 dB (Table 6), although it misclassifies 64-QAM somewhat more and 256-QAM somewhat less. At 5 dB (Table 7) it does not perform as well as PCA due to misclassifying the QAM schemes more often.

Table 6. Confusion Matrix for LDA classifier in AWGN, SNR = 20 dB.

Actual Modulation Type \ Classifier Output (Average Performance = 94.9%, Standard Deviation = 0.66%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11073	0	0	0	0	0	0	0	0	11073
QFSK	0	11102	0	0	0	0	0	0	0	11102
8FSK	0	2	11006	0	0	109	0	0	16	11133
BPSK	0	0	1	11117	0	0	0	0	0	11118
QPSK	0	0	0	0	11309	0	0	0	0	11309
8PSK	0	0	0	0	0	11111	0	0	0	11111
16QAM	0	0	0	0	0	0	10946	0	0	10946
64QAM	0	0	0	0	0	0	1	8220	2892	11113
256QAM	0	0	0	0	0	0	0	2060	9035	11095

Table 7. Confusion Matrix for LDA classifier in AWGN, SNR = 5 dB.

Actual Modulation Type \ Classifier Output (Average Performance = 87.2%, Standard Deviation = 1.1%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11042	0	0	0	0	0	0	0	0	11042
QFSK	0	10999	0	0	0	0	0	0	0	10999
8FSK	0	1	10774	0	0	432	0	0	2	11209
BPSK	0	0	0	10984	0	0	0	0	0	10984
QPSK	0	0	0	0	11145	0	0	0	0	11145
8PSK	0	0	0	0	0	10955	0	0	5	10960
16QAM	0	0	0	0	0	0	9938	1226	91	11255
64QAM	0	0	0	0	0	0	1771	4847	4538	11156
256QAM	0	0	0	0	0	0	933	3829	6488	11250

3. KPCA

For KPCA, the best kernel turns out to be $k(x, y) = (x \cdot y + 1)^1$, which is actually a linear scheme. As one would expect, its performance is the same as that of PCA at both 20 dB and 5 dB (Tables 8 and 9, respectively). Because of the extra computational overhead required by KPCA, PCA is preferred.

Table 8. Confusion matrix for KPCA classifier in AWGN, SNR = 20 dB.

Actual Modulation Type \ Classifier Output (Average Performance = 95.1%, Standard Deviation = 0.7%	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	10975	0	0	0	0	0	0	0	0	10975
QFSK	0	11103	0	0	0	0	0	0	0	11103
8FSK	0	0	10982	0	0	119	0	0	0	11101
BPSK	0	0	1	10929	0	0	0	0	0	10930
QPSK	0	0	0	0	11062	0	0	0	0	11062
8PSK	0	0	0	0	0	11213	0	0	0	11213
16QAM	0	0	0	0	0	0	11317	0	0	11317
64QAM	0	0	0	0	0	0	4	9434	1656	11094
256QAM	0	0	0	0	0	0	0	3171	8034	11205

Table 9. Confusion matrix for KPCA classifier in AWGN, SNR = 5 dB.

Actual Modulation Type \ Classifier Output (Average Performance = 88.7%, Standard Deviation = 1.0%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11182	0	0	0	0	0	0	0	0	11182
QFSK	0	11328	0	0	0	0	0	0	0	11328
8FSK	0	1	10455	0	0	467	0	0	0	10923
BPSK	0	0	0	11078	0	0	0	0	0	11078
QPSK	0	0	0	0	11075	0	0	0	0	11075
8PSK	0	0	0	0	0	11073	0	0	0	11073
16QAM	0	0	0	0	0	0	10334	780	11	11125
64QAM	0	0	0	0	0	0	1431	5872	3889	11192
256QAM	0	0	0	0	0	0	493	4199	6332	11024

4. GDA

For GDA, the best kernel was Gaussian: $k(x, y) = \exp\left(\frac{-(x-y)^2}{1}\right)$. Its overall performance at both 20 dB and 5 dB (Tables 10 and 11, respectively) is about the same as LDA, but with slightly better performance for 256-QAM and slightly different performance for the QAM schemes.

Table 10. Confusion Matrix for GDA classifier in AWGN, SNR = 20 dB.

Actual Modulation Type \ Classifier Output (Average Performance = 94.8%, Standard Deviation = 0.6%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11082	0	0	0	0	0	0	0	0	11082
QFSK	0	10981	0	0	0	0	0	0	0	10981
8FSK	0	0	10994	0	0	118	0	0	0	11112
BPSK	0	0	0	11204	0	0	0	0	0	11204
QPSK	0	0	0	0	11007	0	0	0	0	11007
8PSK	0	0	0	0	0	11238	0	0	0	11238
16QAM	0	0	0	0	0	0	10677	316	0	10993
64QAM	0	0	0	0	0	0	0	8830	2425	11255
256QAM	0	0	0	0	0	0	0	2393	8735	11128

Table 11. Confusion Matrix for GDA classifier in AWGN, SNR = 5 dB.

		Classifier Output (Average Performance = 87.7%, Standard Deviation = 1.1%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11038	0	0	0	0	0	0	0	0	11038
	QFSK	0	10978	0	0	0	0	0	0	0	10978
	8FSK	0	1	10617	0	0	412	0	0	0	11030
	BPSK	0	0	0	11057	0	0	0	0	0	11057
	QPSK	0	0	0	0	11060	0	0	0	0	11060
	8PSK	0	0	0	0	0	11120	0	0	0	11120
	16QAM	0	0	0	0	0	0	8511	2834	15	11360
	64QAM	0	0	0	0	0	0	331	6637	4293	11261
	256QAM	0	0	0	0	0	0	85	4331	6680	11096

D. PERFORMANCE OF THE PCA AND LDA CLASSIFIERS IN AWGN AND FADING CONDITIONS

Tables 12 through 27 are confusion matrices for the LDA classifier. Tables 28 through 43 are confusion matrices for the PCA classifier. PCA performs slightly better than LDA in most conditions.

In MATLAB, the functions used to simulate fading effects were rayleighchan.m and ricianchan.m. The parameters passed to these functions were a sampling interval of 1×10^{-6}; a maximum Doppler shift of 3.5 Hz for slow fading and 5000 Hz for fast fading; path delays of 0 and 1×10^{-7} for frequency-flat channels and 0 and 2×10^{-6} for frequency-selective channels; average path gains of 0 and -10 dB; and a K-factor of 3 for Ricean channels.

1. LDA

In a slow, frequency-flat Rayleigh channel, LDA misclassifies some of the PSK signals for QAM. The main difference between the results at 20 dB (shown in Table 12) and 5 dB (shown in Table 13) is once again additional confusion of the QAM schemes for each other.

Table 12. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 20 dB).

Actual Modulation Type \ Classifier Output (Average Performance = 86.2%, Standard Deviation = 1.1%	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11118	0	0	0	0	1	0	0	0	11119
QFSK	0	11269	1	0	0	3	0	0	12	11285
8FSK	0	10	10557	0	0	301	0	0	136	11004
BPSK	0	0	0	10768	219	8	0	0	0	10995
QPSK	0	0	0	0	10460	0	503	21	149	11133
8PSK	0	0	0	0	0	10948	0	0	3	10951
16QAM	0	0	0	0	0	0	7173	2169	1838	11180
64QAM	0	0	0	0	0	0	0	3635	7632	11267
256QAM	0	0	0	0	0	0	0	782	10284	11066

Table 13. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 5 dB).

Actual Modulation Type \ Classifier Output (Average Performance = 82.3%, Standard Deviation = 1.2%	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	10940	0	0	0	0	10	0	0	1	10951
QFSK	0	11083	1	0	0	36	0	0	7	11127
8FSK	0	9	10178	0	0	919	0	0	87	11193
BPSK	0	0	0	10737	255	13	1	0	0	11006
QPSK	0	0	0	0	10393	0	474	22	168	11057
8PSK	0	0	0	0	0	11116	0	0	91	11207
16QAM	0	0	0	0	0	15	6369	2683	2077	11144
64QAM	0	0	0	0	0	17	888	3224	7051	11180
256QAM	0	0	0	0	0	21	471	2356	8287	11135

The classifier fares better in Ricean fading conditions due to the presence of the line-of-sight path. Once again, most of the difference between the results at 20 dB and 5 dB (shown in Tables 14 and 15, respectively) is due to additional misclassifications of QAM signals.

Table 14. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 91.1%, Standard Deviation = 0.9%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11208	0	0	0	0	0	0	0	0	11208
	QFSK	0	11100	1	0	0	10	0	0	0	11111
	8FSK	0	3	11002	0	0	307	0	0	0	11312
	BPSK	0	0	0	10989	57	0	0	0	0	11046
	QPSK	0	0	0	0	10914	0	135	6	25	11080
	8PSK	0	0	0	0	0	11064	0	0	0	11064
	16QAM	0	0	0	0	0	0	9552	861	564	10977
	64QAM	0	0	0	0	0	0	1	5170	5862	11033
	256QAM	0	0	0	0	0	0	0	1043	10126	11169

Table 15. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 85.5%, Standard Deviation = 1.0%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11119	0	0	0	0	0	0	0	0	11119
	QFSK	0	11117	0	0	0	24	0	0	0	11141
	8FSK	0	1	10347	0	0	765	0	0	1	11114
	BPSK	0	0	0	11032	56	0	0	0	0	11088
	QPSK	0	0	0	0	11033	0	141	6	29	11209
	8PSK	0	0	0	0	0	11159	0	0	0	11159
	16QAM	0	0	0	0	0	1	8181	2059	761	11002
	64QAM	0	0	0	0	0	0	1162	4175	5781	11118
	256QAM	0	0	0	0	0	0	561	3133	7356	11050

In Tables 16 and 17, the classifier's performance in fast, frequency-flat Rayleigh conditions is no better than a guess. The increased Doppler shift causes most signals to be misclassified as either 8-PSK or 256-QAM, and there is almost no difference between the results at 20 dB (shown in Table 16) and the results at 5 dB (shown in Table 17).

Table 16. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 20 dB).

Actual Modulation Type \ Classifier Output (Average Performance = 11.9%, Standard Deviation = 1.0%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	0	4	71	0	0	9774	0	0	1270	11119
QFSK	0	1	15	0	0	9952	0	0	1207	11175
8FSK	0	0	4	0	0	9879	0	0	1188	11071
BPSK	0	0	0	289	160	3799	480	50	6190	10968
QPSK	0	0	0	244	168	3593	581	26	6430	11042
8PSK	0	0	0	296	0	10610	0	0	134	11040
16QAM	0	0	0	2574	315	7234	168	91	790	11172
64QAM	0	0	0	2808	243	7376	95	100	601	11223
256QAM	0	0	0	2941	195	7355	77	87	535	11190

Table 17. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 5 dB).

Actual Modulation Type \ Classifier Output (Average Performance = 11.3%, Standard Deviation = 0.98%)	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	0	0	67	72	0	10337	0	0	584	11060
QFSK	0	0	21	74	0	10555	0	0	595	11245
8FSK	0	0	1	73	0	10307	0	0	643	11024
BPSK	0	0	0	320	190	4187	668	49	5697	11111
QPSK	0	0	0	307	197	4028	668	47	6059	11306
8PSK	0	0	0	536	0	10108	2	0	373	11019
16QAM	0	0	0	3014	419	6666	233	72	602	11006
64QAM	0	0	0	4009	336	6376	131	68	335	11255
256QAM	0	0	0	4153	281	6048	100	55	337	10974

In Tables 18 and 19, the classifier appears to perform much better in Ricean conditions even though the fading parameters are otherwise the same as in Tables 16 and 17. This is because the line-of-sight component preserves the spectral content of the FSK signals, resulting in relatively high classification rates for them. The rates for the PSK and QAM signals are just as bad as in fast, frequency-flat Rayleigh conditions.

Table 18. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 52.6%, Standard Deviation = 1.6%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11014	2	0	0	0	22	0	0	27	11065
	QFSK	0	10881	5	0	0	124	0	0	131	11141
	8FSK	0	3	7799	0	0	1600	0	0	1592	10994
	BPSK	0	0	0	1149	9979	0	5	0	0	11133
	QPSK	0	0	0	0	34	23	1550	1632	7849	11088
	8PSK	0	0	0	0	0	10894	0	0	211	11105
	16QAM	0	0	0	0	12	161	117	8	10881	11179
	64QAM	0	0	0	0	14	255	141	6	10696	11112
	256QAM	0	0	0	0	21	258	147	7	10750	11183

Table 19. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 49.3%, Standard Deviation = 1.5%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11096	1	2	0	0	56	0	0	10	11165
	QFSK	0	10669	9	0	0	314	0	0	34	11026
	8FSK	0	1	5995	0	0	4404	0	0	533	10933
	BPSK	0	0	0	1309	9709	0	0	4	1	11023
	QPSK	0	0	0	0	64	36	2208	1524	7193	11025
	8PSK	0	0	0	3	0	10227	0	0	897	11127
	16QAM	0	0	0	2	47	459	414	24	10268	11214
	64QAM	0	0	0	6	56	866	586	9	9713	11236
	256QAM	0	0	0	2	64	1014	633	14	9524	11251

In Tables 20 and 21, the results for slow, frequency-selective Rayleigh conditions are shown. At 20 dB, many signals are mistaken for QAM. At 5 dB, 8-FSK is also frequently mistaken for 8-PSK.

Table 20. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 65.0%, Standard Deviation = 1.4%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10643	5	8	0	0	36	0	0	365	11057
	QFSK	333	8621	37	0	0	5	0	0	2302	11298
	8FSK	0	654	6289	0	0	10	0	0	4258	11211
	BPSK	0	0	0	9612	1528	27	40	0	40	11247
	QPSK	0	0	0	0	6383	209	2280	207	2024	11103
	8PSK	0	0	0	0	0	10598	0	0	588	11186
	16QAM	0	0	0	0	0	96	1606	1799	7426	10927
	64QAM	0	0	0	0	0	60	0	304	10568	10932
	256QAM	0	0	0	0	0	78	0	38	10923	11039

Table 21. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 63.2%, Standard Deviation = 1.3%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10916	1	0	0	0	314	0	0	129	11360
	QFSK	304	8397	22	0	0	1443	0	0	1016	11182
	8FSK	0	573	5120	0	0	3461	0	0	1851	11005
	BPSK	0	0	0	9257	1548	44	40	3	30	10922
	QPSK	0	0	0	0	6490	188	2269	184	1998	11129
	8PSK	0	0	0	0	0	10094	0	0	877	10971
	16QAM	0	0	0	0	1	149	1788	1657	7535	11130
	64QAM	0	0	0	0	0	182	247	866	9820	11115
	256QAM	0	0	0	0	0	161	148	596	10281	11186

In Tables 22 and 23, it can be seen once again that the classifier performs better in Ricean conditions (all other parameters being the same as in Tables 20 and 21). Many signals are misclassified as QAM, and at 5 dB 8-FSK is often misclassified as 8-PSK.

Table 22. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 69.3%, Standard Deviation = 1.3%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10889	1	0	0	0	0	0	0	188	11078
	QFSK	222	9072	32	0	0	0	0	0	1654	10980
	8FSK	0	496	6705	0	0	0	0	0	3925	11126
	BPSK	0	0	0	10194	864	12	28	24	43	11165
	QPSK	0	0	0	0	7958	93	1935	224	1018	11228
	8PSK	0	0	0	0	0	10903	0	0	277	11180
	16QAM	0	0	0	0	0	56	2037	2052	6814	10959
	64QAM	0	0	0	0	0	50	0	409	10637	11096
	256QAM	0	0	0	0	0	48	0	37	11103	11188

Table 23. Confusion Matrix for LDA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 67.3%, Standard Deviation = 1.2%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10970	0	1	0	0	125	0	0	51	11147
	QFSK	236	9065	19	0	0	1047	0	1	892	11260
	8FSK	0	438	5670	0	0	3076	0	2	1924	11110
	BPSK	0	0	0	10259	817	16	28	29	49	11198
	QPSK	0	0	0	0	7884	96	1855	255	959	11049
	8PSK	0	0	0	0	0	10154	0	0	947	11101
	16QAM	0	0	0	0	5	65	2307	1890	6761	11028
	64QAM	0	0	0	0	1	76	360	1141	9550	11128
	256QAM	0	0	0	0	2	71	202	810	9894	10979

The classifier once again breaks down in fast, frequency-selective Rayleigh conditions due to Doppler shift, which has a more severe effect than frequency selectivity. The results at 5 dB (shown in Table 25) do not differ significantly from those at 20 dB (shown in Table 24).

Table 24. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 20 dB).

Actual Modulation Type	Classifier Output (Average Performance = 11.7%, Standard Deviation = 0.9%									
	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	3	1	65	0	0	10263	0	0	759	11091
QFSK	0	0	22	0	0	10298	0	0	698	11018
8FSK	0	0	3	0	0	10408	0	0	742	11153
BPSK	0	0	0	108	128	4252	360	211	6081	11140
QPSK	0	0	0	90	102	4001	191	125	6664	11173
8PSK	0	0	0	124	0	10702	0	0	247	11073
16QAM	0	0	0	1561	358	8042	119	65	1026	11171
64QAM	0	0	1	1804	262	8115	42	39	794	11057
256QAM	0	0	0	1955	243	8224	36	22	644	11124

Table 25. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 5 dB).

Actual Modulation Type	Classifier Output (Average Performance = 11.2%, Standard Deviation = 1.1%									
	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	2	1	70	62	0	10589	0	0	517	11241
QFSK	0	0	16	71	0	10408	0	0	483	10978
8FSK	0	0	6	53	0	10599	0	0	523	11181
BPSK	0	0	0	154	198	4636	511	289	5309	11097
QPSK	0	0	0	148	135	4581	361	238	5618	11081
8PSK	0	0	0	262	0	10314	0	3	461	11040
16QAM	0	0	0	2290	602	7463	167	62	716	11300
64QAM	0	0	0	2997	477	7033	108	32	430	11077
256QAM	0	0	0	3357	471	6719	79	32	347	11005

In Tables 26 and 27, the presence of a line-of-sight component once again assists in the classification of FSK signals despite the fast, frequency-selective nature of the channel.

Table 26. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 53.6%, Standard Deviation = 1.6%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11142	0	0	0	0	20	0	0	36	11198
	QFSK	0	10956	8	0	0	127	0	0	127	11218
	8FSK	0	3	7711	0	0	1428	0	0	1917	11059
	BPSK	0	0	0	2635	8432	0	0	0	0	11067
	QPSK	0	0	0	0	0	0	567	1762	8831	11160
	8PSK	0	0	0	0	0	10093	0	0	1063	11156
	16QAM	0	0	0	0	6	8	25	14	10912	10965
	64QAM	0	0	0	0	7	22	35	16	11048	11128
	256QAM	0	0	0	0	2	20	48	7	10972	11049

Table 27. Confusion Matrix for LDA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 50.8%, Standard Deviation = 1.6%									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11059	1	2	0	0	39	0	0	4	11105
	QFSK	0	10831	11	0	0	350	0	0	49	11241
	8FSK	0	2	5885	0	0	4527	0	0	658	11072
	BPSK	0	0	0	2804	8214	0	0	0	0	11018
	QPSK	0	0	0	0	53	1	1877	1777	7371	11079
	8PSK	0	0	0	0	0	9800	0	0	1428	11228
	16QAM	0	0	0	0	36	129	324	82	10506	11077
	64QAM	0	0	0	0	33	347	478	75	10111	11044
	256QAM	0	0	0	0	31	435	594	74	10002	11136

2. PCA

PCA performs better than LDA in some cases. In Tables 28 and 29 no significant difference can be seen between the results for a slow, frequency-flat Rayleigh channel at 20 dB and 5 dB, respectively. Most errors are due to the QAM schemes being mistaken for each other.

Table 28. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 87.0%, Standard Deviation = 1.0%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11190	1	1	0	0	5	0	0	0	11197
	QFSK	3	11232	2	0	0	35	0	0	0	11272
	8FSK	0	17	10728	0	0	377	0	0	0	11122
	BPSK	0	0	1	10742	393	1	33	3	13	11186
	QPSK	0	0	0	5	10052	36	550	62	198	10903
	8PSK	0	0	0	0	0	10991	0	0	4	10995
	16QAM	0	0	0	0	1	103	8219	1372	1525	11220
	64QAM	0	0	0	0	1	144	310	4570	5962	10987
	256QAM	0	0	0	0	2	123	160	1605	9228	11118

Table 29. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Rayleigh Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 86.8%, Standard Deviation = 1.1%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11051	0	2	0	0	0	0	0	0	11053
	QFSK	0	10931	0	0	0	40	0	0	0	10971
	8FSK	0	19	10884	0	0	415	0	0	0	11318
	BPSK	0	0	1	10847	352	0	26	4	13	11243
	QPSK	0	0	0	8	10209	34	572	45	269	11137
	8PSK	0	0	0	0	0	10966	0	0	1	10967
	16QAM	0	0	0	0	3	108	8157	1412	1520	11200
	64QAM	0	0	0	0	0	146	319	4577	5929	10971
	256QAM	0	0	0	0	1	152	187	1634	9166	11140

As with LDA, the PCA classifier achieves better results in Ricean conditions. However, the PCA classifier does just as well at 5 dB (shown in Table 31) and 20 dB (shown in Table 30).

Table 30. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 91.5%, Standard Deviation = 1.0%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11197	0	0	0	0	0	0	0	0	11197
	QFSK	2	11249	0	0	0	21	0	0	0	11272
	8FSK	0	4	10808	0	0	310	0	0	0	11122
	BPSK	0	0	1	11089	85	0	7	0	4	11186
	QPSK	0	0	0	0	10664	13	150	13	63	10903
	8PSK	0	0	0	0	0	10995	0	0	0	10995
	16QAM	0	0	0	0	0	44	9917	779	480	11220
	64QAM	0	0	0	0	0	42	33	6128	4784	10987
	256QAM	0	0	0	0	0	44	11	1573	9490	11118

Table 31. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Flat Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 91.7%, Standard Deviation = 0.9%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11052	0	1	0	0	0	0	0	0	11053
	QFSK	2	10959	1	0	0	9	0	0	0	10971
	8FSK	0	11	11018	0	0	289	0	0	0	11318
	BPSK	0	0	0	11141	90	0	8	1	3	11243
	QPSK	0	0	0	0	10925	15	146	15	36	11137
	8PSK	0	0	0	0	0	10967	0	0	0	10967
	16QAM	0	0	0	0	0	33	10030	704	433	11200
	64QAM	0	0	0	0	0	46	33	6089	4803	10971
	256QAM	0	0	0	0	1	40	9	1605	9485	11140

In Tables 32 and 33, the classifier's performance degrades due to Doppler shift in the fast, frequency-flat Rayleigh channel, although, not to the same extent as the LDA classifier. Once again, the results are about the same at 20 dB and 5 dB.

Table 32. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 19.0%, Standard Deviation = 1.2%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	1	3	71	0	0	11122	0	0	0	11197
	QFSK	0	1	19	0	0	11252	0	0	0	11272
	8FSK	0	0	2	0	0	11120	0	0	0	11122
	BPSK	0	0	0	0	37	5283	171	91	5604	11186
	QPSK	0	0	0	0	37	5295	187	85	5299	10903
	8PSK	0	0	0	0	3	10292	25	14	661	10995
	16QAM	0	0	0	5	498	2049	1123	387	7158	11220
	64QAM	0	0	0	13	495	2121	1005	379	6974	10987
	256QAM	0	0	0	8	467	2053	1031	407	7152	11118

Table 33. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Rayleigh Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 18.9%, Standard Deviation = 1.2%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	0	0	86	0	0	10967	0	0	0	11053
	QFSK	0	0	15	0	0	10956	0	0	0	10971
	8FSK	0	0	5	0	0	11313	0	0	0	11318
	BPSK	0	0	0	0	49	5330	169	88	5607	11243
	QPSK	0	0	0	0	57	5407	150	82	5441	11137
	8PSK	0	0	0	0	8	10259	26	12	662	10967
	16QAM	0	0	0	5	457	2135	1128	425	7050	11200
	64QAM	0	0	0	13	471	2047	1080	388	6972	10971
	256QAM	0	0	0	17	498	2008	1102	412	7103	11140

In fast, frequency-flat Ricean conditions, PCA performs worse than LDA, mainly due to additional misclassifications of BPSK as QPSK. Tables 34 and 35 show no significant difference between SNRs of 20 dB and 5 dB, respectively.

Table 34. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 20 dB).

Classifier Output (Average Performance = 43.7%, Standard Deviation = 1.5%)

Actual Modulation Type	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11070	3	0	0	0	50	0	0	0	11123
QFSK	0	10794	6	0	0	219	0	0	0	11019
8FSK	0	1	7844	0	0	3231	0	0	0	11076
BPSK	0	0	0	0	11214	0	24	0	2	11240
QPSK	0	0	0	0	1	131	762	765	9427	11086
8PSK	0	0	0	0	0	10967	0	0	0	10967
16QAM	0	0	0	0	0	5764	6	3	5456	11229
64QAM	0	0	0	0	0	7730	5	5	3391	11131
256QAM	0	0	0	0	1	8079	4	4	3041	11129

Table 35. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Flat Ricean Fading (SNR = 5 dB).

Classifier Output (Average Performance = 43.8%, Standard Deviation = 1.7%)

Actual Modulation Type	BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
BFSK	11020	1	1	0	0	50	0	0	0	11072
QFSK	0	10903	8	0	0	248	0	0	0	11159
8FSK	0	0	7918	0	0	3291	0	0	0	11209
BPSK	0	0	0	0	11154	0	21	1	1	11177
QPSK	0	0	0	0	0	123	731	741	9532	11127
8PSK	0	0	0	0	0	10942	0	0	0	10942
16QAM	0	0	0	0	0	5647	3	3	5467	11120
64QAM	0	0	0	0	0	7675	1	3	3404	11083
256QAM	0	0	0	0	1	8097	8	3	3002	11111

PCA also performs slightly worse than LDA for slow, frequency-selective channels, whether a Rayleigh or Ricean channel model is used. In Tables 36 and 37, the results for Rayleigh fading are shown at 20 dB and 5 dB, respectively. Once again, the performance is nearly identical.

Table 36. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 62.0%, Standard Deviation = 1.6%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10897	3	5	0	0	320	0	0	0	11225
	QFSK	302	8476	34	0	0	2138	0	0	0	10950
	8FSK	1	568	6373	0	0	4369	0	0	0	11311
	BPSK	0	0	1	7889	3050	1	121	7	29	11098
	QPSK	0	0	0	0	5452	241	2655	331	2264	10943
	8PSK	0	0	0	0	0	11071	0	0	0	11071
	16QAM	0	0	0	0	0	1158	1952	1719	6380	11209
	64QAM	0	0	0	0	0	1493	13	585	9104	11195
	256QAM	0	0	0	0	0	1589	6	134	9269	10998

Table 37. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Rayleigh Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 61.9%, Standard Deviation = 1.7%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10698	3	2	0	0	337	0	0	0	11040
	QFSK	325	8564	48	0	0	2202	0	0	0	11139
	8FSK	0	573	6340	0	0	4240	0	0	0	11153
	BPSK	0	0	3	8001	3056	0	106	6	14	11186
	QPSK	0	0	0	0	5478	265	2702	355	2340	11140
	8PSK	0	0	0	0	0	11084	0	0	0	11084
	16QAM	0	0	0	0	0	1159	1855	1749	6401	11164
	64QAM	0	0	0	0	1	1468	11	615	8949	11044
	256QAM	0	0	0	0	0	1633	7	137	9273	11050

In Tables 38 and 39, the classifier performs slightly better in a slow, frequency-selective Ricean channel due to the line-of-sight component.

Table 38. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 66.3%, Standard Deviation = 1.5%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11043	3	3	0	0	176	0	0	0	11225
	QFSK	197	9057	38	0	0	1658	0	0	0	10950
	8FSK	0	471	6927	0	0	3913	0	0	0	11311
	BPSK	0	0	0	8934	2112	0	46	0	6	11098
	QPSK	0	0	0	0	6276	112	2758	277	1520	10943
	8PSK	0	0	0	0	0	11071	0	0	0	11071
	16QAM	0	0	0	0	0	591	2232	2170	6216	11209
	64QAM	0	0	0	0	0	757	1	671	9766	11195
	256QAM	0	0	0	0	0	853	0	104	10041	10998

Table 39. Confusion Matrix for PCA Classifier in AWGN Plus Slow, Frequency-Selective Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 66.4%, Standard Deviation = 1.4%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10826	1	3	0	0	210	0	0	0	11040
	QFSK	196	9285	29	0	0	1629	0	0	0	11139
	8FSK	0	480	6864	0	0	3809	0	0	0	11153
	BPSK	0	0	0	9017	2120	0	40	2	7	11186
	QPSK	0	0	0	0	6446	126	2827	297	1444	11140
	8PSK	0	0	0	0	0	11084	0	0	0	11084
	16QAM	0	0	0	0	0	578	2160	2163	6263	11164
	64QAM	0	0	0	0	0	831	0	638	9575	11044
	256QAM	0	0	0	0	0	818	0	116	10116	11050

In Tables 40 and 41, it can be seen that PCA performs better than LDA in a fast, frequency-selective Rayleigh channel. Once again, there are no significant differences between results at 20 dB and at 5 dB.

Table 40. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 18.1%, Standard Deviation = 1.3%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	2	0	61	0	0	11162	0	0	0	11225
	QFSK	0	0	12	0	0	10938	0	0	0	10950
	8FSK	0	0	0	0	0	11311	0	0	0	11311
	BPSK	0	0	0	0	23	6899	80	36	4060	11098
	QPSK	0	0	0	0	19	7561	56	25	3282	10943
	8PSK	0	0	0	0	0	10693	12	4	362	11071
	16QAM	0	0	0	3	205	3567	585	244	6605	11209
	64QAM	0	0	0	2	213	3606	528	213	6633	11195
	256QAM	0	0	1	4	213	3376	575	242	6587	10998

Table 41. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Rayleigh Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 18.2%, Standard Deviation = 1.4%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	0	1	57	0	0	10982	0	0	0	11040
	QFSK	0	0	22	0	0	11117	0	0	0	11139
	8FSK	0	0	1	0	0	11152	0	0	0	11153
	BPSK	0	0	0	1	21	7017	81	44	4022	11186
	QPSK	0	0	0	0	19	7609	58	41	3413	11140
	8PSK	0	0	0	0	1	10751	11	8	313	11084
	16QAM	0	0	0	2	203	3817	547	245	6350	11164
	64QAM	0	0	0	7	233	3491	578	234	6501	11044
	256QAM	0	0	0	3	223	3354	603	253	6614	11050

In Tables 42 and 43, the classifier's performance is seen to improve due to the line-of-sight component in the Ricean channel. The results are not quite as good as those for LDA.

Table 42. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 20 dB).

		Classifier Output (Average Performance = 42.4%, Standard Deviation = 1.5%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	10998	1	1	0	0	48	0	0	0	11048
	QFSK	0	10785	11	0	0	249	0	0	0	11045
	8FSK	0	2	7853	0	0	3464	0	0	0	11319
	BPSK	0	0	0	0	11173	0	2	0	0	11175
	QPSK	0	0	0	0	0	22	153	413	10402	10990
	8PSK	0	0	0	0	0	10866	0	0	0	10866
	16QAM	0	0	0	0	0	5386	0	0	5776	11162
	64QAM	0	0	0	0	0	8653	0	0	2418	11071
	256QAM	0	0	0	0	0	9432	0	0	1892	11324

Table 43. Confusion Matrix for PCA Classifier in AWGN Plus Fast, Frequency-Selective Ricean Fading (SNR = 5 dB).

		Classifier Output (Average Performance = 42.7%, Standard Deviation = 1.6%)									
		BFSK	QFSK	8FSK	BPSK	QPSK	8PSK	16QAM	64QAM	256QAM	Total
Actual Modulation Type	BFSK	11069	0	2	0	0	52	0	0	0	11123
	QFSK	0	10683	5	0	0	258	0	0	0	10946
	8FSK	0	0	7772	0	0	3357	0	0	0	11129
	BPSK	0	0	0	0	11119	0	2	0	0	11121
	QPSK	0	0	0	0	0	22	129	394	10684	11229
	8PSK	0	0	0	0	0	11196	0	0	0	11196
	16QAM	0	0	0	0	0	5267	0	0	5759	11026
	64QAM	0	0	0	0	0	8773	1	0	2356	11130
	256QAM	0	0	0	0	0	9090	0	0	2010	11100

In summary, LDA performs better than PCA in some channel conditions, but the results for PCA do not vary as much with SNR. In general, as channel conditions deteriorate, the two classifiers tend to mistakenly select either 8-PSK or 256-QAM. Ricean channels also generally show better results than Rayleigh channels because the line-of-sight component preserves the spectral content of the signal, allowing FSK signals to be correctly classified.

The next chapter summarizes this thesis and presents recommendations for further work.

VI. CONCLUSIONS

In this thesis, previous work that investigated the use of higher-ordered moments and cumulants in the problem of blind modulation classification was built upon. The criteria for selecting moments and cumulants to form feature vectors for the signals were discussed. Two classical classification algorithms, PCA and LDA, as well their nonlinear, kernel-based equivalents, KPCA and GDA, were applied, and the effects of AWGN and multipath fading on resulting performances were investigated.

Whereas previous work assumed that the statistics would be purely real, this work demonstrated the effect of rotating the symbol constellation by a phase offset. This result was used to select cumulants whose magnitudes are unaffected by phase shifts. Additionally, power normalizations were applied to the cumulants to make them more robust to fading effects.

Nonlinear classifiers turned out not to have any advantage in a problem with low-dimension feature vectors. The linear schemes turned out to work better (and with fewer computations). The steps taken to make the features more resistant to the effects of fading were effective for modest Doppler shift and for frequency-flat channels. The classifiers' performance dropped sharply, however, with increasing Doppler shift.

The linear classifiers proposed here are very effective in channels that only undergo corruption by AWGN and in the sorts of fading channels one might expect from a person walking and using a mobile device. On the whole, however, fading remains a challenge for blind modulation classification. Future work in this area should focus on methods to compensate for fading.

Additionally, one drawback to the work presented here is that it uses continuous streams of symbols. A more sophisticated simulation should attempt to account for phenomena above the physical layer of communications systems, such as packetized data. Furthermore, simulations in the passband should be conducted to see if this problem is tractable when dealing with modern techniques such as spread spectrum transmission.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: CUMULANTS EXPRESSED AS FUNCTIONS OF EQUAL AND LOWER ORDER MOMENTS

Table 44. Cumulants Expressed In Terms of Moments.

Cumulant	In Terms of Moments
$C_{x,2,0}$	$E_{x,2,0}$
$C_{x,1,1}$	$E_{x,1,1}$
$C_{x,4,0}$	$E_{x,4,0}-3\left(E_{x,2,0}\right)^2$
$C_{x,3,1}$	$E_{x,3,1}-3E_{x,2,0}E_{x,1,1}$
$C_{x,2,2}$	$E_{x,2,2}-\left(E_{x,2,0}\right)^2-2\left(E_{x,1,1}\right)^2$
$C_{x,6,0}$	$E_{x,6,0}-15E_{x,2,0}E_{x,4,0}+30\left(E_{x,2,0}\right)^3$
$C_{x,5,1}$	$E_{x,5,1}-10E_{x,2,0}E_{x,3,1}-5E_{x,1,1}E_{x,4,0}+30\left(E_{x,2,0}\right)^2E_{x,1,1}$
$C_{x,4,2}$	$E_{x,4,2}-E_{x,2,0}E_{x,4,0}-8E_{x,1,1}E_{x,3,1}-6E_{x,2,0}E_{x,2,2}+6\left(E_{x,2,0}\right)^3+24\left(E_{x,1,1}\right)^2E_{x,2,0}$
$C_{x,3,3}$	$E_{x,3,3}-6E_{x,2,0}E_{x,3,1}-9E_{x,1,1}E_{x,2,2}+18\left(E_{x,2,0}\right)^2E_{x,1,1}+12\left(E_{x,1,1}\right)^3$
$C_{x,8,0}$	$E_{x,8,0}-35\left(E_{x,4,0}\right)^2-630\left(E_{x,2,0}\right)^4+420\left(E_{x,2,0}\right)^2E_{x,4,0}$
$C_{x,7,1}$	$E_{x,7,1}-35E_{x,4,0}E_{x,3,1}-630\left(E_{x,2,0}\right)^3E_{x,1,1}+210E_{x,4,0}E_{x,2,0}E_{x,1,1}+210E_{x,2,0}E_{x,3,1}$
$C_{x,6,2}$	$E_{x,6,2}-15E_{x,4,0}E_{x,2,2}-20\left(E_{x,3,1}\right)^2+30E_{x,4,0}\left(E_{x,2,0}\right)^2+60E_{x,4,0}\left(E_{x,1,1}\right)^2$ $+240E_{x,3,1}E_{x,1,1}E_{x,2,0}+90E_{x,2,2}\left(E_{x,2,0}\right)^2-90\left(E_{x,2,0}\right)^4-540\left(E_{x,2,0}\right)^2\left(E_{x,1,1}\right)^2$

$C_{x,5,3}$	$E_{x,5,3}-5E_{x,4,0}E_{x,3,1}-30E_{x,3,1}E_{x,2,2}+90E_{x,3,1}\left(E_{x,2,0}\right)^2+120E_{x,3,1}\left(E_{x,1,1}\right)^2$ $+180E_{x,2,2}E_{x,1,1}E_{x,2,0}+30E_{x,4,0}E_{x,2,0}E_{x,1,1}-270\left(E_{x,2,0}\right)^3E_{x,1,1}-360\left(E_{x,1,1}\right)^3E_{x,2,0}$
$C_{x,4,4}$	$E_{x,4,4}-\left(E_{x,4,0}\right)^2-18\left(E_{x,2,2}\right)^2-16\left(E_{x,3,1}\right)^2-54\left(E_{x,2,0}\right)^4-144\left(E_{x,1,1}\right)^4$ $-432\left(E_{x,2,0}\right)^2\left(E_{x,1,1}\right)^2+12E_{x,4,0}\left(E_{x,2,0}\right)^2+96E_{x,3,1}E_{x,1,1}E_{x,2,0}+144E_{x,2,2}\left(E_{x,1,1}\right)^2$ $+72E_{x,2,2}\left(E_{x,2,0}\right)^2+96E_{x,3,1}E_{x,2,0}E_{x,1,1}$

APPENDIX B: BEHAVIOR OF MOMENTS AND CUMULANTS WITH DECREASING SNR

A. AWGN ONLY

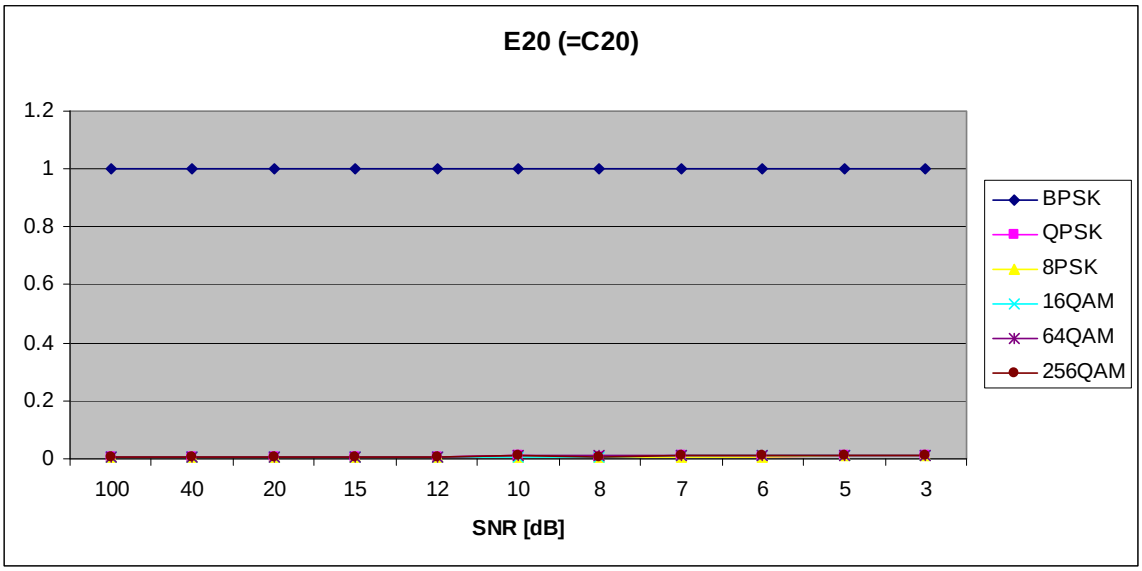

Figure 17. $E_{x,2,0}$ in AWGN.

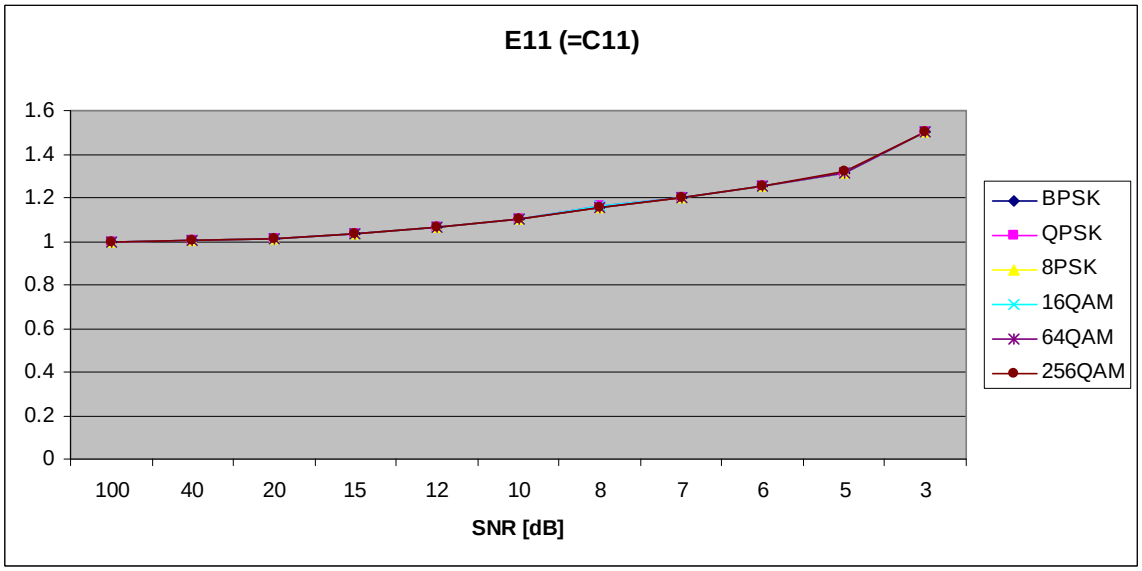

Figure 18. $E_{x,1,1}$ in AWGN.

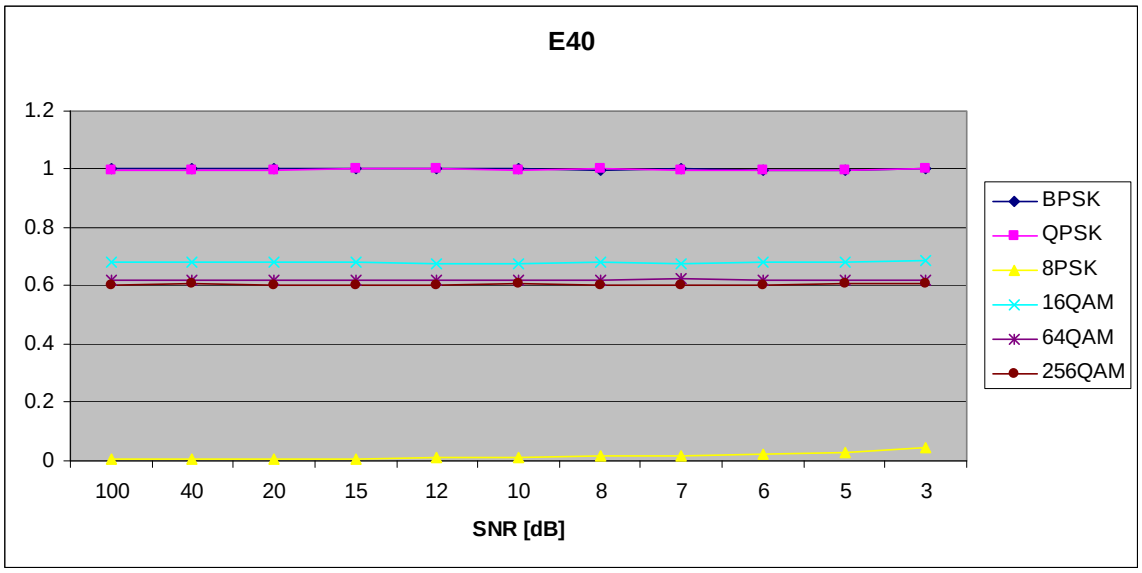

Figure 19. $E_{x,4,0}$ in AWGN.

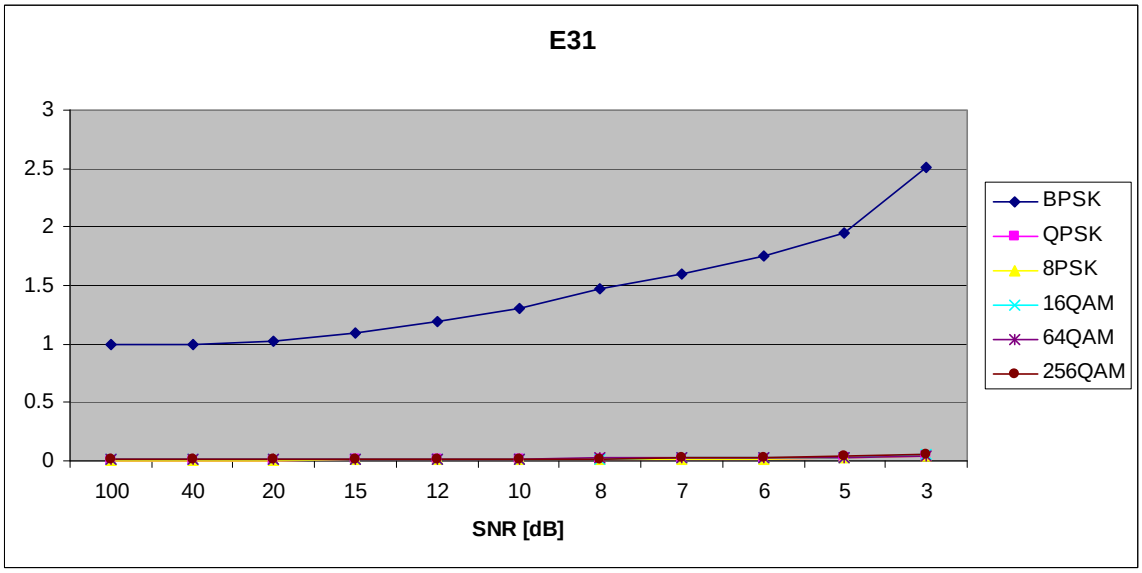

Figure 20. $E_{x,3,1}$ in AWGN.

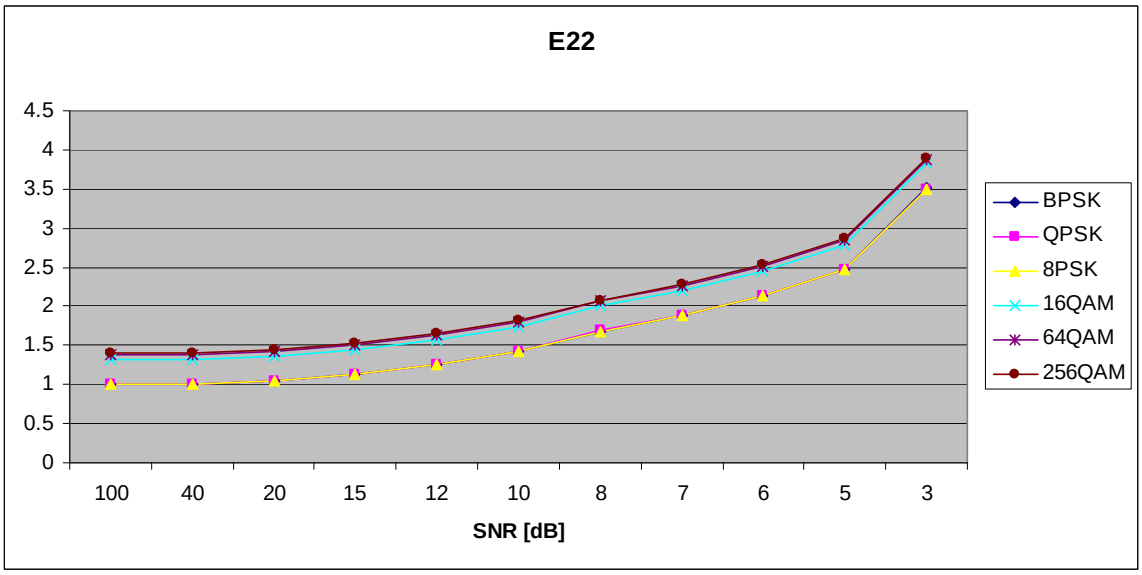

Figure 21. $E_{x,2,2}$ in AWGN.

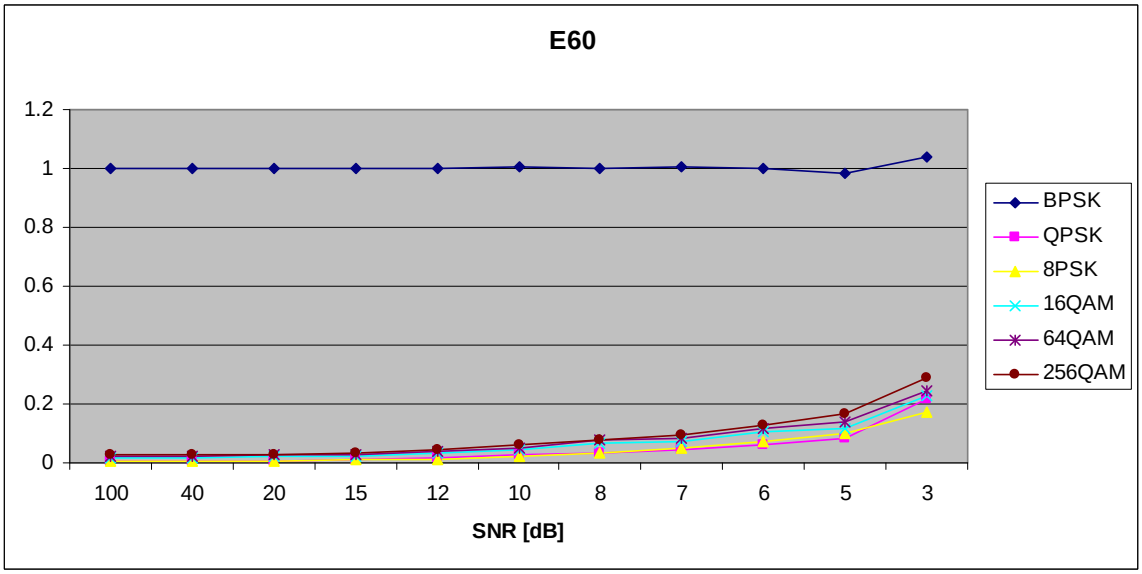

Figure 22. $E_{x,6,0}$ in AWGN.

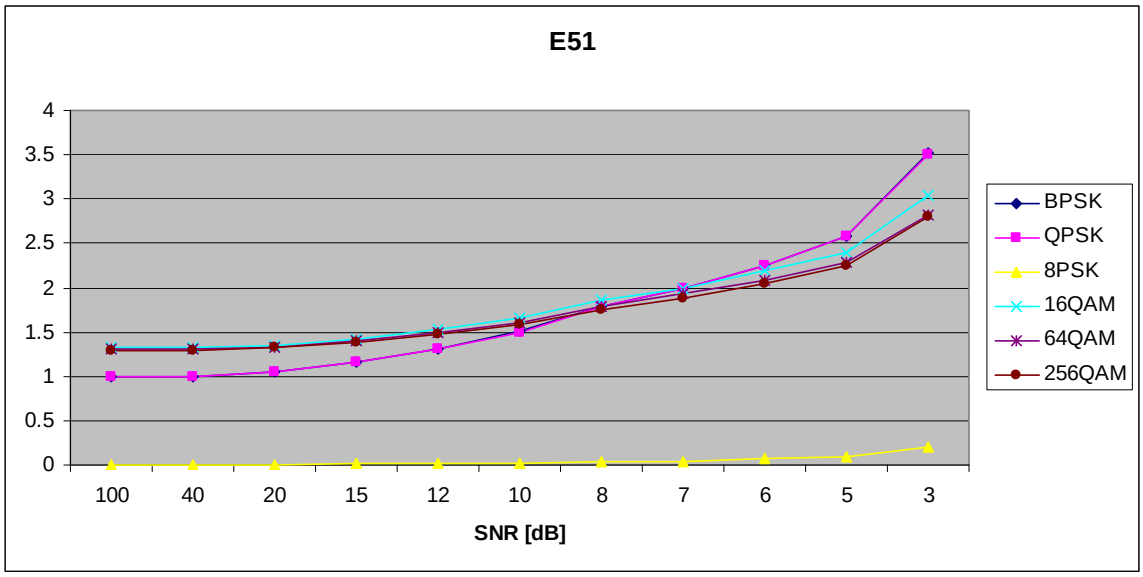

Figure 23. $E_{x,5,1}$ in AWGN.

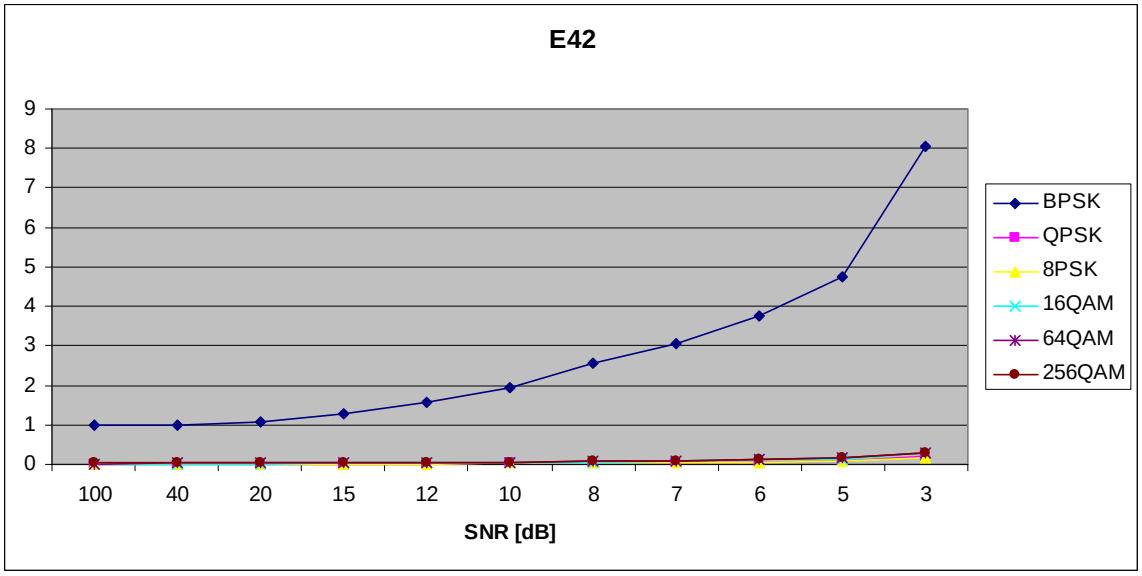

Figure 24. $E_{x,4,2}$ in AWGN.

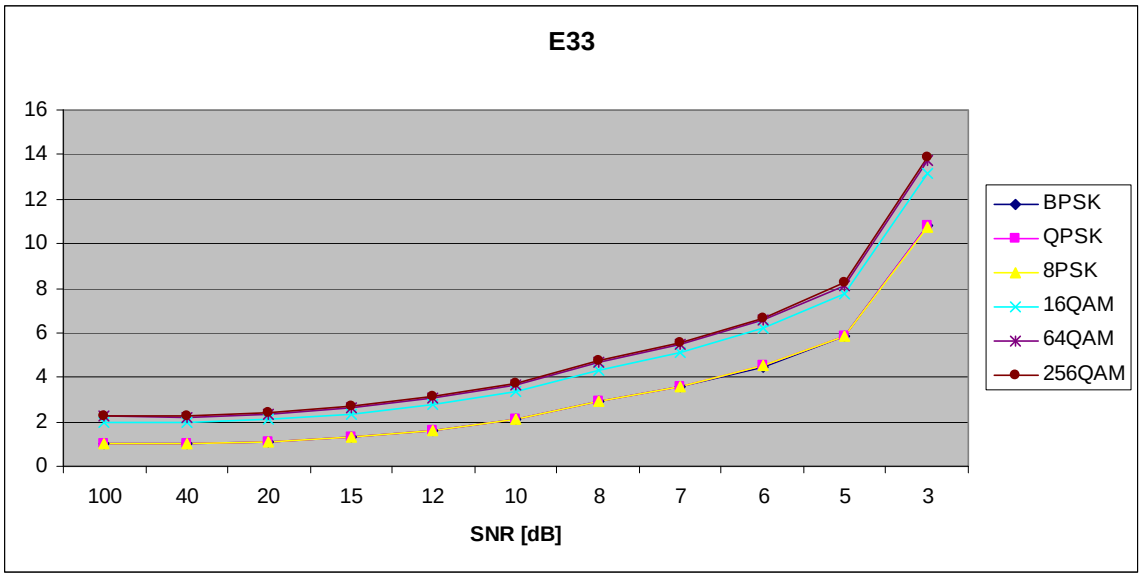

Figure 25. $E_{x,3,3}$ in AWGN.

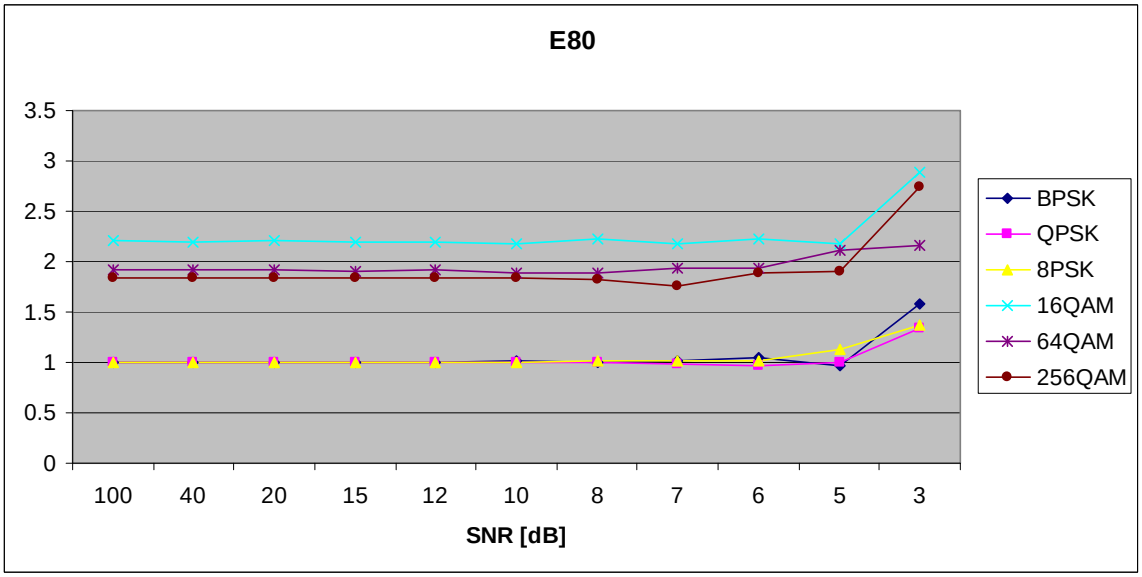

Figure 26. $E_{x,8,0}$ in AWGN.

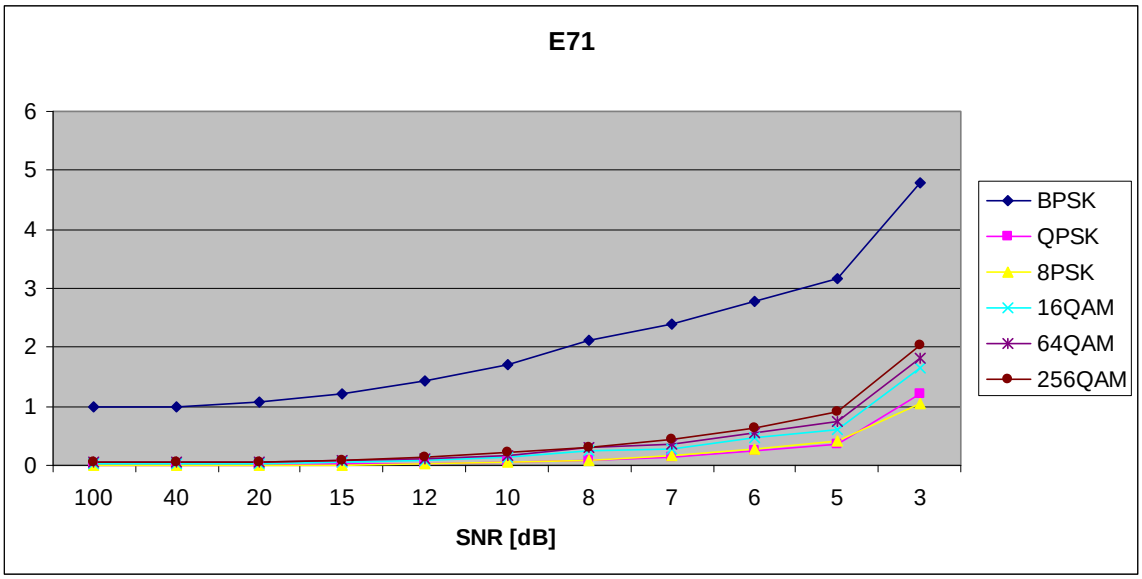

Figure 27. $E_{x,7,1}$ in AWGN.

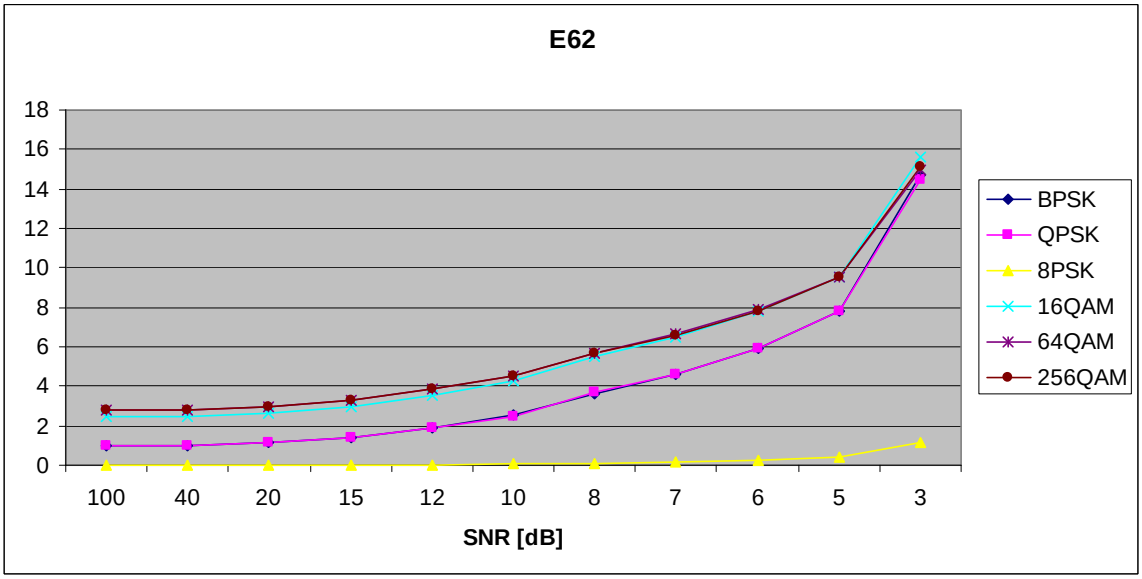

Figure 28. $E_{x,6,2}$ in AWGN.

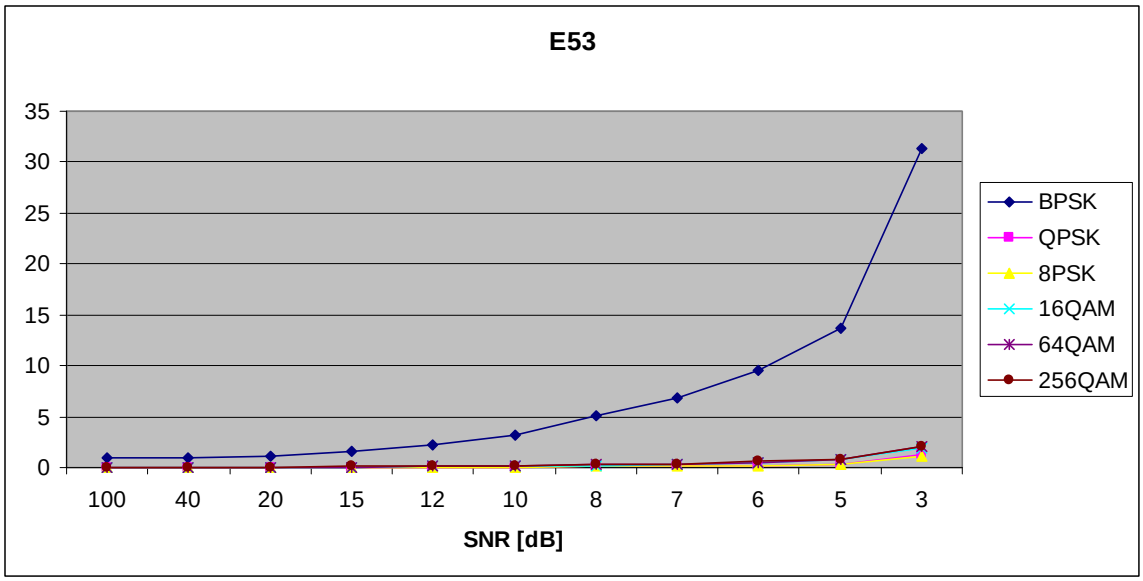

Figure 29. $E_{x,5,3}$ in AWGN.

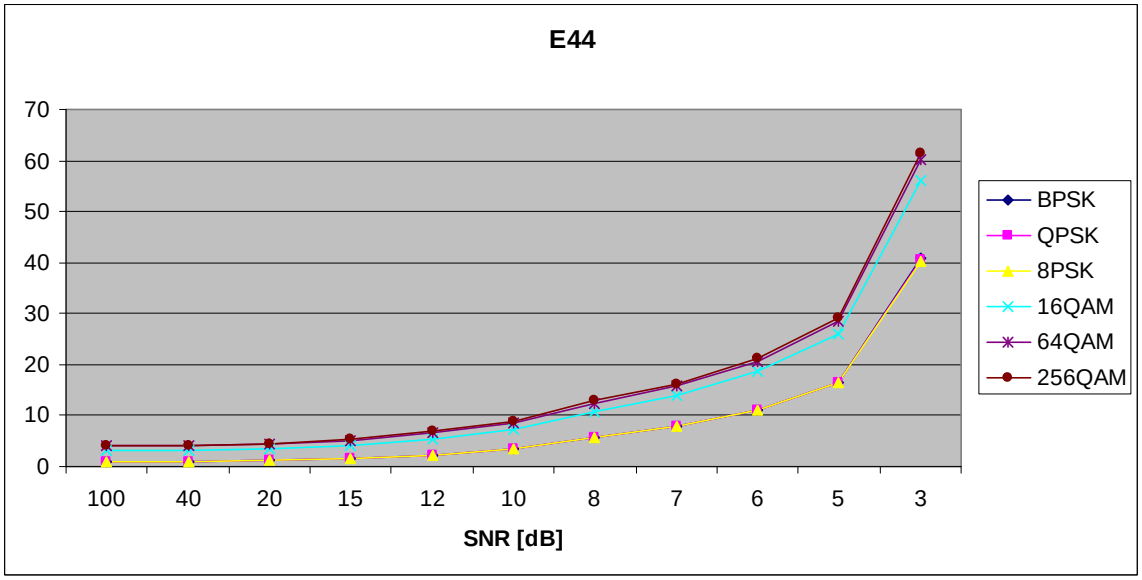

Figure 30. $E_{x,4,4}$ in AWGN.

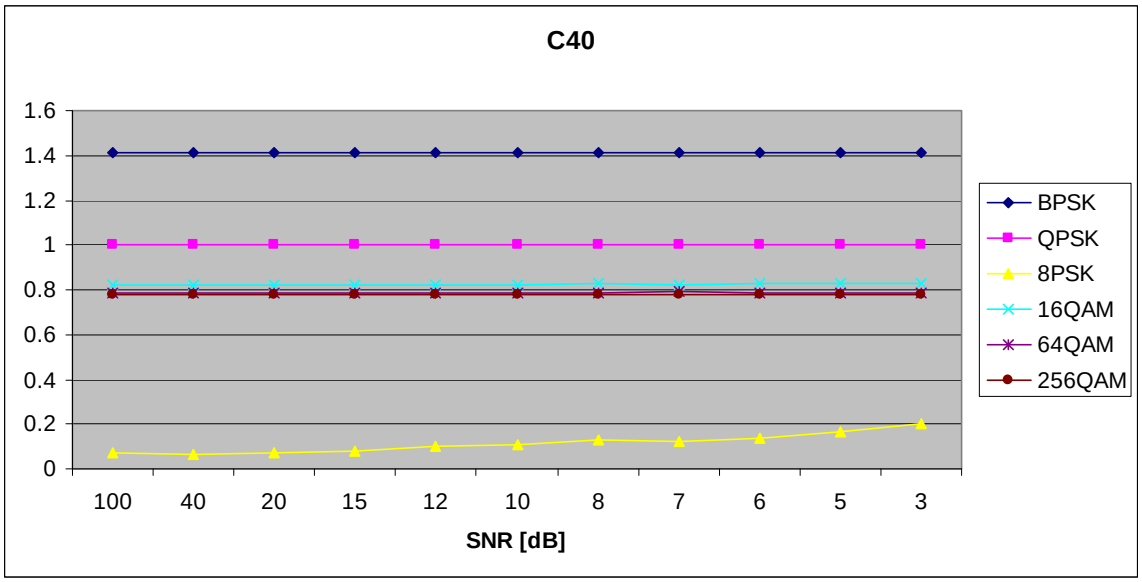

Figure 31. $C_{x,4,0}$ in AWGN.

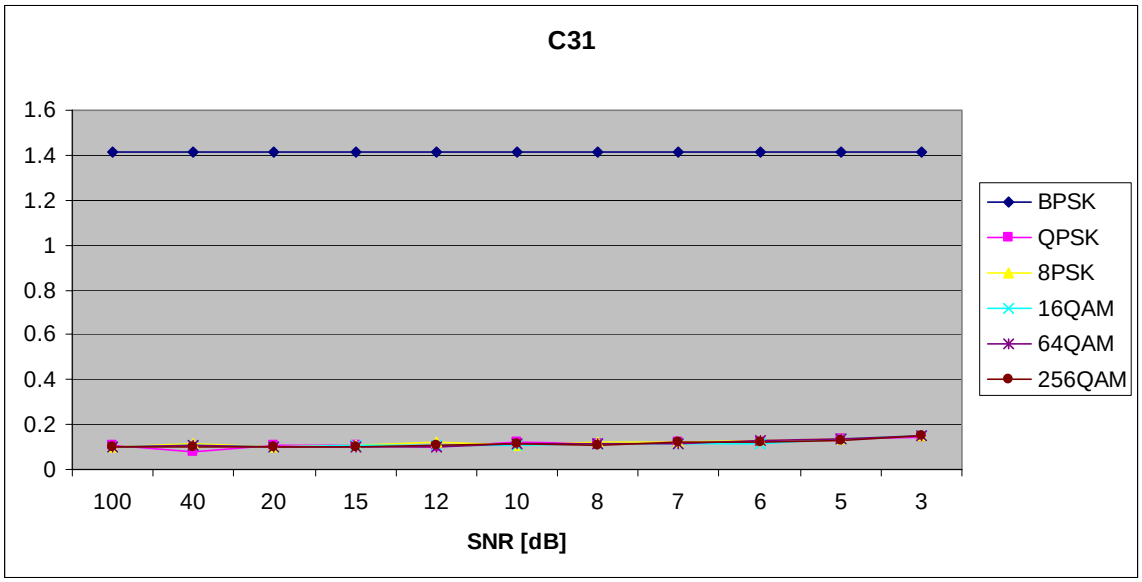

Figure 32. $C_{x,3,1}$ in AWGN.

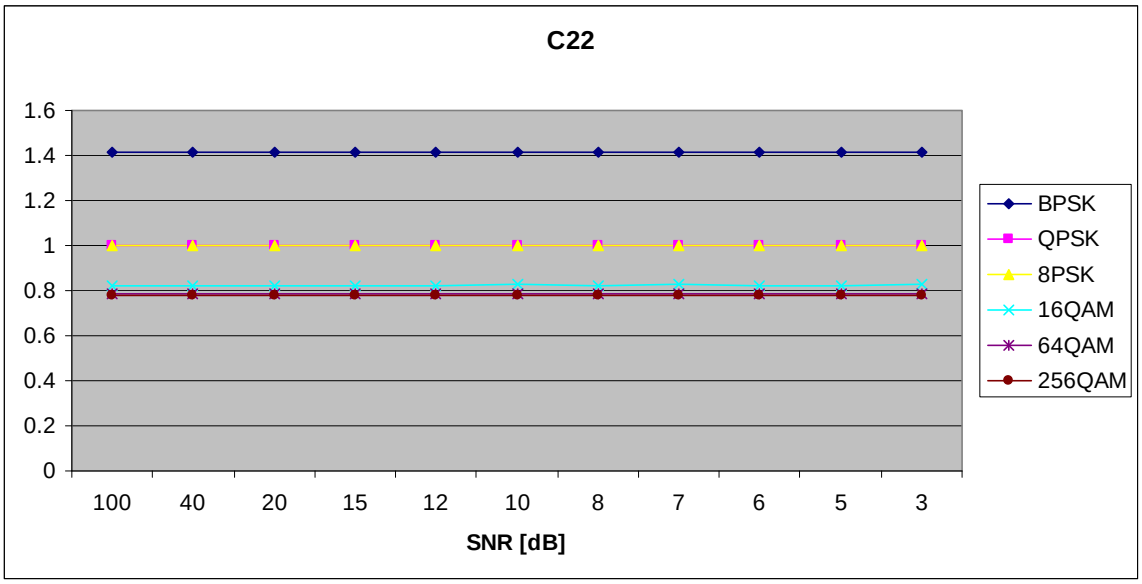

Figure 33. $C_{x,2,2}$ in AWGN.

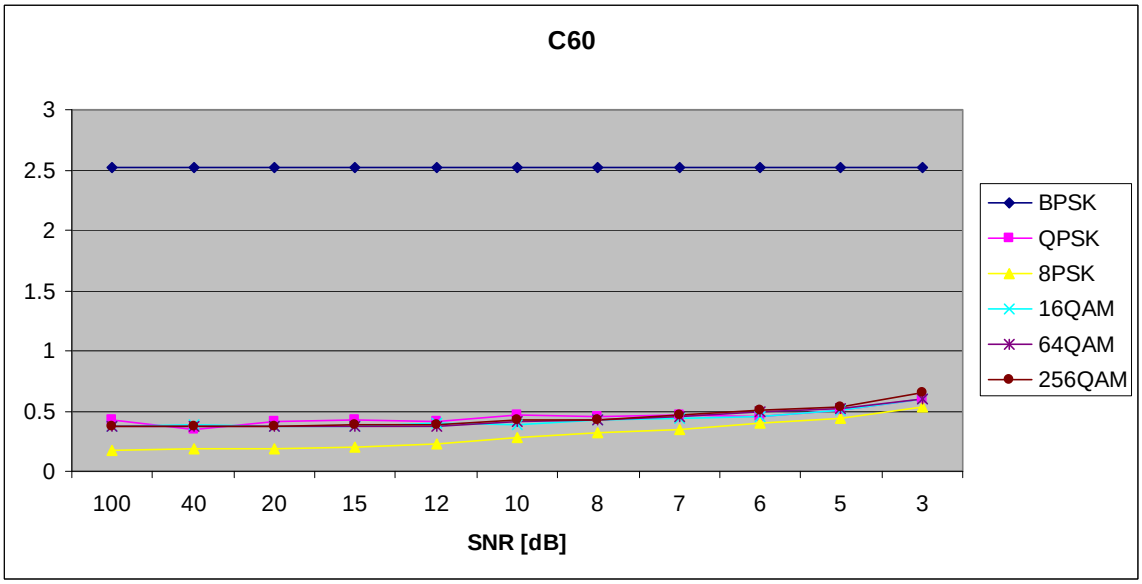

Figure 34. $C_{x,6,0}$ in AWGN.

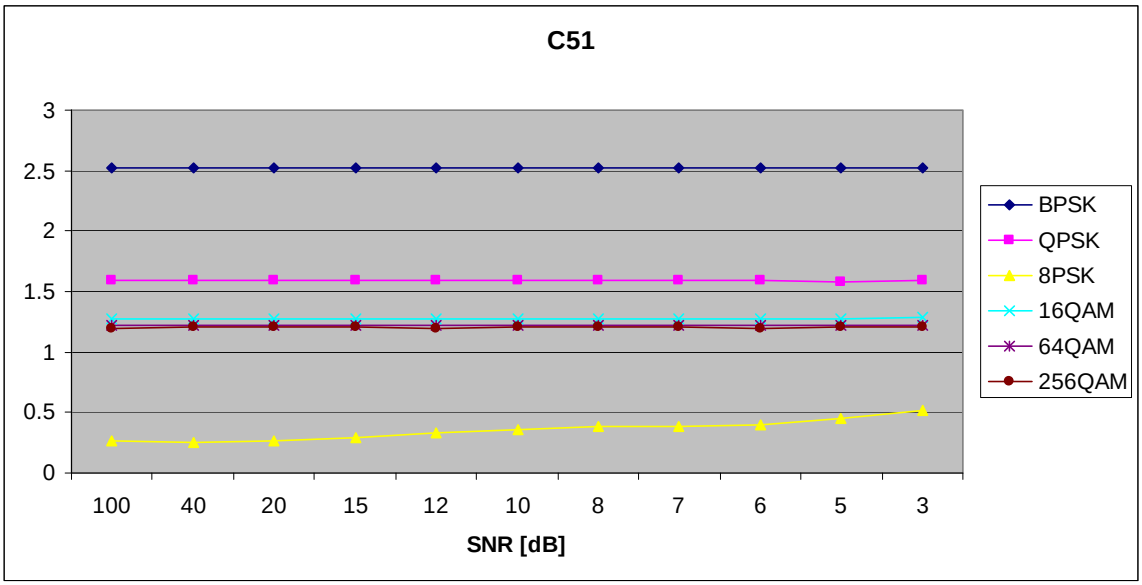

Figure 35. $C_{x,5,1}$ in AWGN.

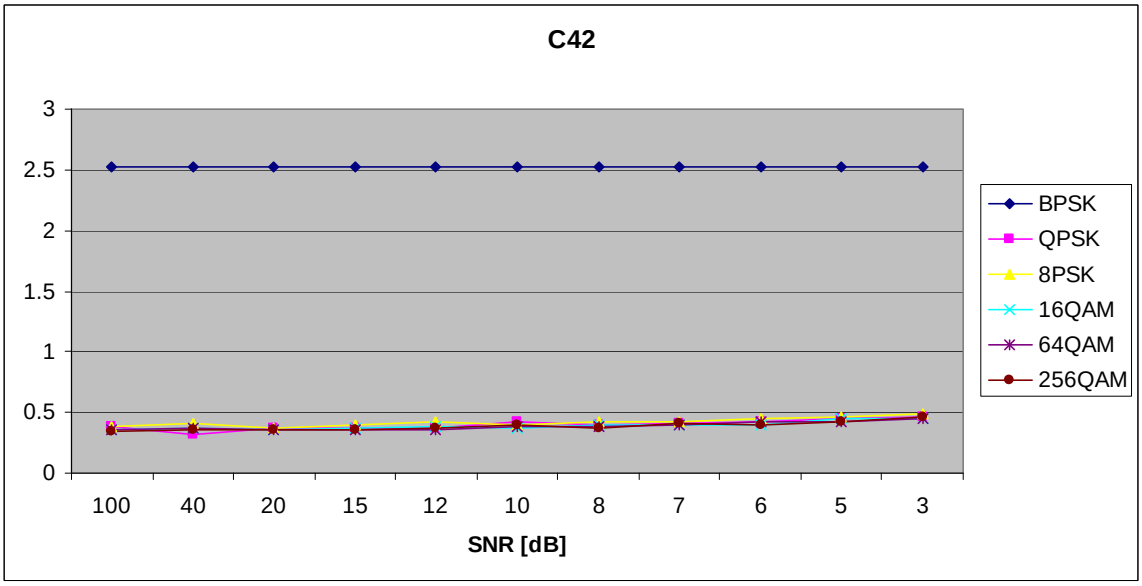

Figure 36. $C_{x,4,2}$ in AWGN.

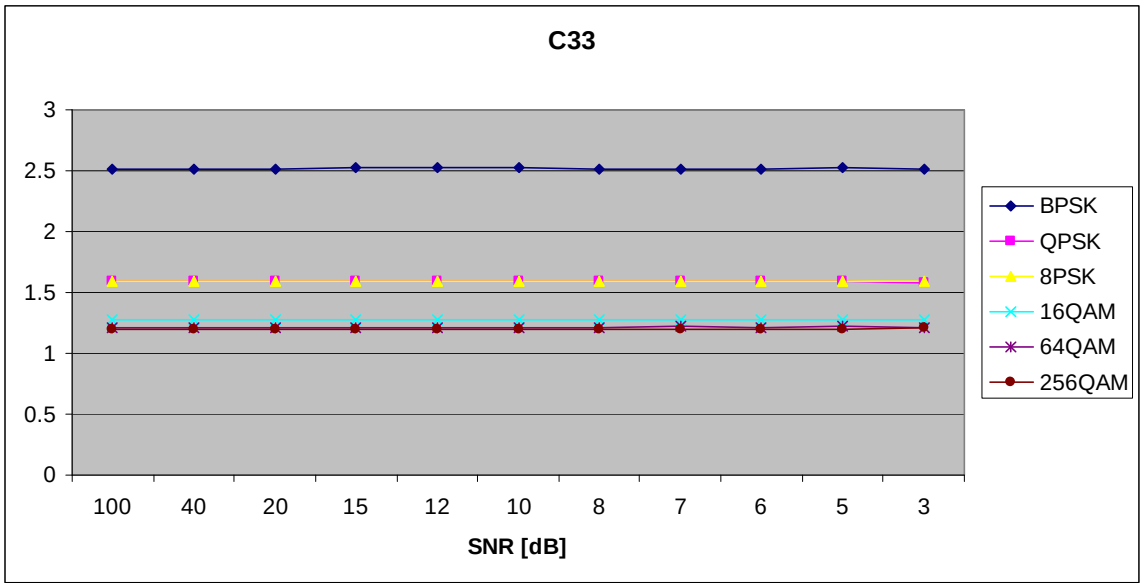

Figure 37. $C_{x,3,3}$ in AWGN.

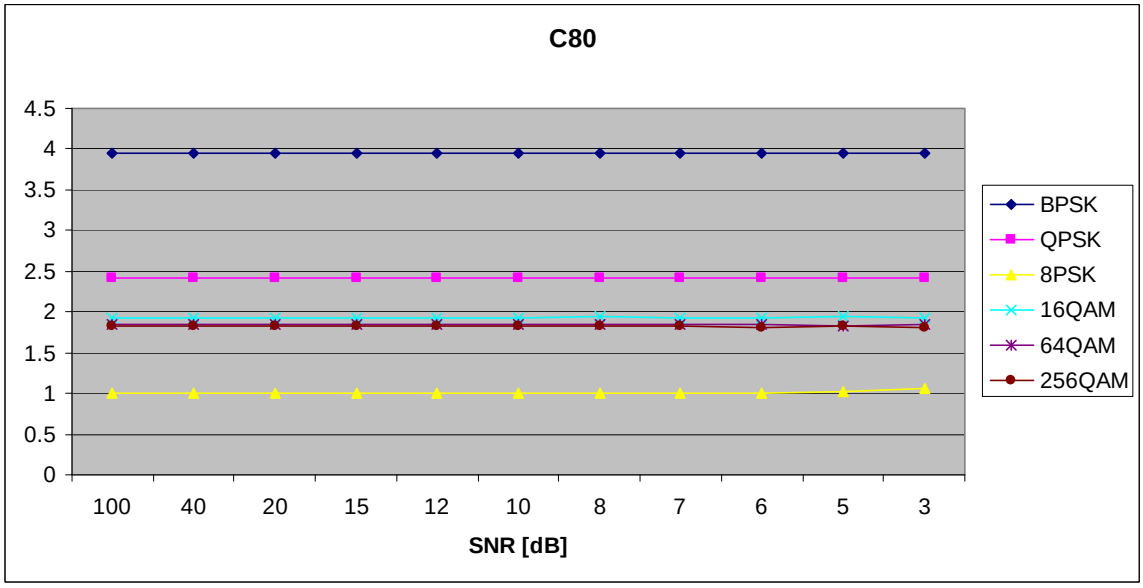

Figure 38. $C_{x,8,0}$ in AWGN.

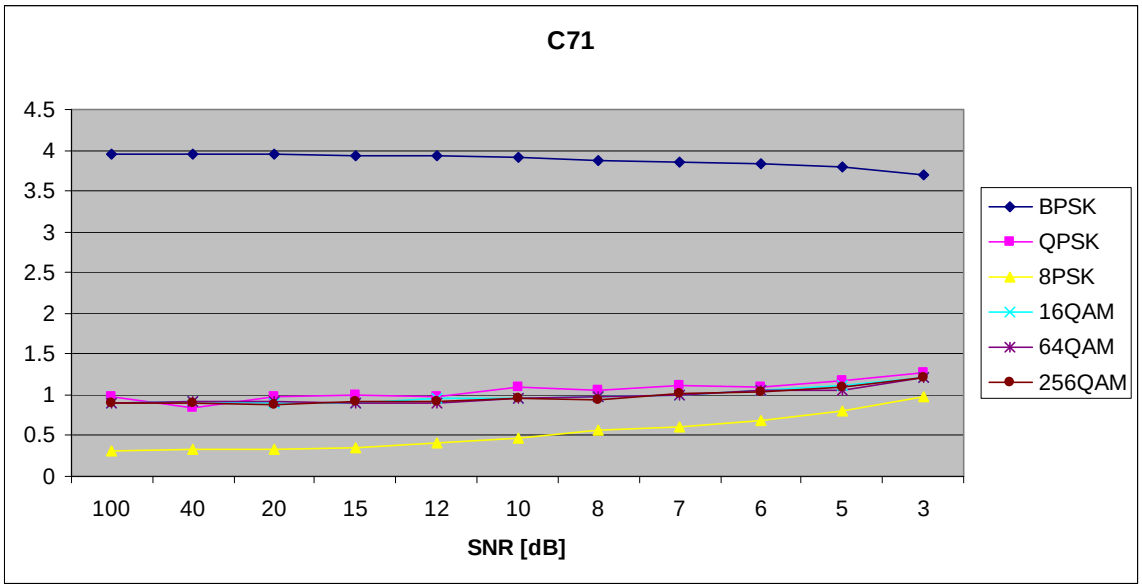

Figure 39. $C_{x,7,1}$ in AWGN.

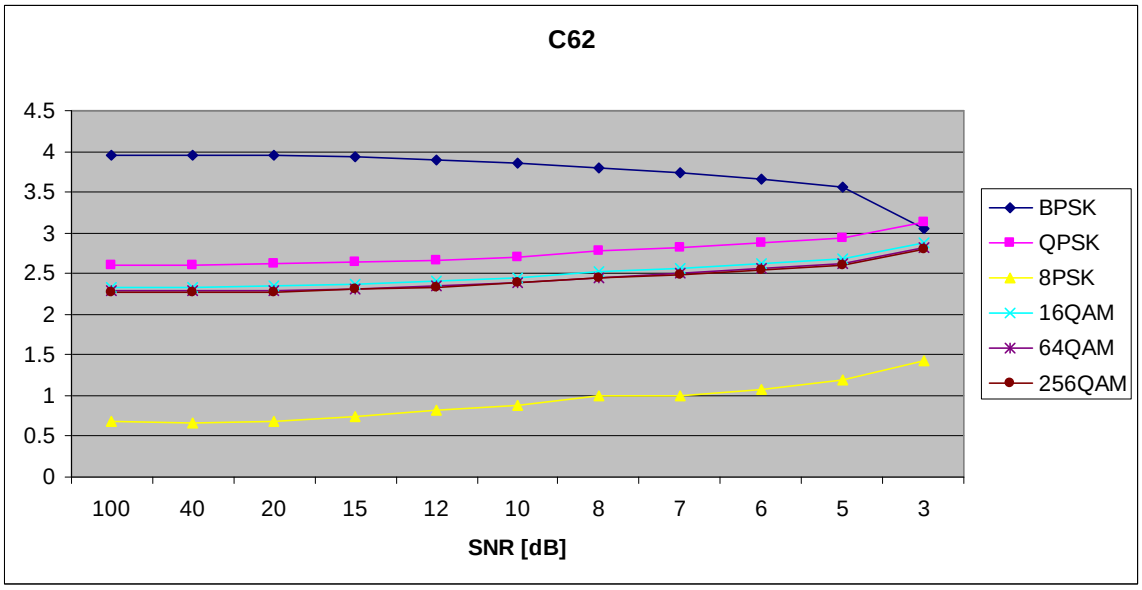

Figure 40. $C_{x,6,2}$ in AWGN.

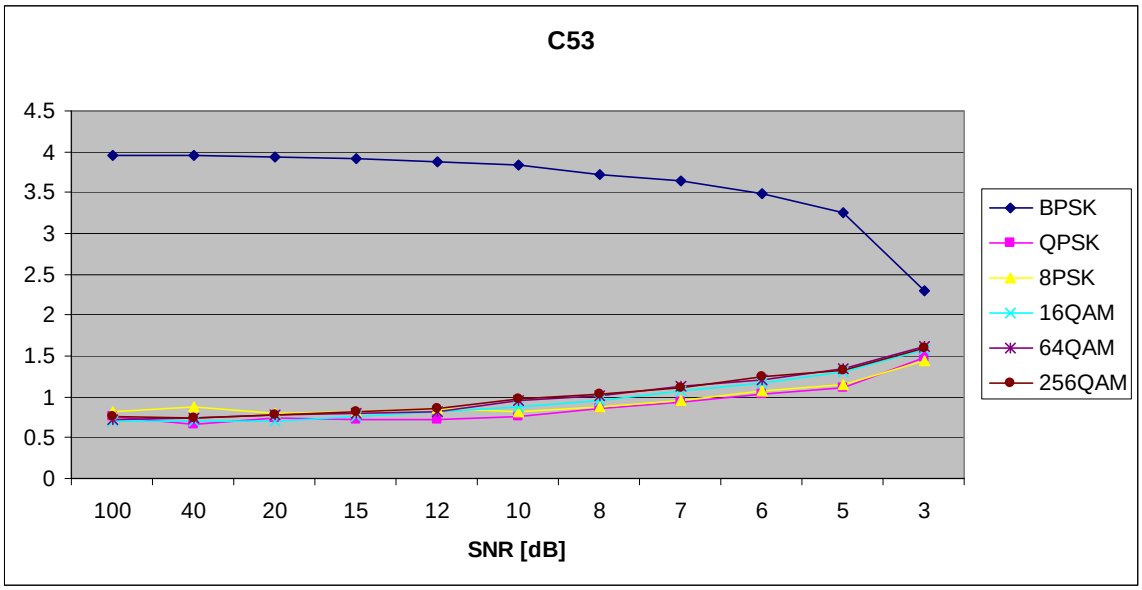

Figure 41. $C_{x,5,3}$ in AWGN.

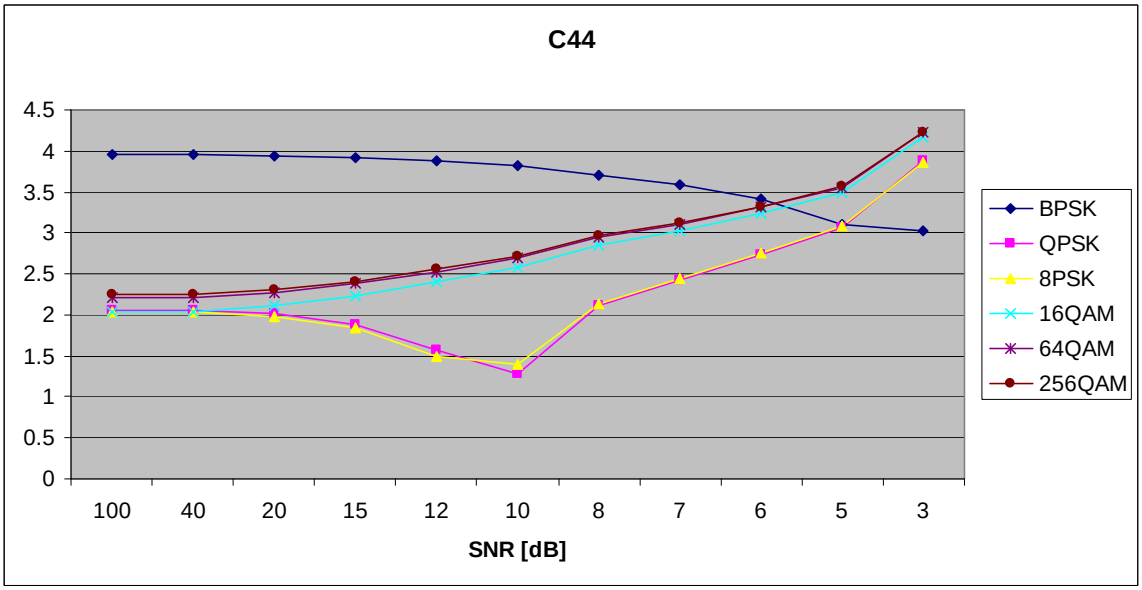

Figure 42. $C_{x,4,4}$ in AWGN.

B. AWGN PLUS SLOW, FREQUENCY-FLAT RAYLEIGH FADING

Parameters for the rayleighchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 3.5 Hz
- Path Delays: [0, 1×10^{-7}]
- Average Path Gains: [0, -10]

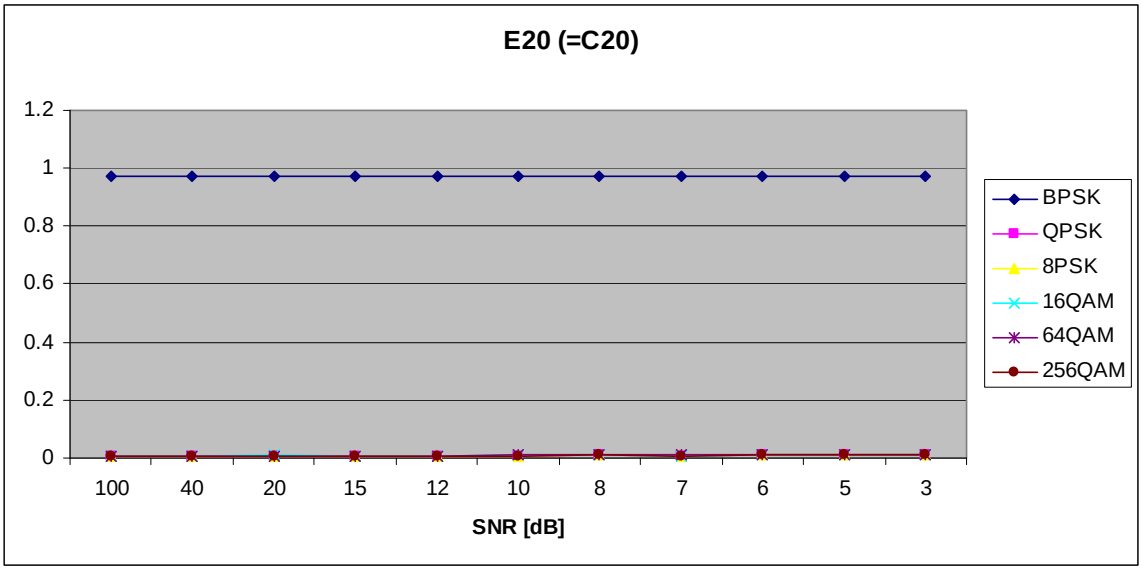

Figure 43. $E_{x,2,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

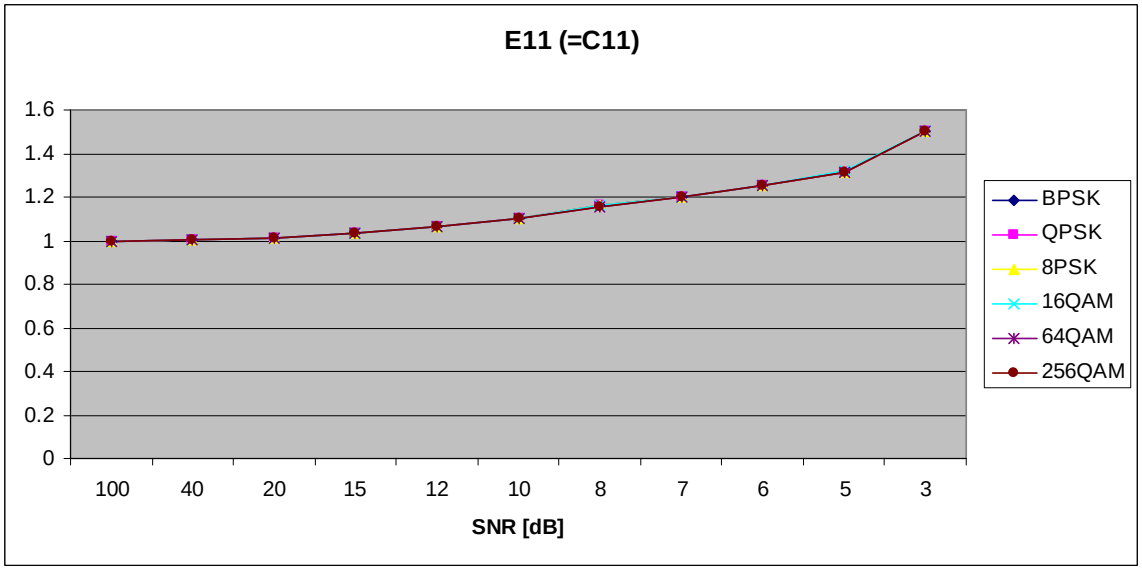

Figure 44. $E_{x,1,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

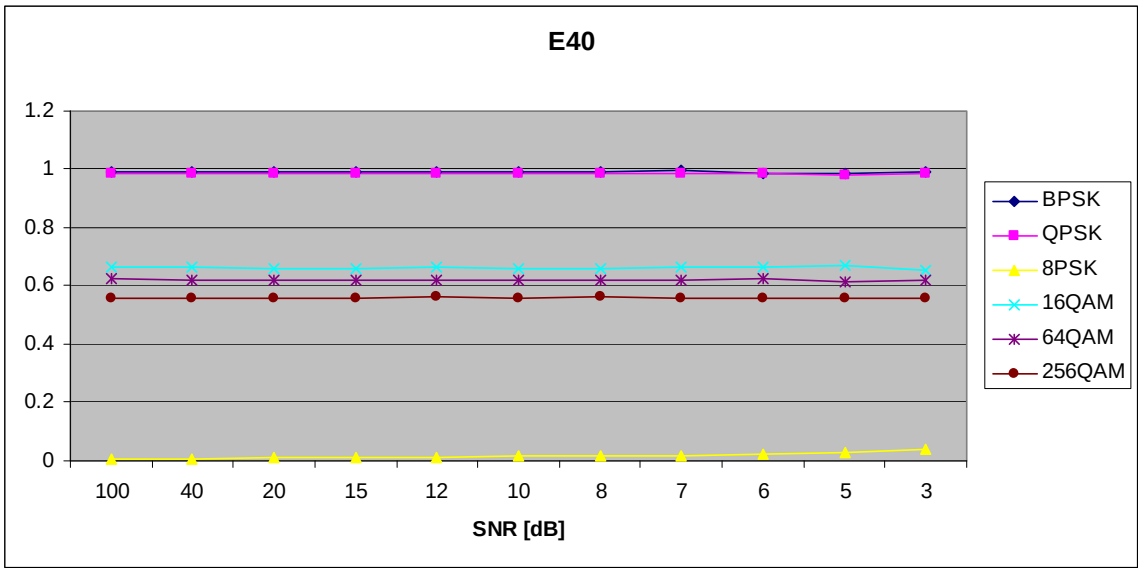

Figure 45. $E_{x,4,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

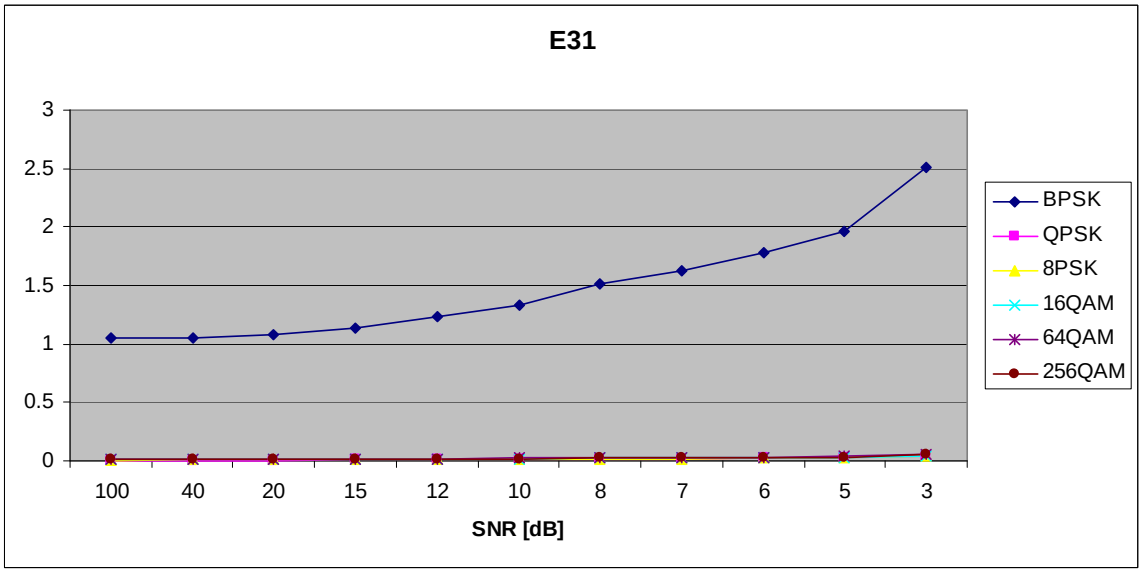

Figure 46. $E_{x,3,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

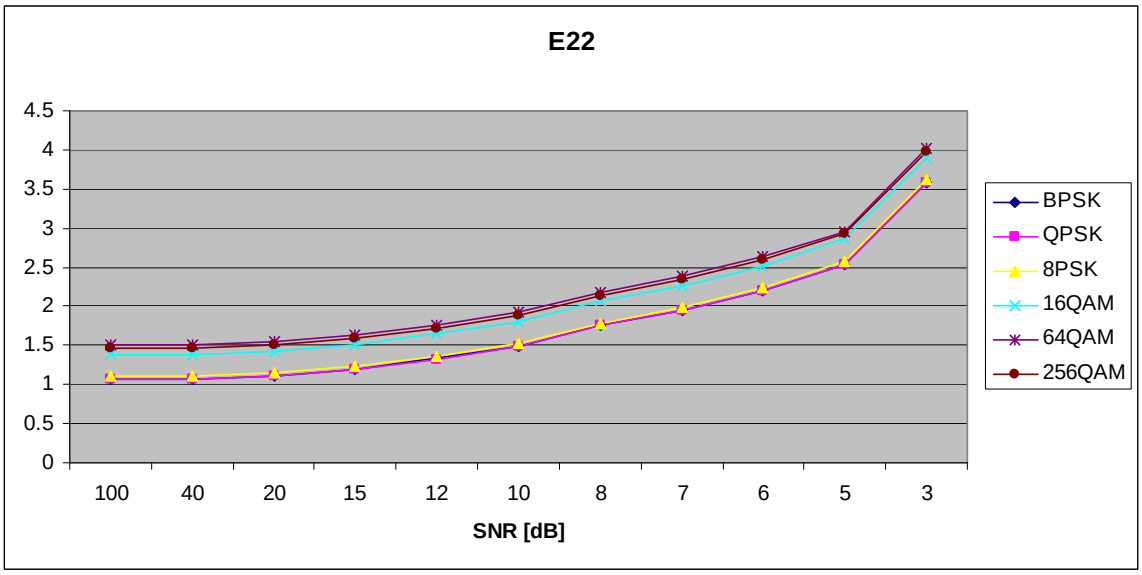

Figure 47. $E_{x,2,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

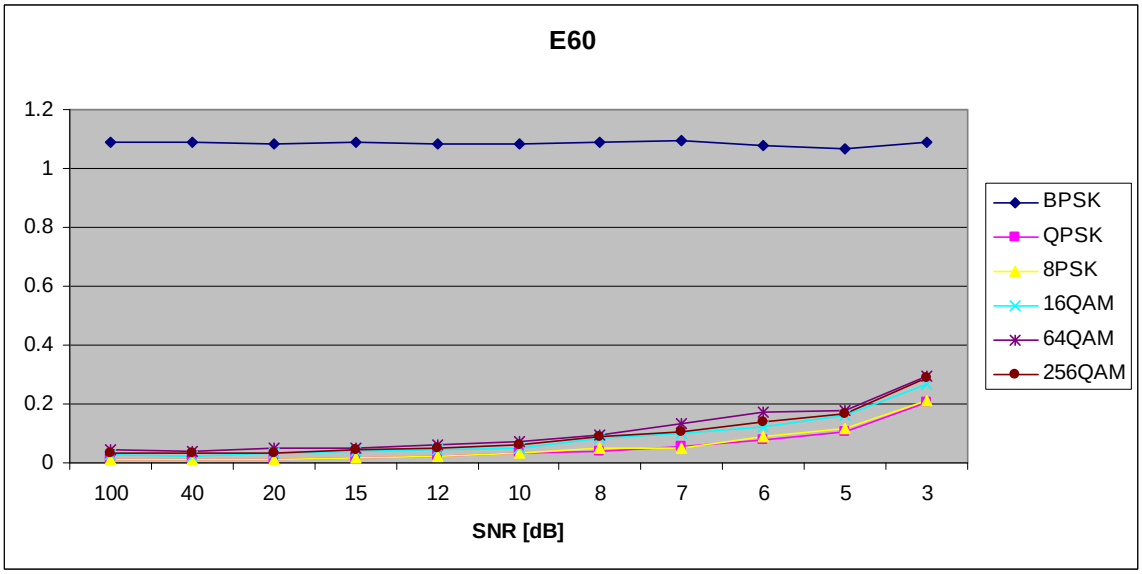

Figure 48. $E_{x,6,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

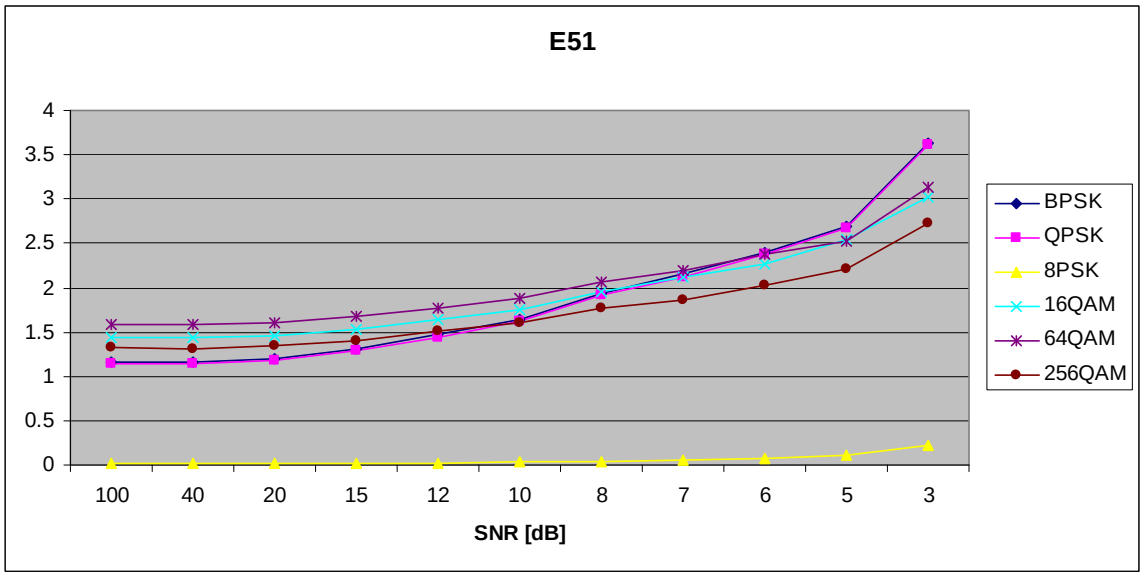

Figure 49. $E_{x,5,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

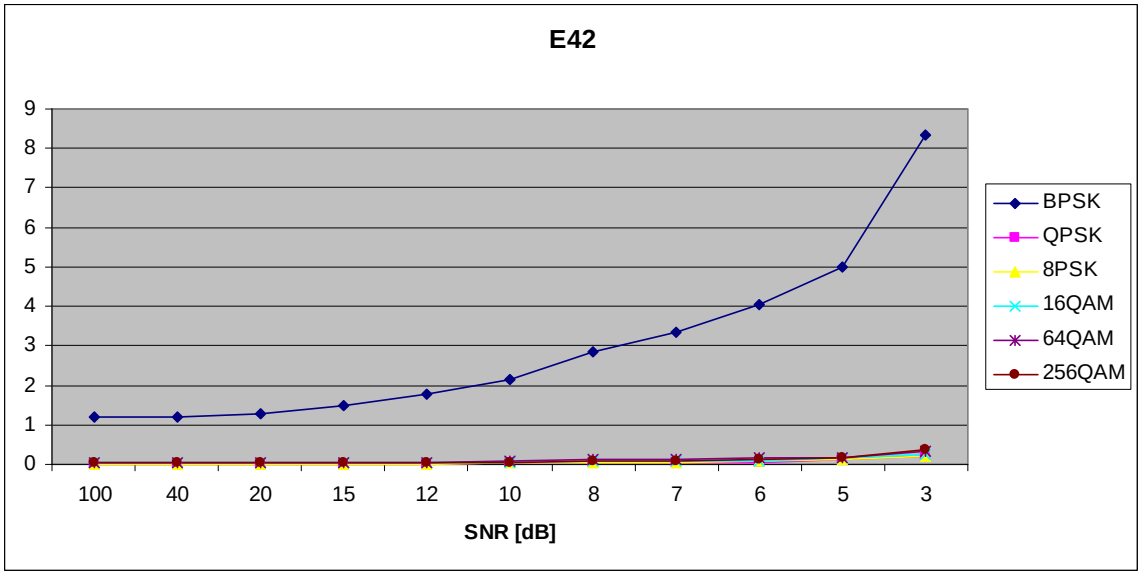

Figure 50. $E_{x,4,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

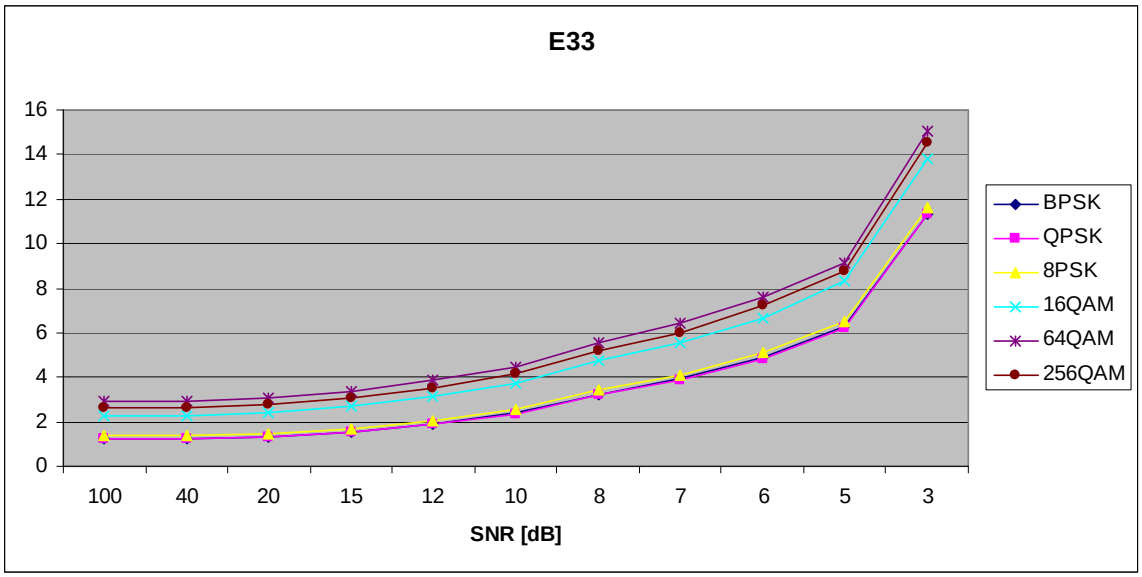

Figure 51. $E_{x,3,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

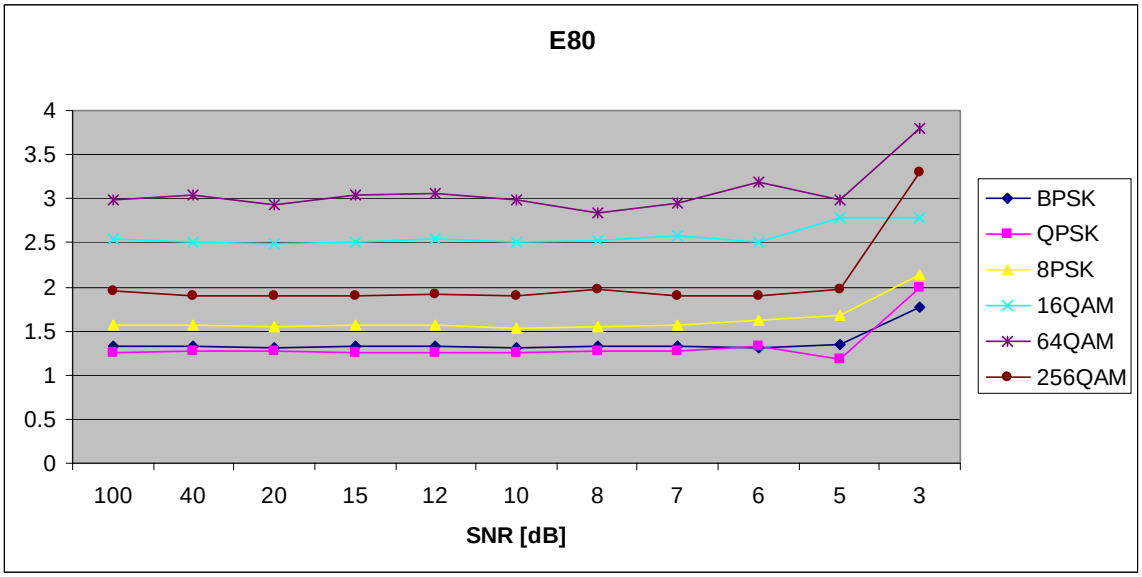

Figure 52. $E_{x,8,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

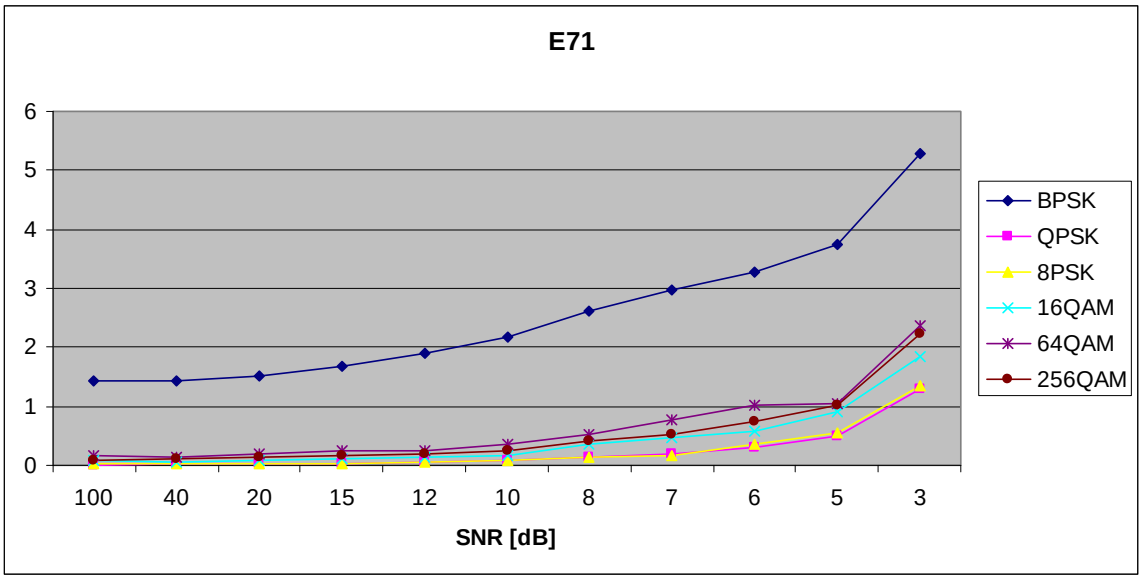

Figure 53. $E_{x,7,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

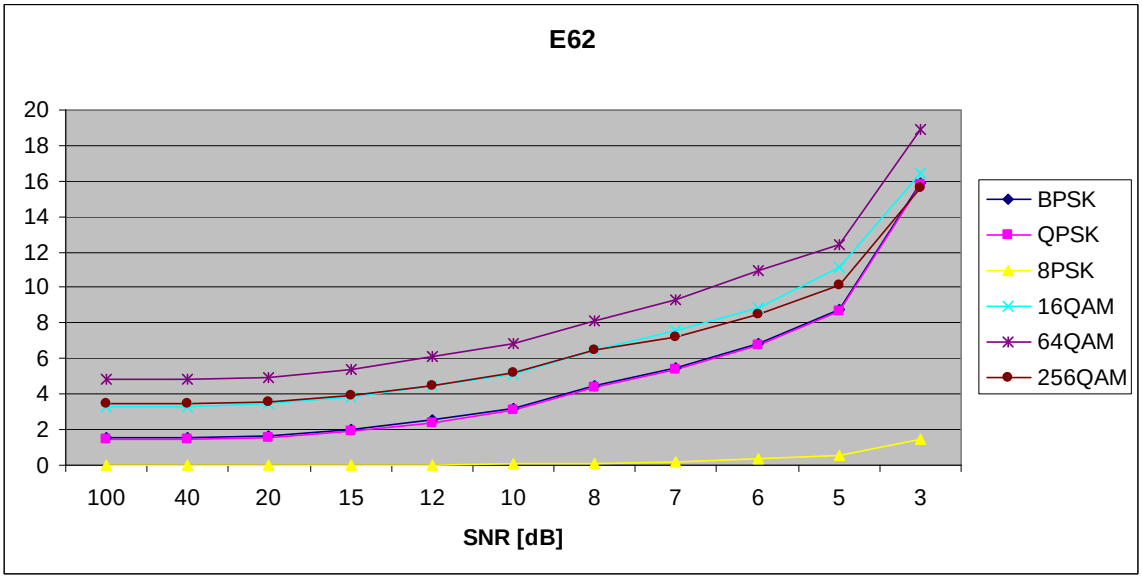

Figure 54. $E_{x,6,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

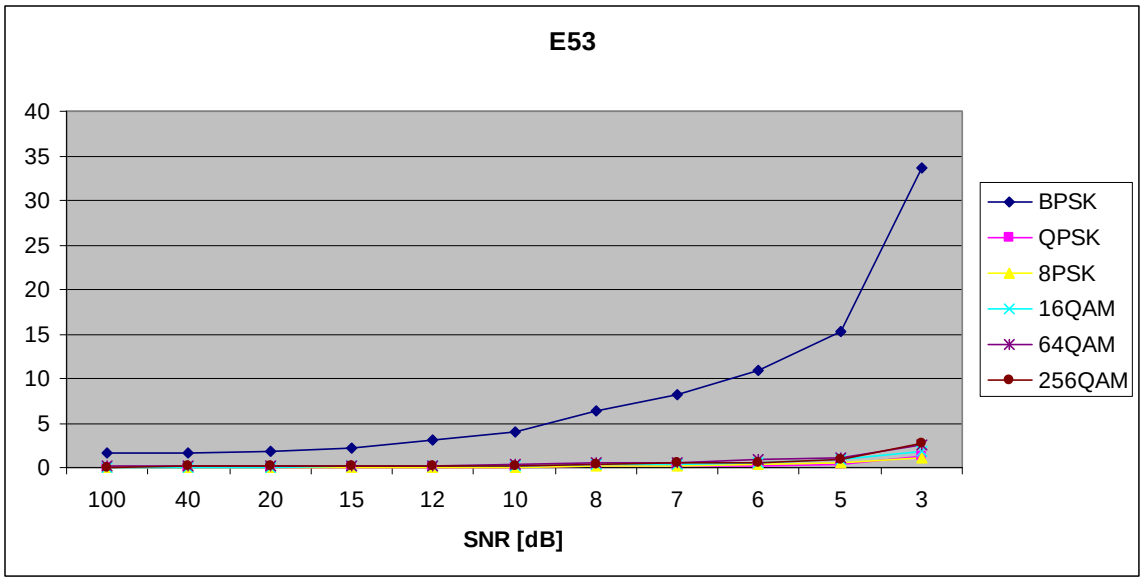

Figure 55. $E_{x,5,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

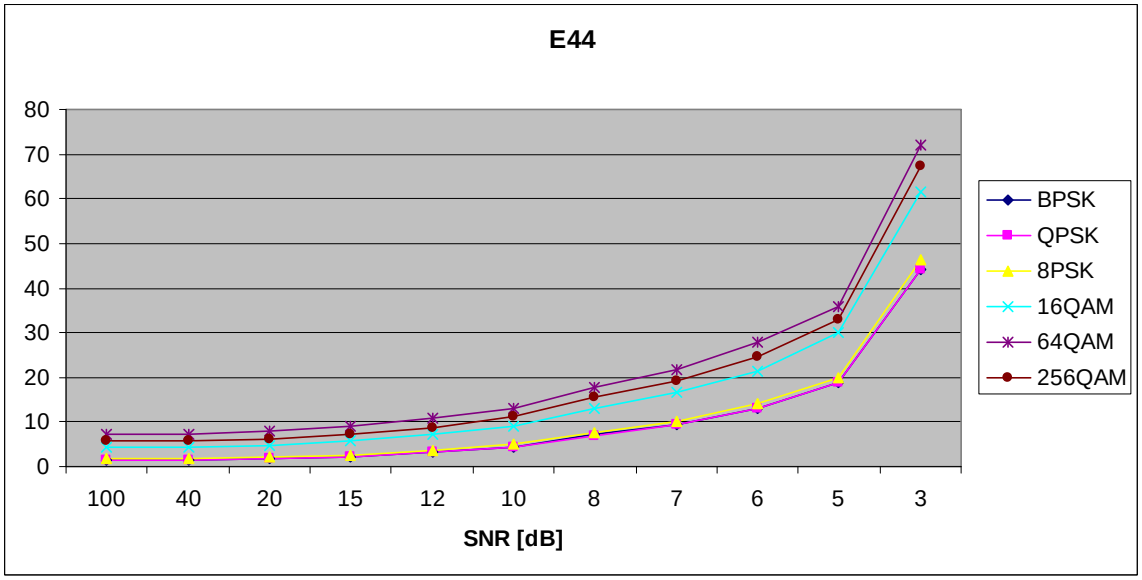

Figure 56. $E_{x,4,4}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

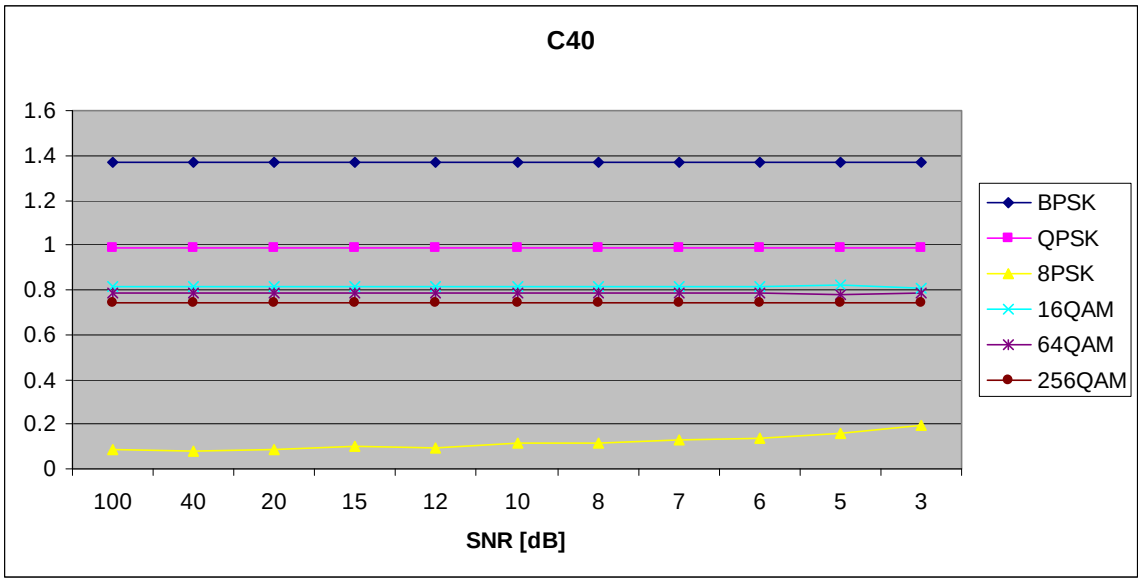

Figure 57. $C_{x,4,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

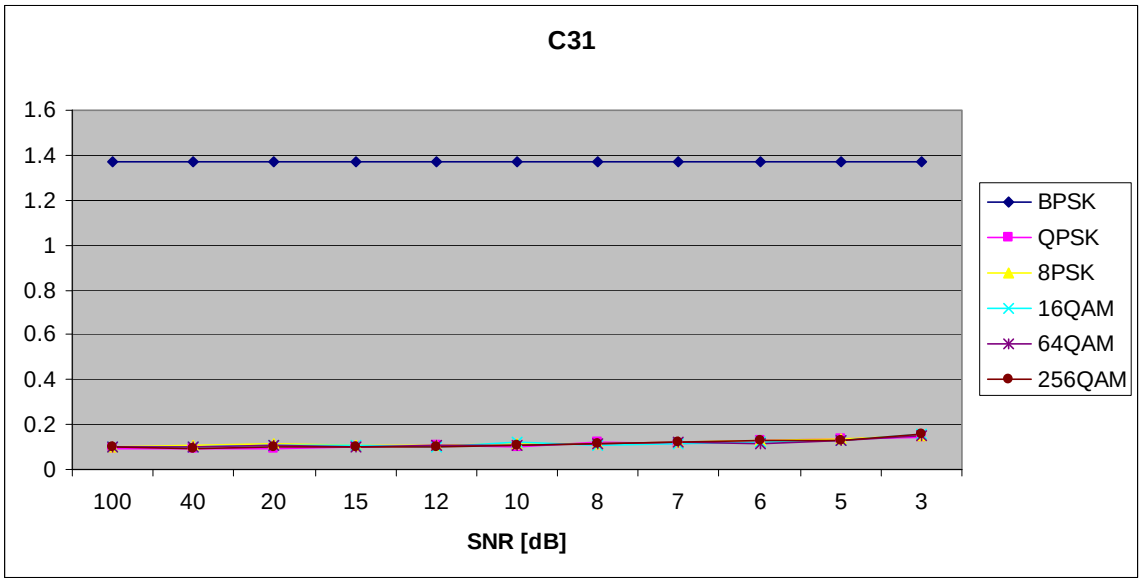

Figure 58. $C_{x,3,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

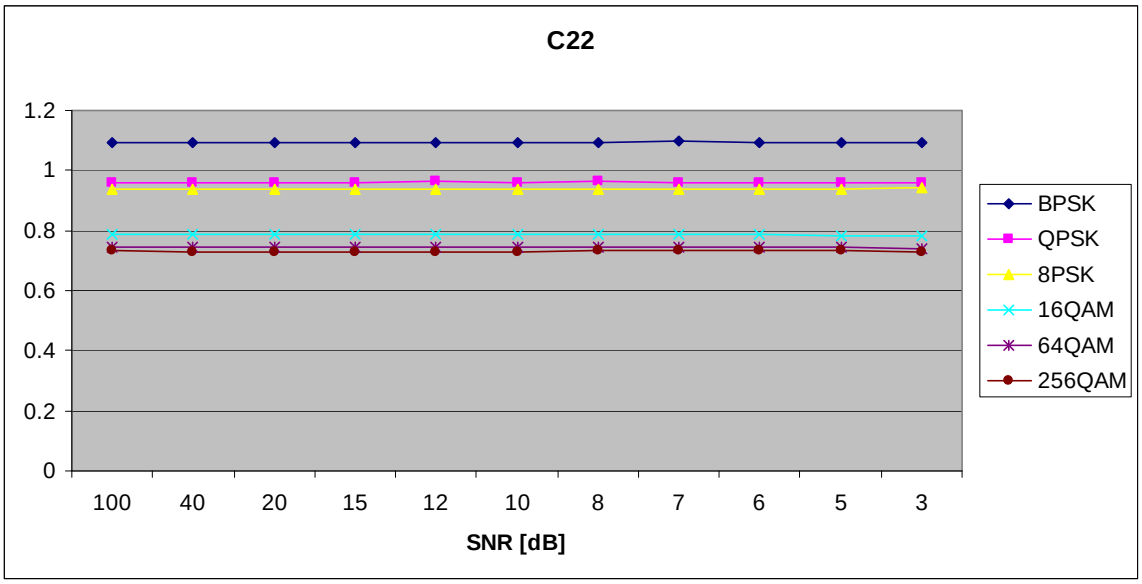

Figure 59. $C_{x,2,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

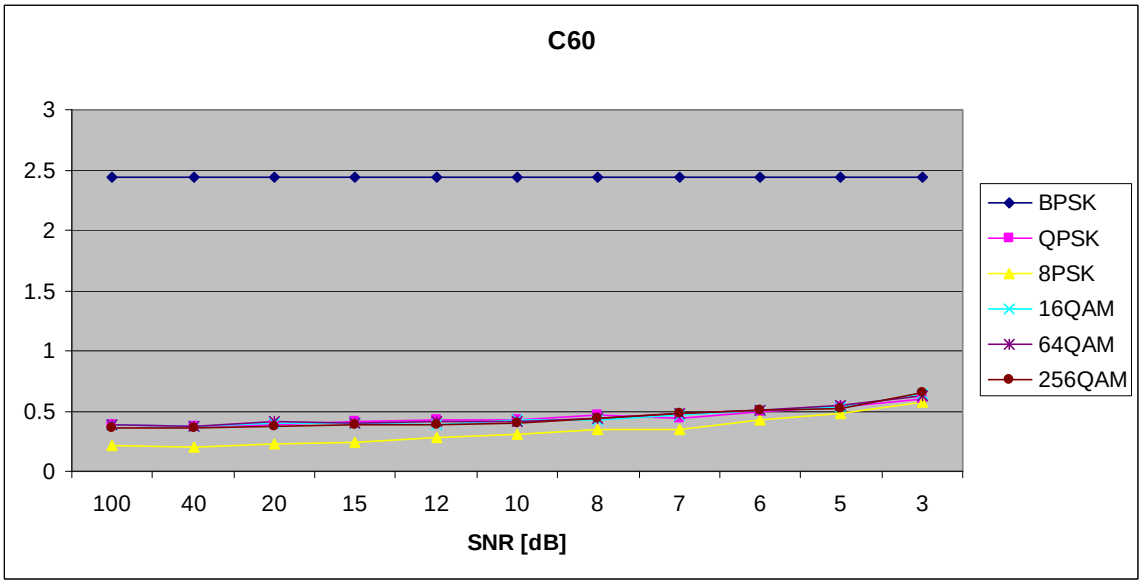

Figure 60. $C_{x,6,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

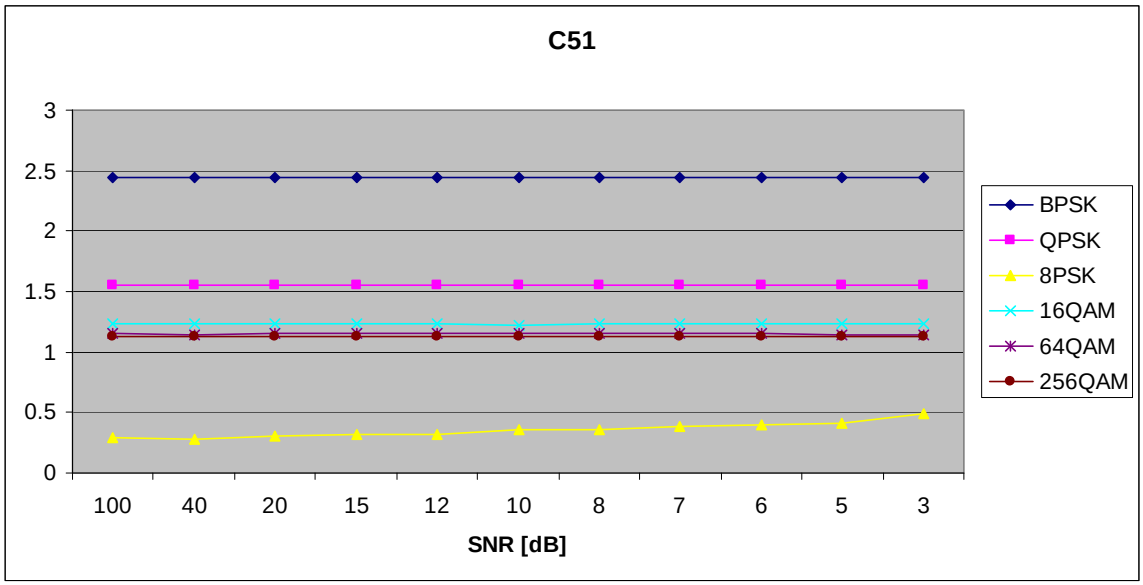

Figure 61. $C_{x,5,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

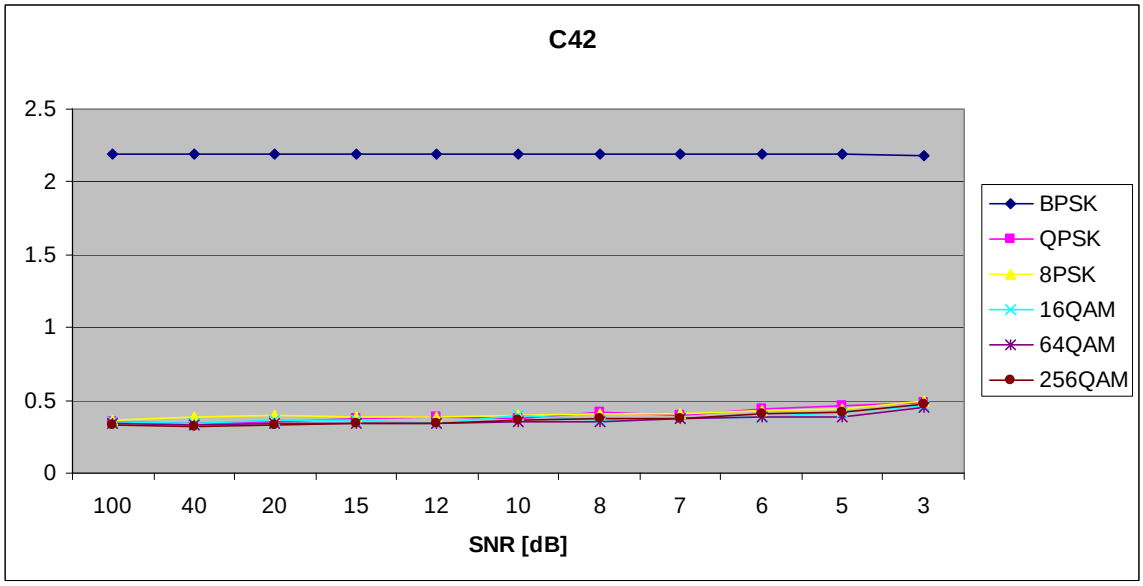

Figure 62. $C_{x,4,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

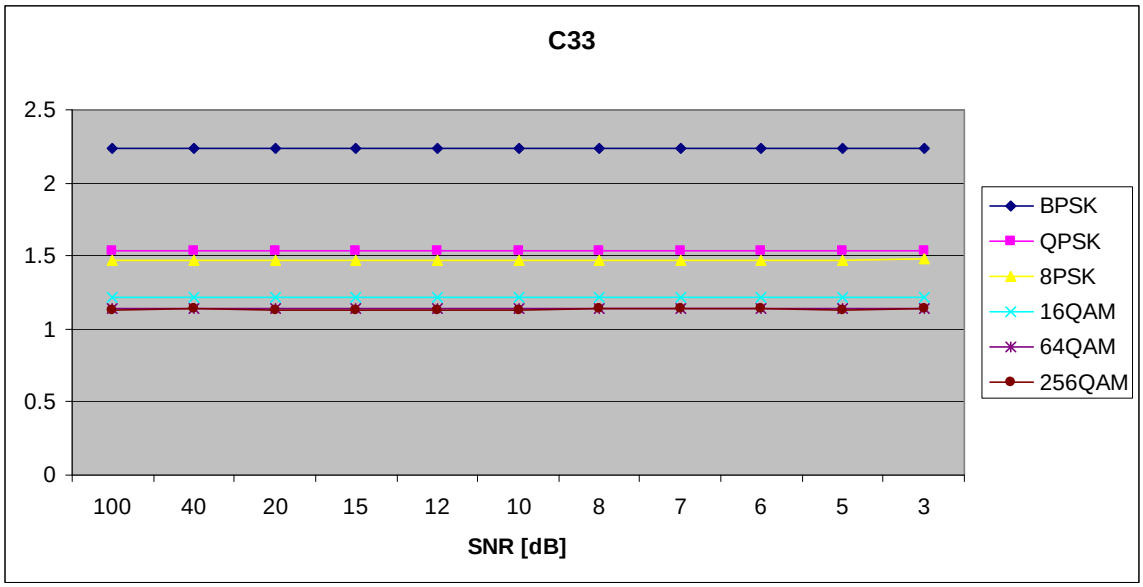

Figure 63. $C_{x,3,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

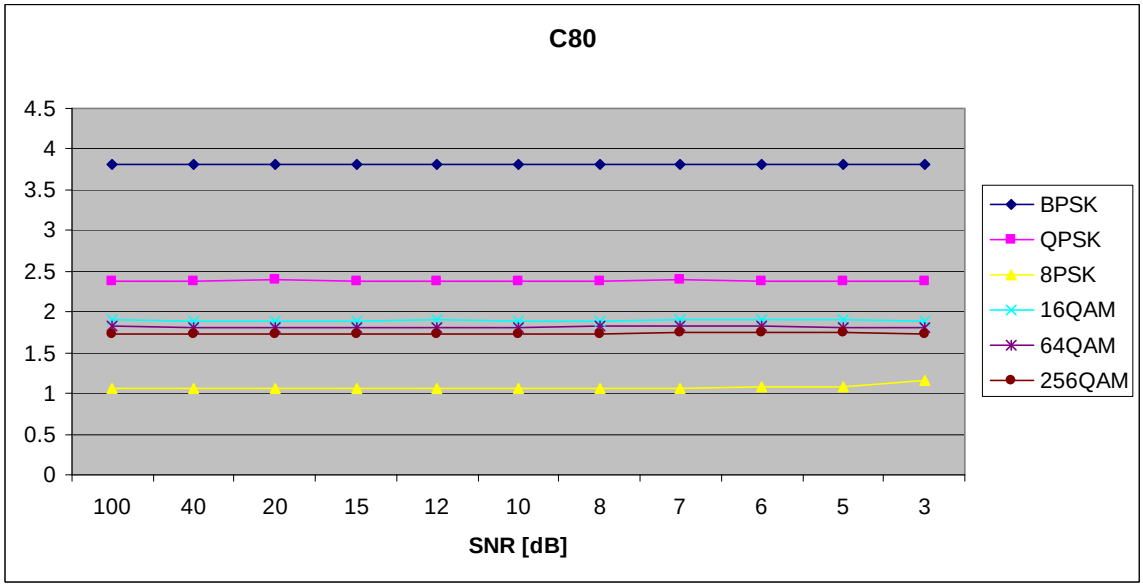

Figure 64. $C_{x,8,0}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

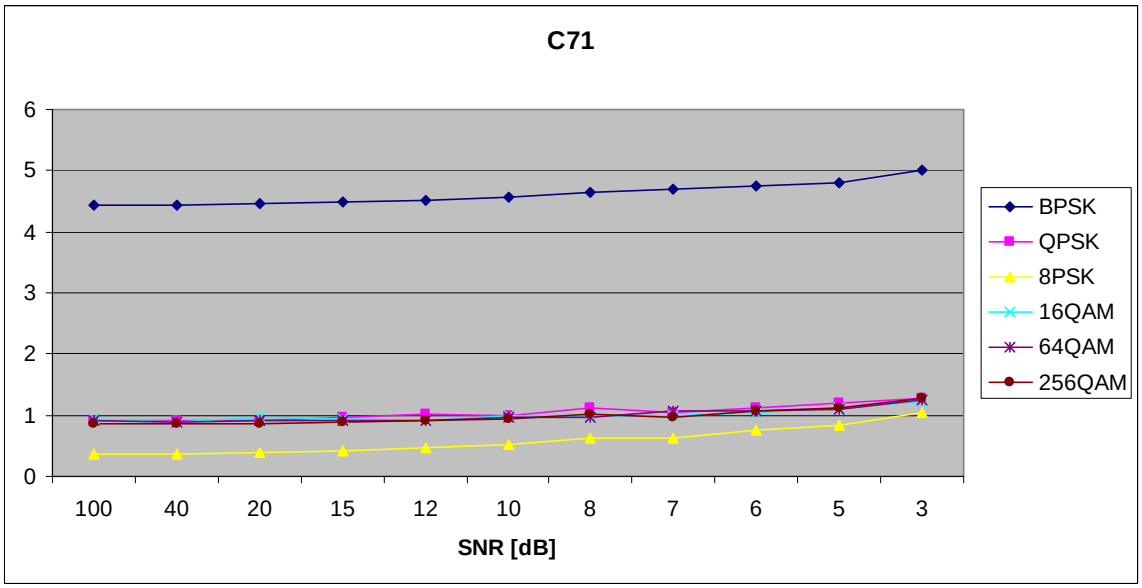

Figure 65. $C_{x,7,1}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

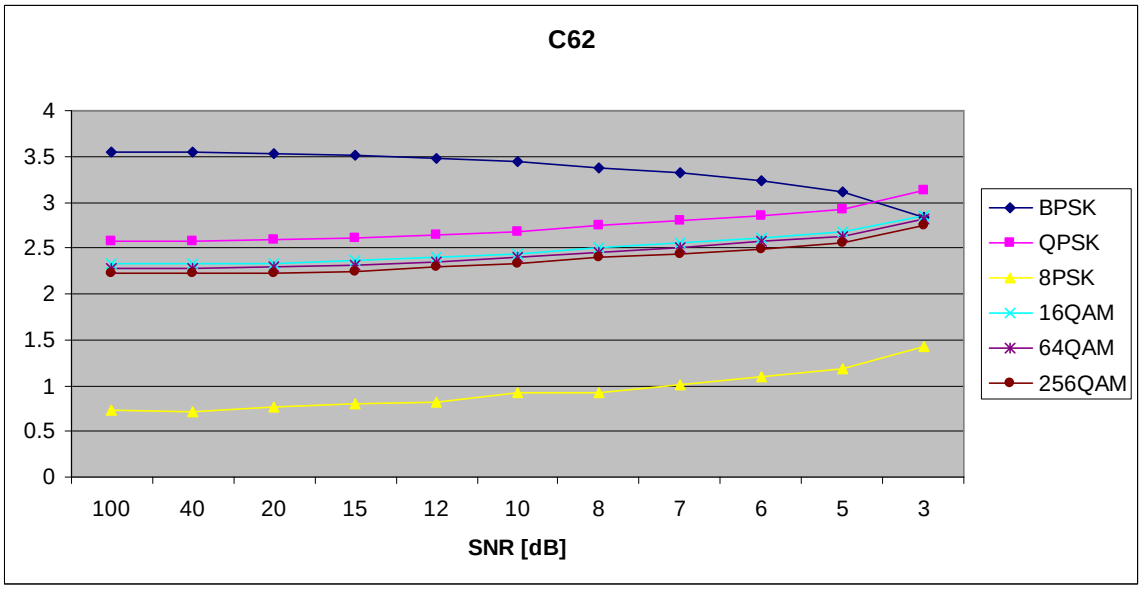

Figure 66. $C_{x,6,2}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

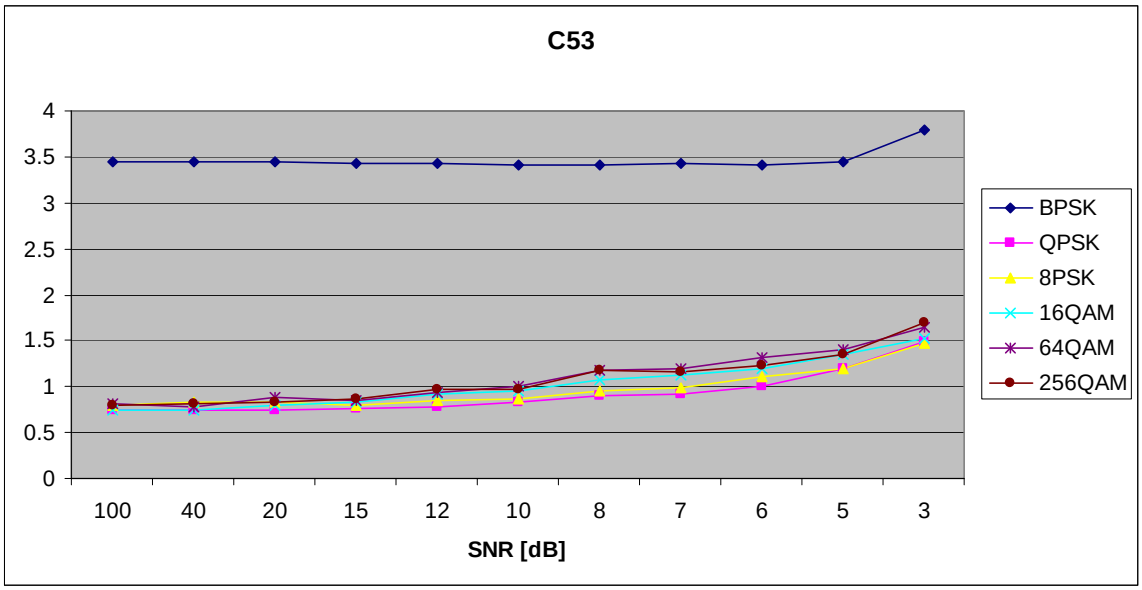

Figure 67. $C_{x,5,3}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

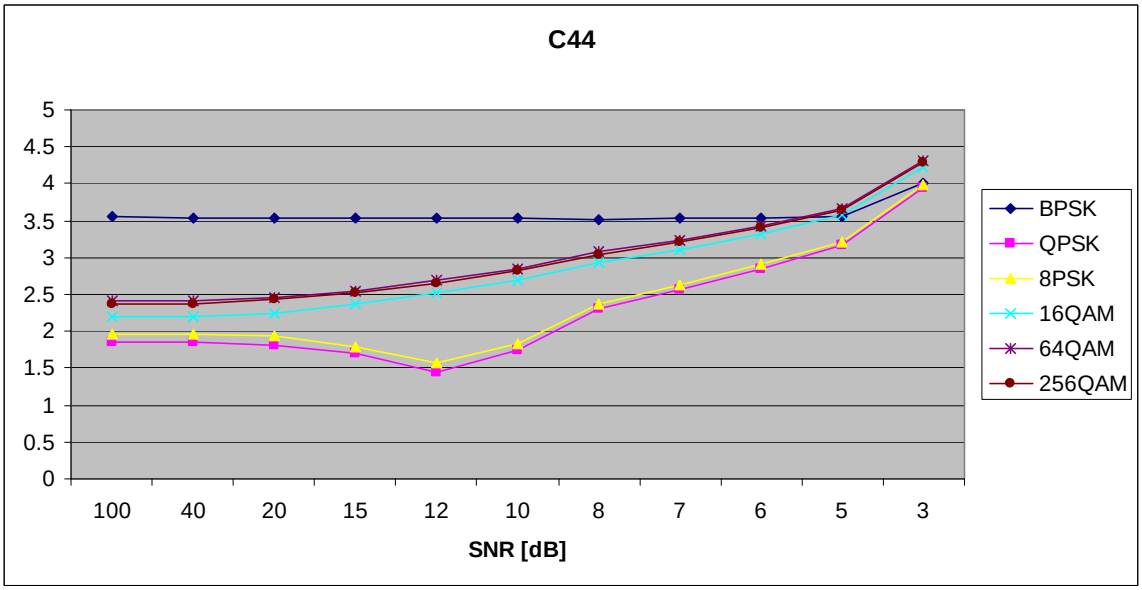

Figure 68. $C_{x,4,4}$ in AWGN and Slow, Frequency-Flat Rayleigh Fading.

C. AWGN PLUS SLOW, FREQUENCY-FLAT RICEAN FADING

Parameters for the ricianchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 3.5 Hz
- K-factor: 3
- Path Delays: [0, 1×10^{-7}]
- Average Path Gains: [0, -10]

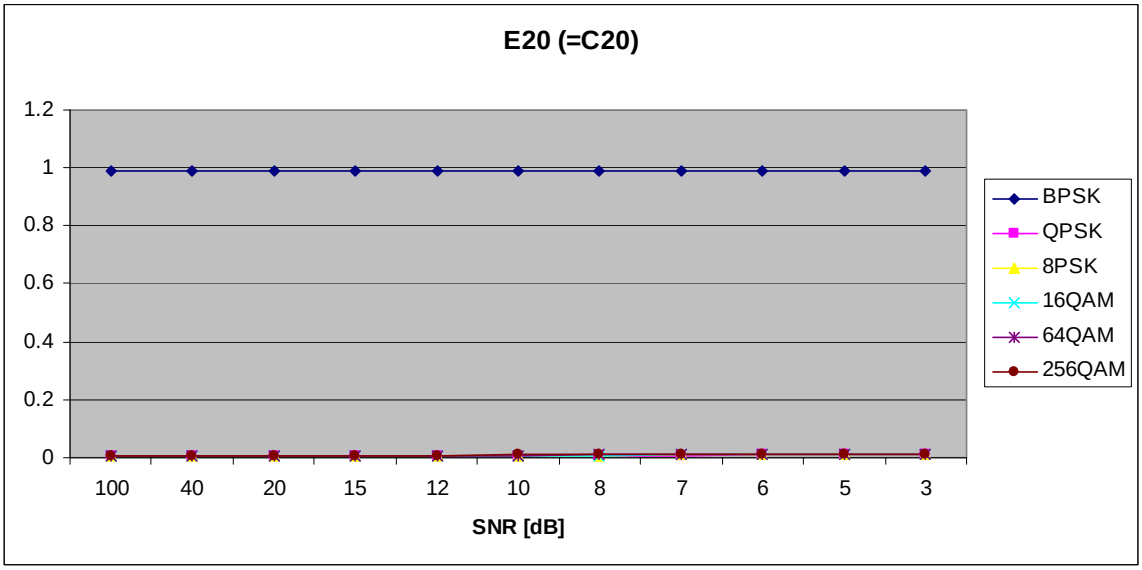

Figure 69. $E_{x,2,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

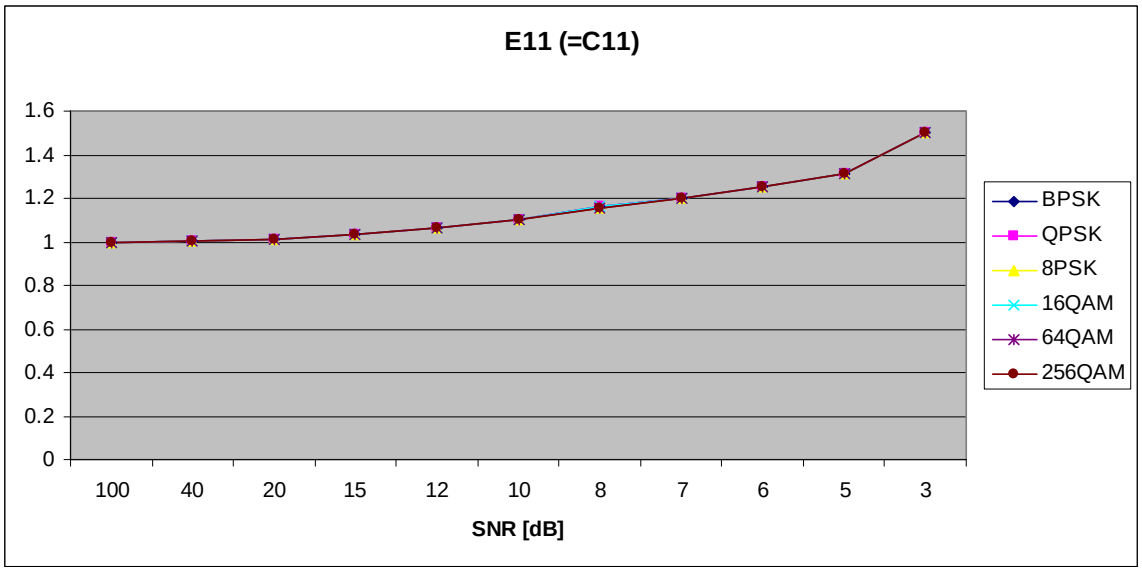

Figure 70. $E_{x,1,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

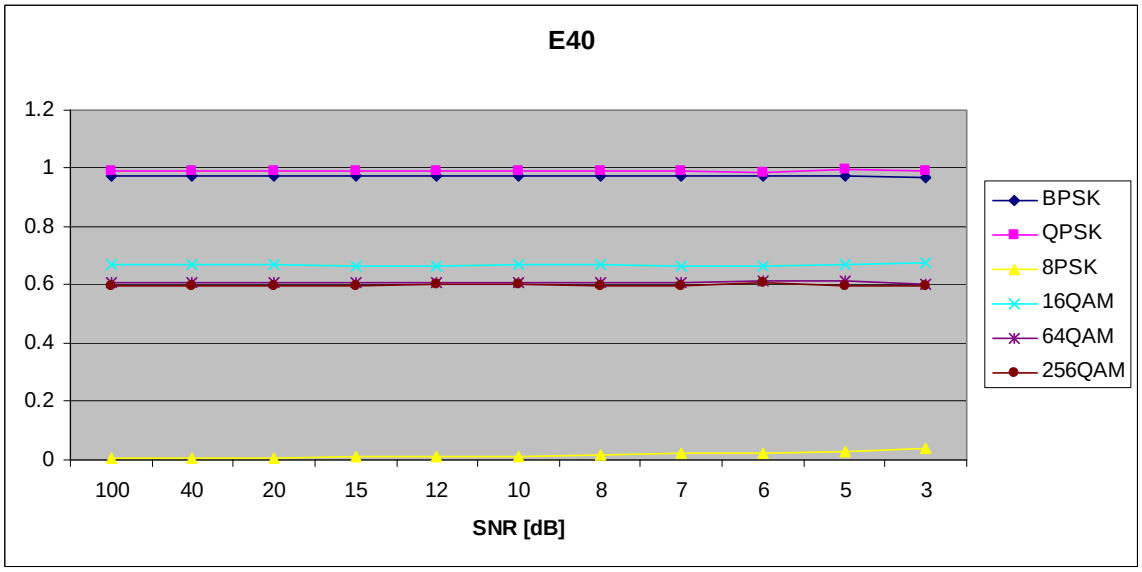

Figure 71. $E_{x,4,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

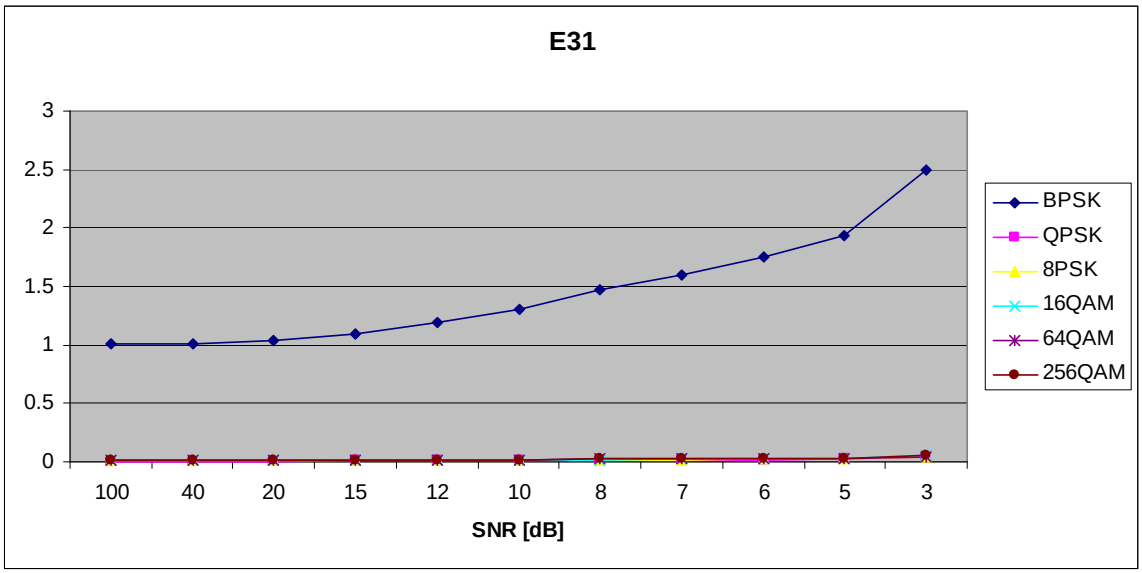

Figure 72. $E_{x,3,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

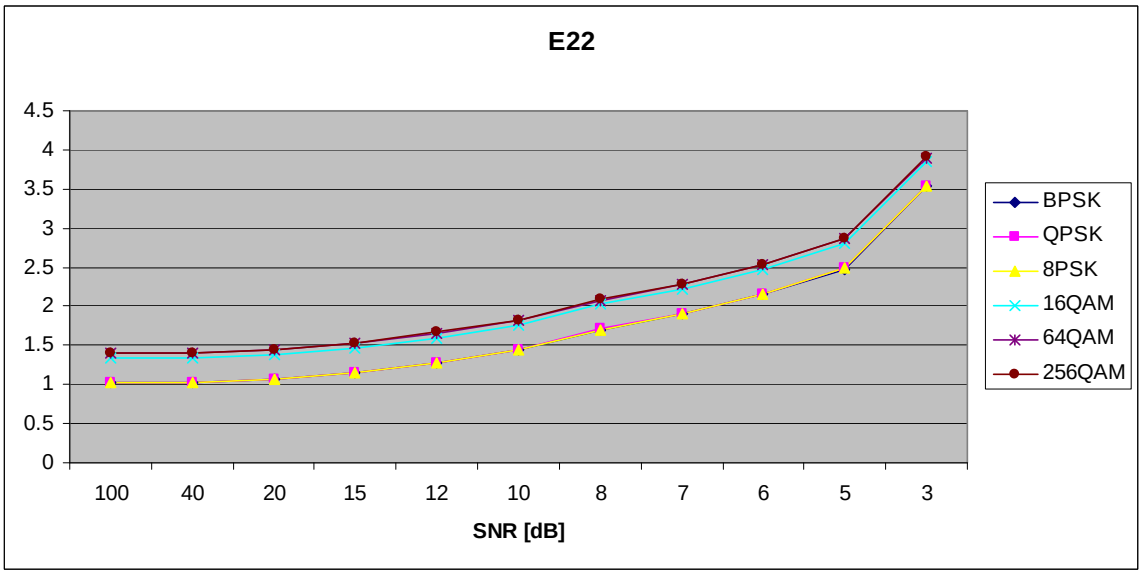

Figure 73. $E_{x,2,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

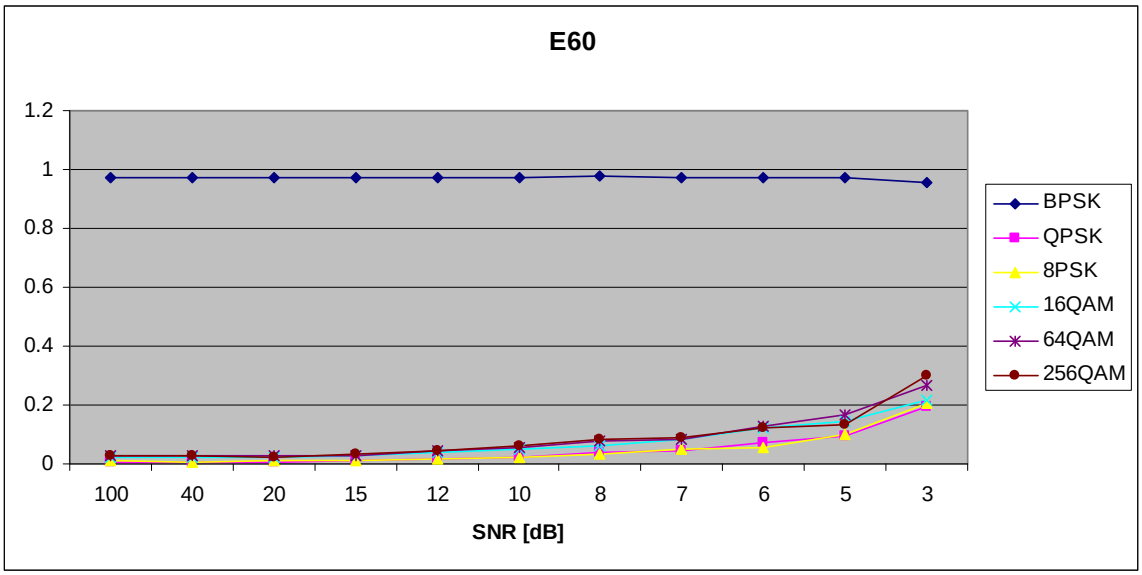

Figure 74. $E_{x,6,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

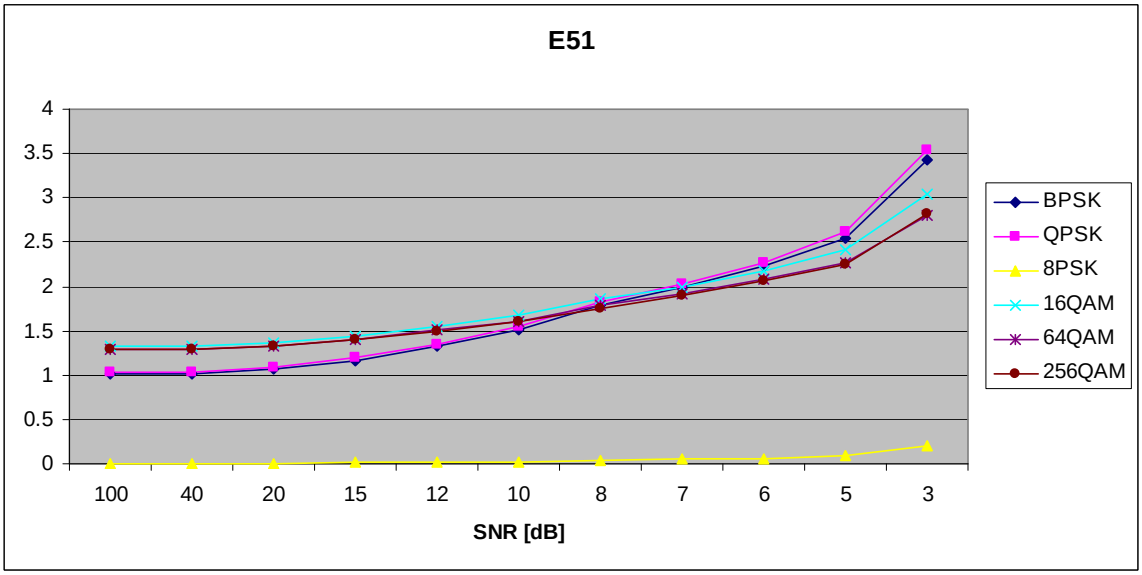

Figure 75. $E_{x,5,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

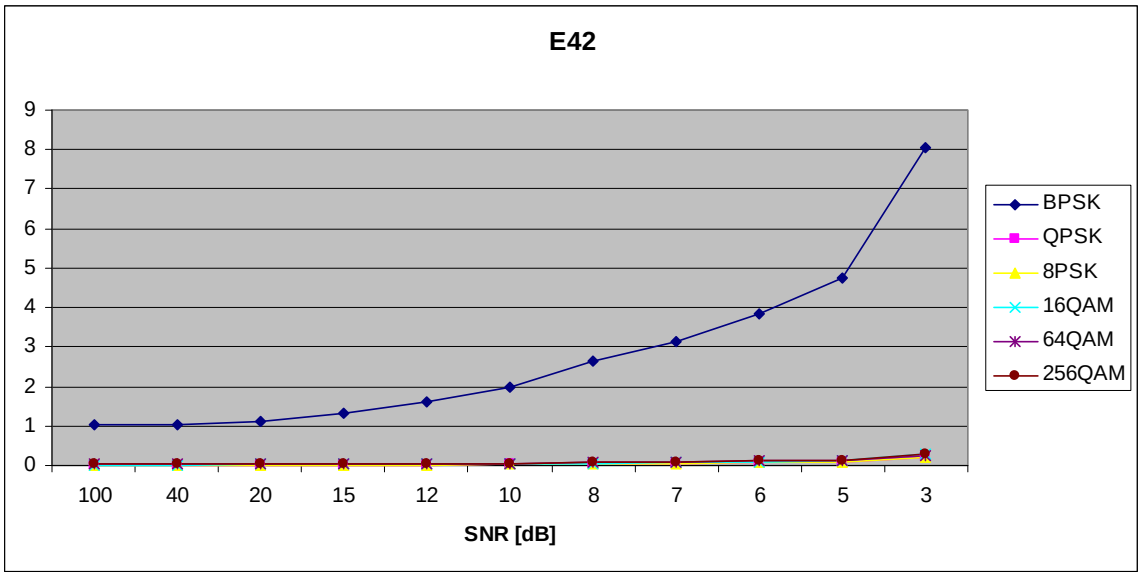

Figure 76. $E_{x,4,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

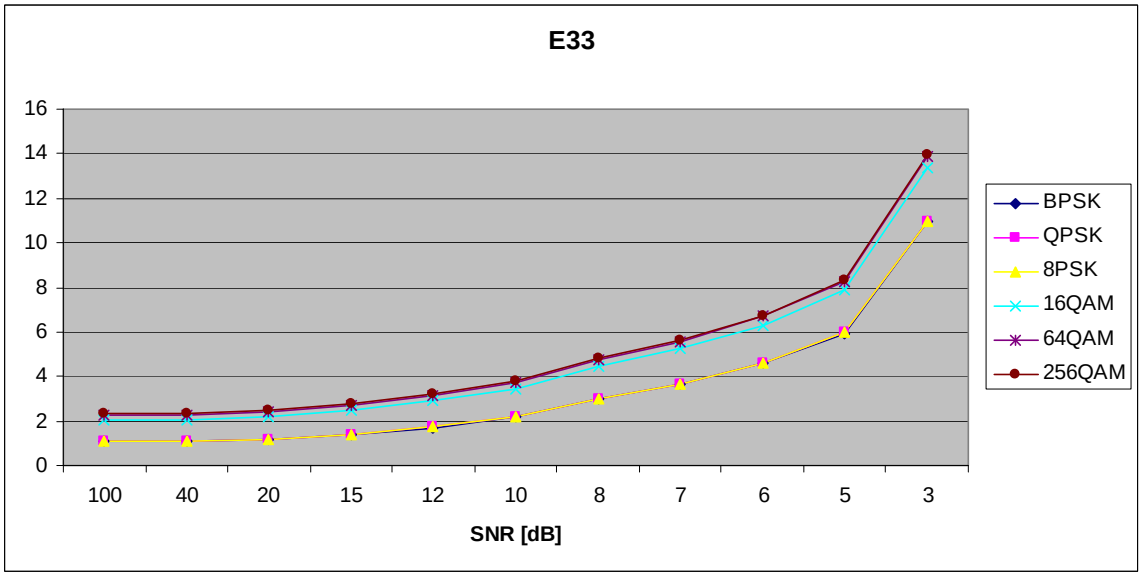

Figure 77. $E_{x,3,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

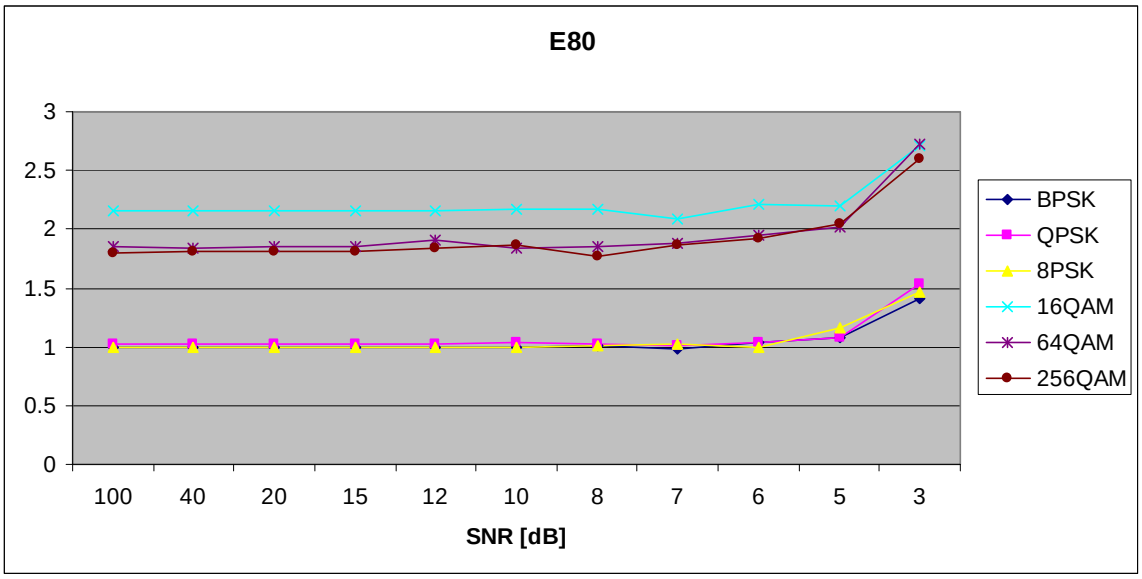

Figure 78. $E_{x,8,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

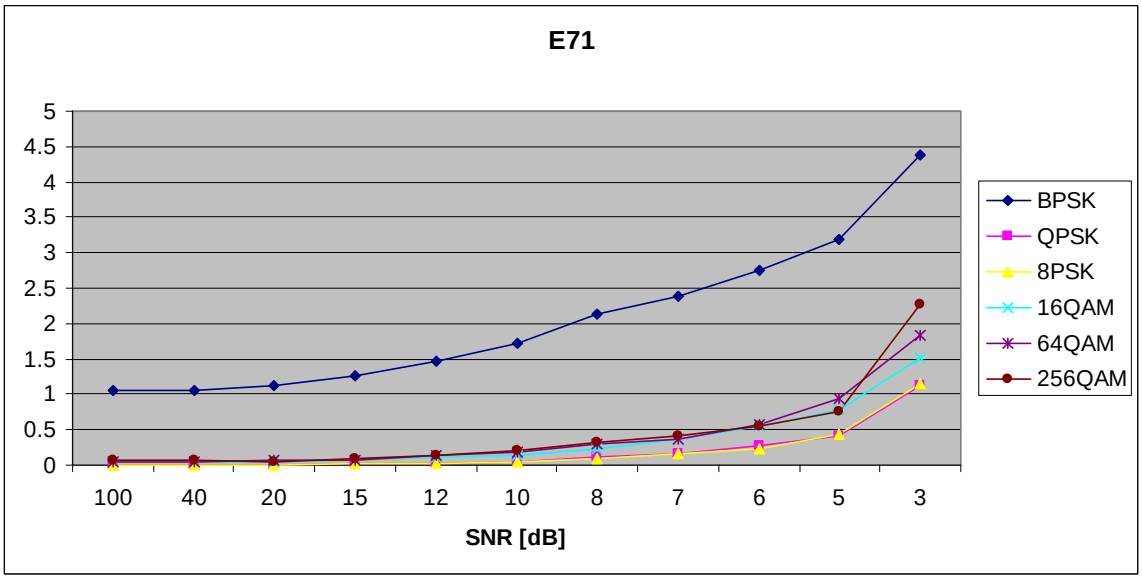

Figure 79. $E_{x,7,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

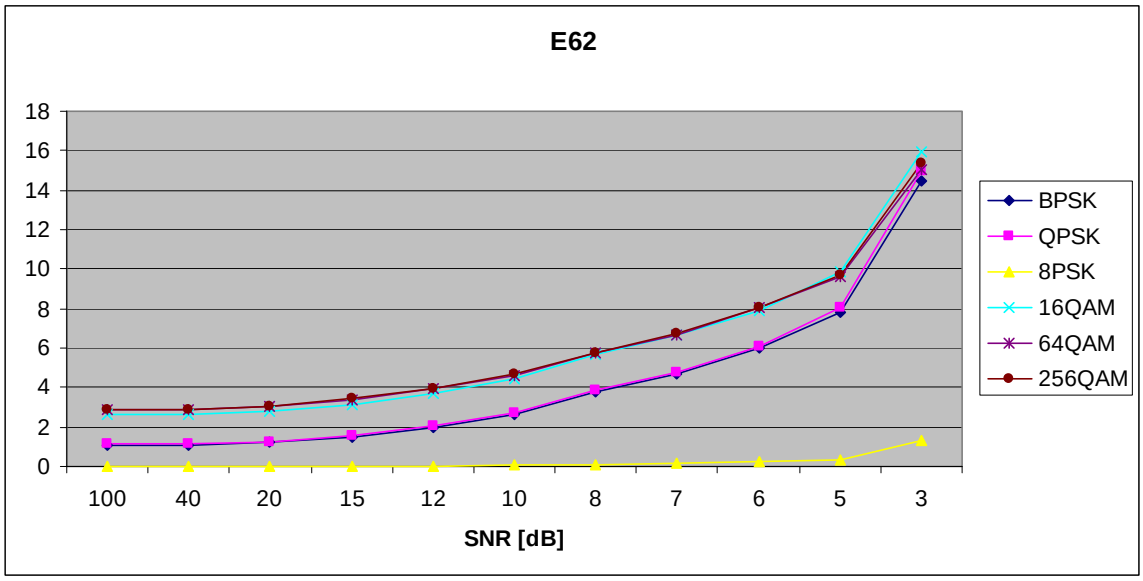

Figure 80. $E_{x,6,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

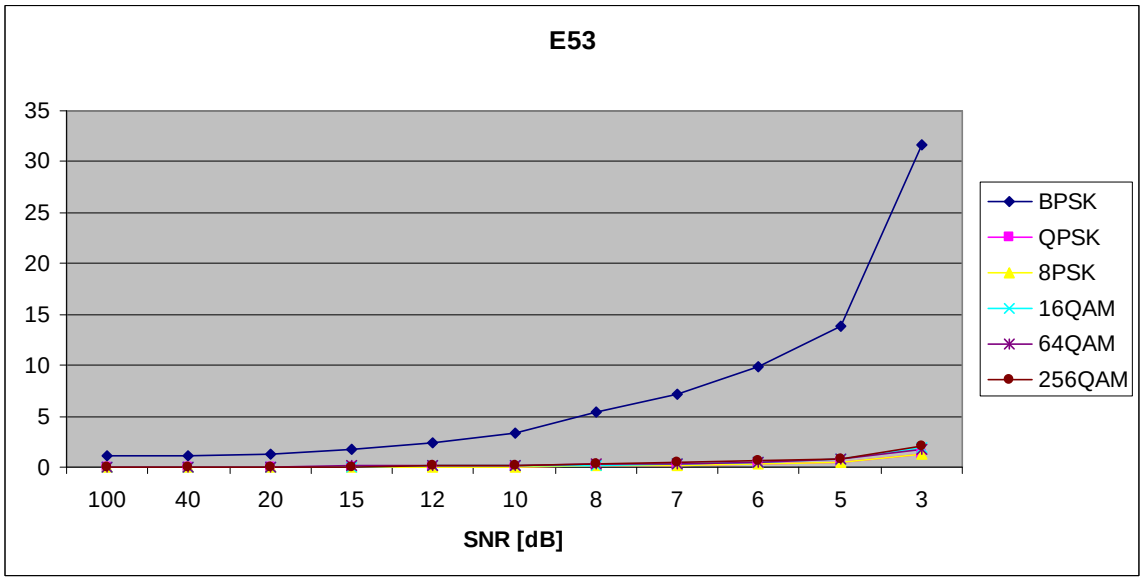

Figure 81. $E_{x,5,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

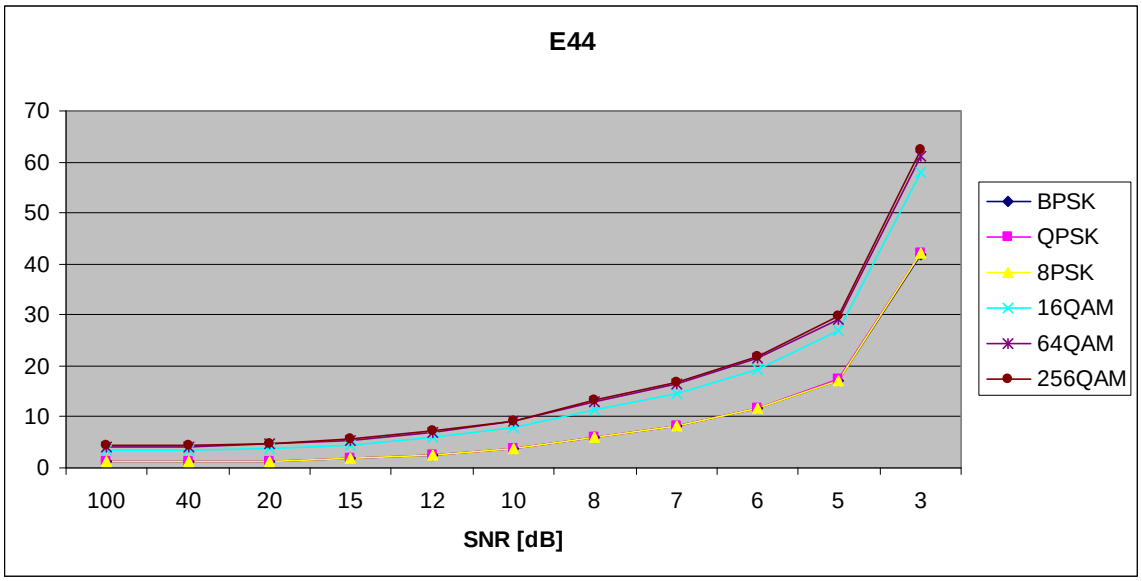

Figure 82. $E_{x,4,4}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

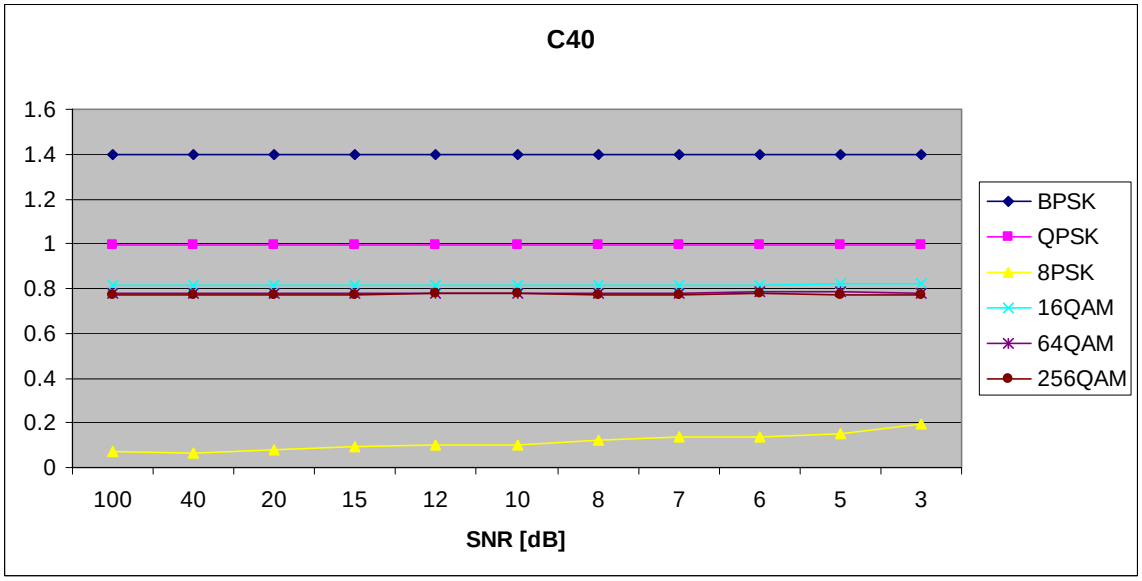

Figure 83. $C_{x,4,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

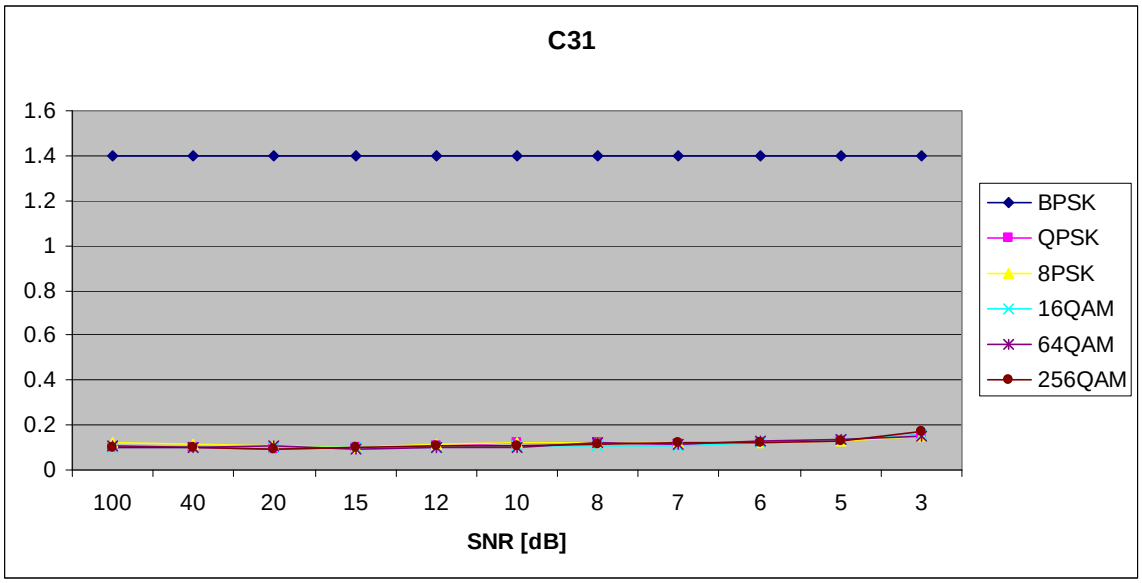

Figure 84. $C_{x,3,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

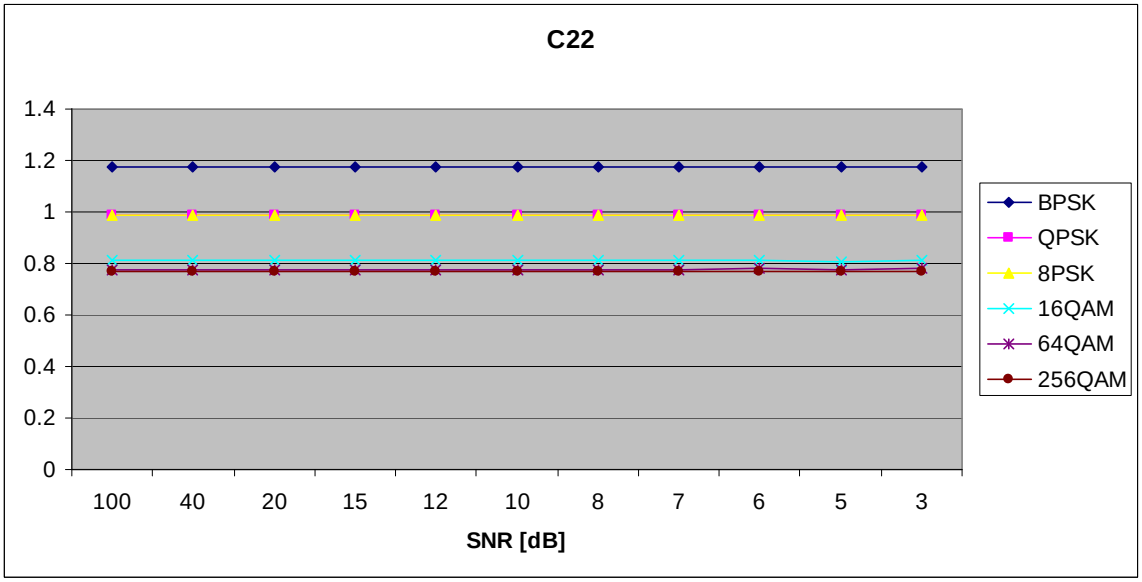

Figure 85. $C_{x,2,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

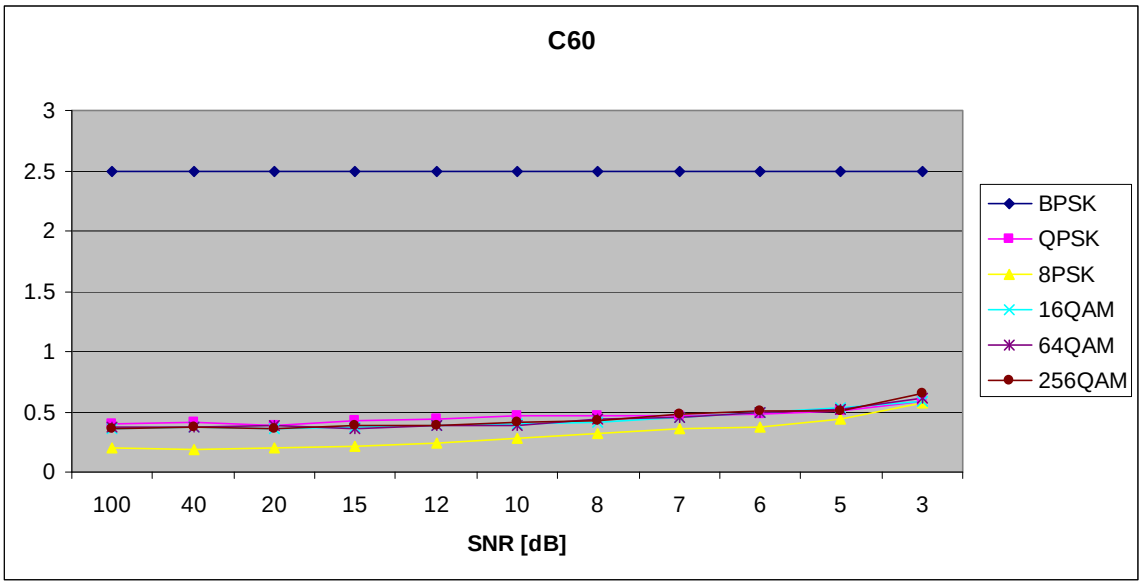

Figure 86. $C_{x,6,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

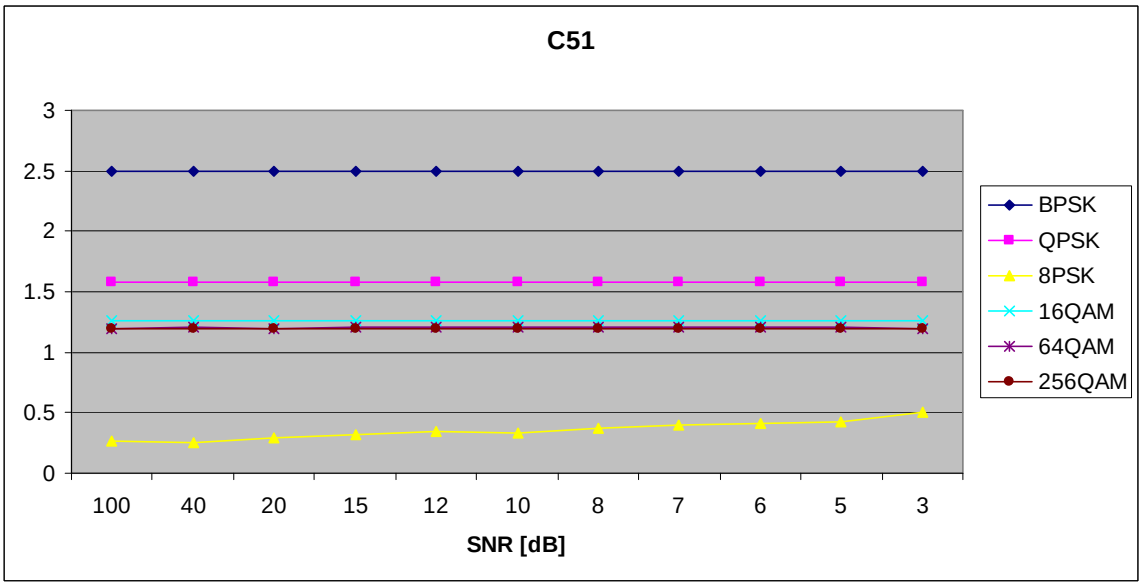

Figure 87. $C_{x,5,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

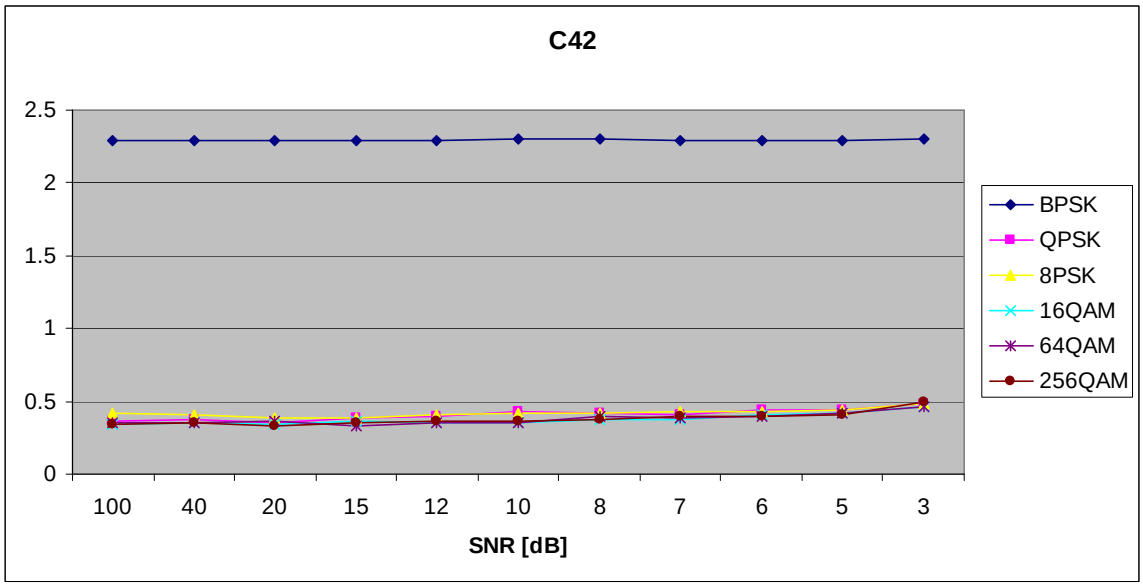

Figure 88. $C_{x,4,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

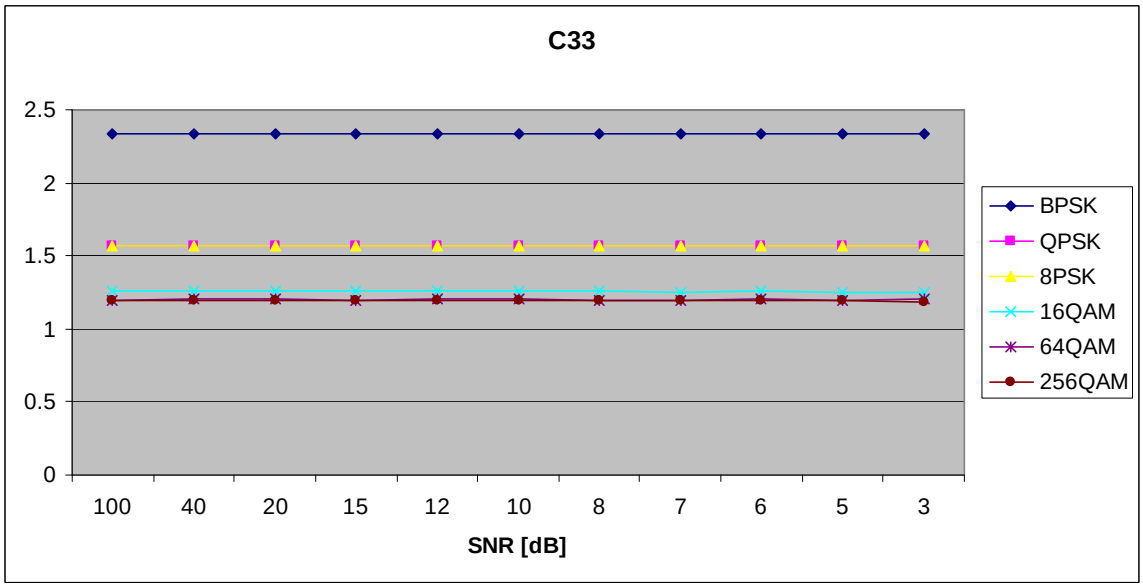

Figure 89. $C_{x,3,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

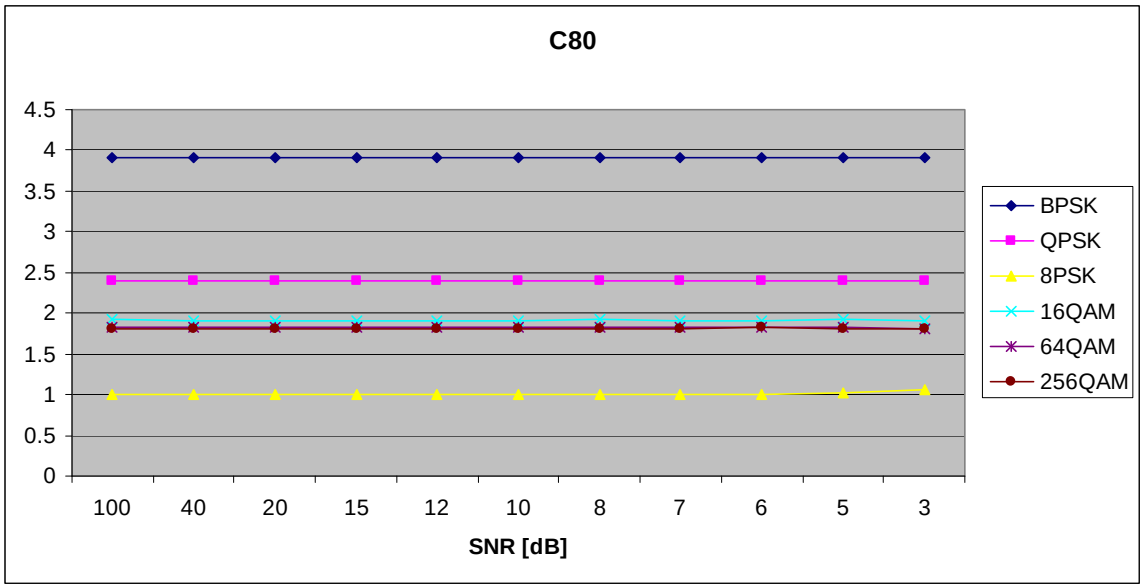

Figure 90. $C_{x,8,0}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

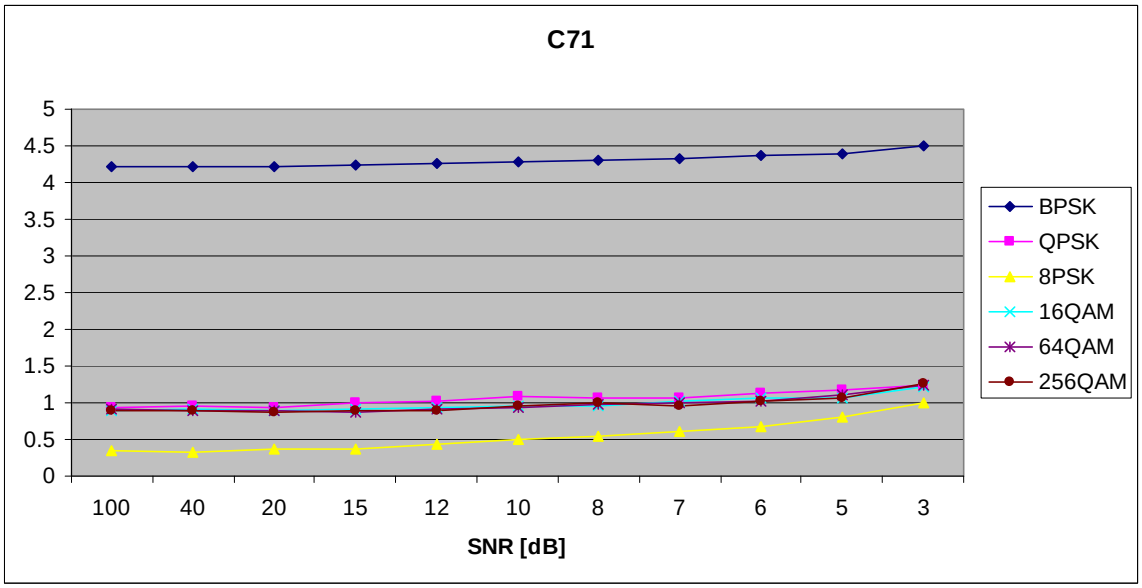

Figure 91. $C_{x,7,1}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

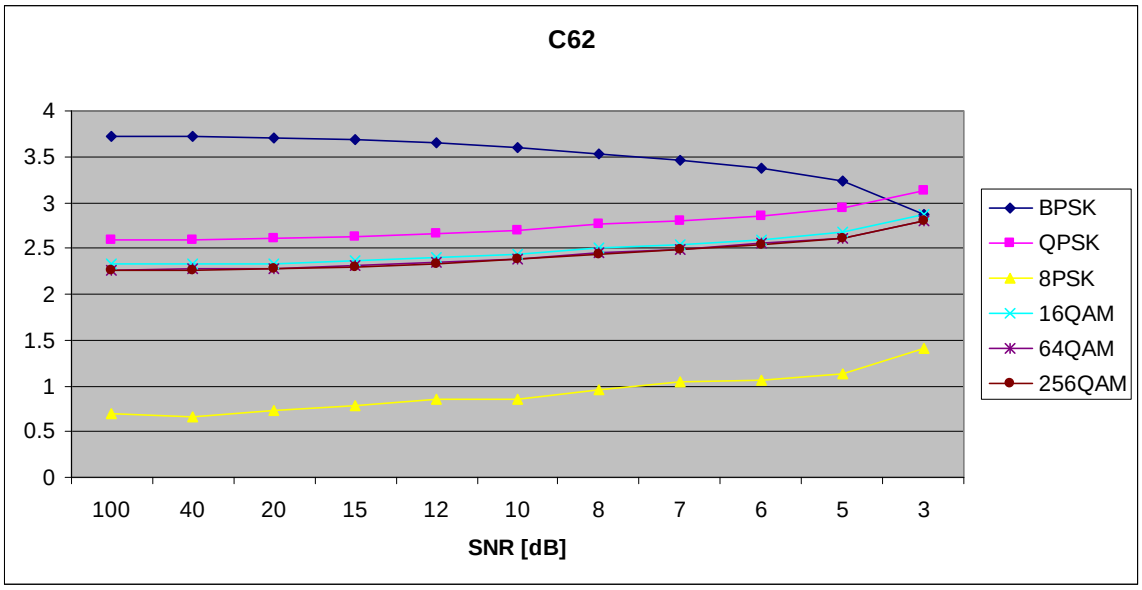

Figure 92. $C_{x,6,2}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

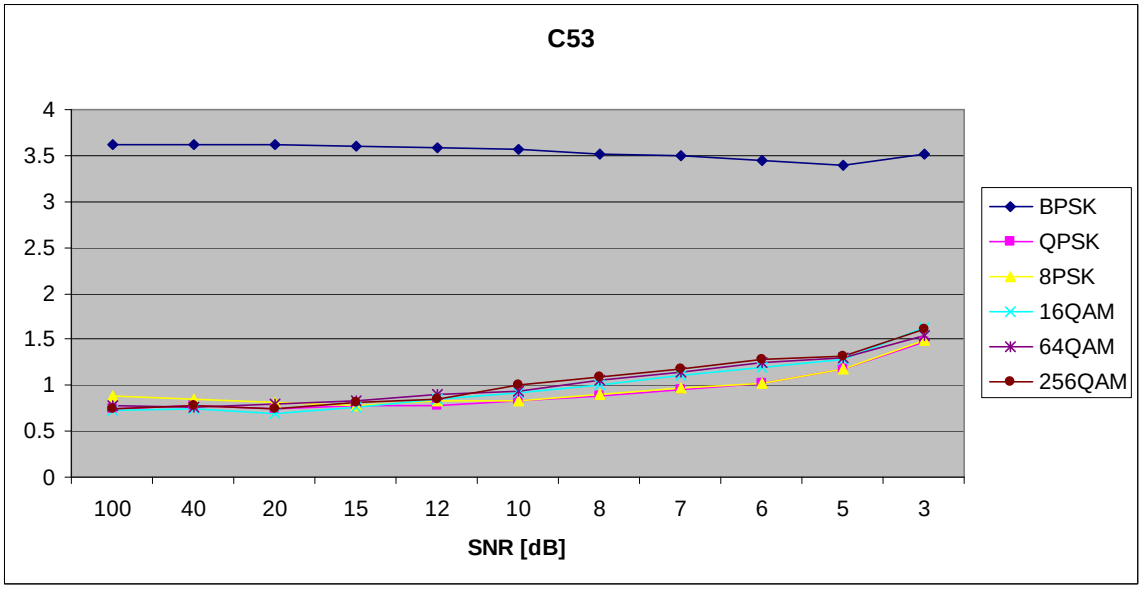

Figure 93. $C_{x,5,3}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

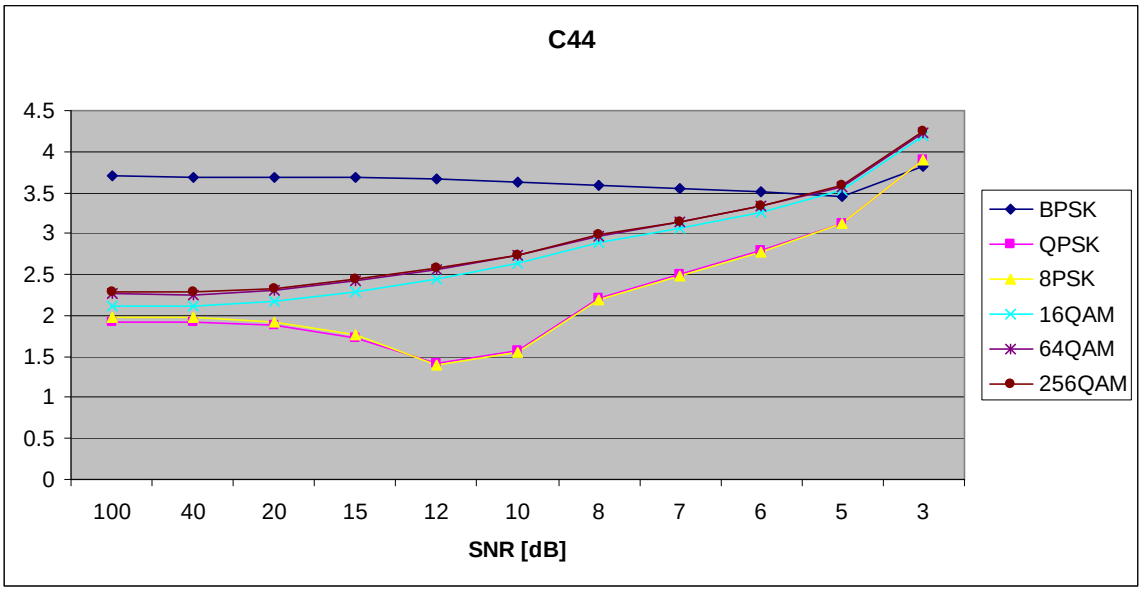

Figure 94. $C_{x,4,4}$ in AWGN and Slow, Frequency-Flat Ricean Fading.

D. AWGN PLUS SLOW, FREQUENCY-SELECTIVE RAYLEIGH FADING

Parameters for the rayleighchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 3.5 Hz
- Path Delays: [0, 2×10^{-6}]
- Average Path Gains: [0, -10]

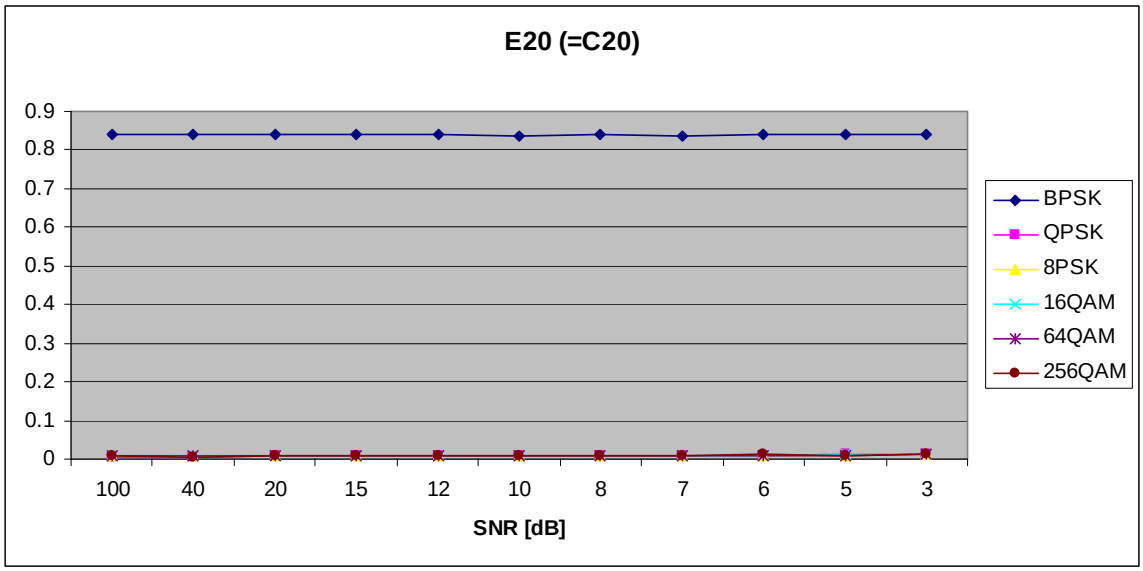

Figure 95. $E_{x,2,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

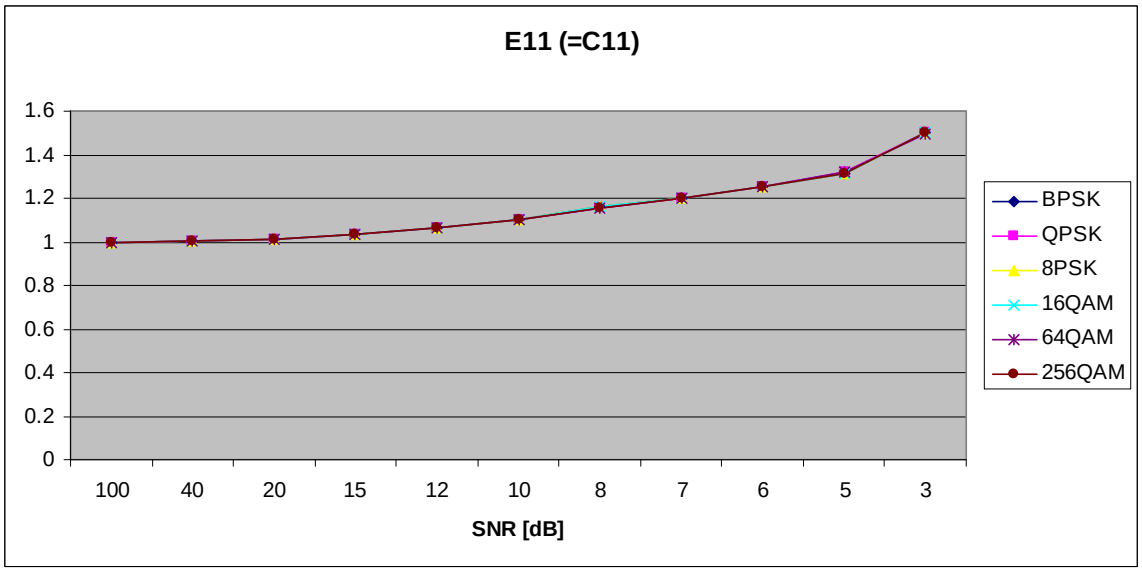

Figure 96. $E_{x,1,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

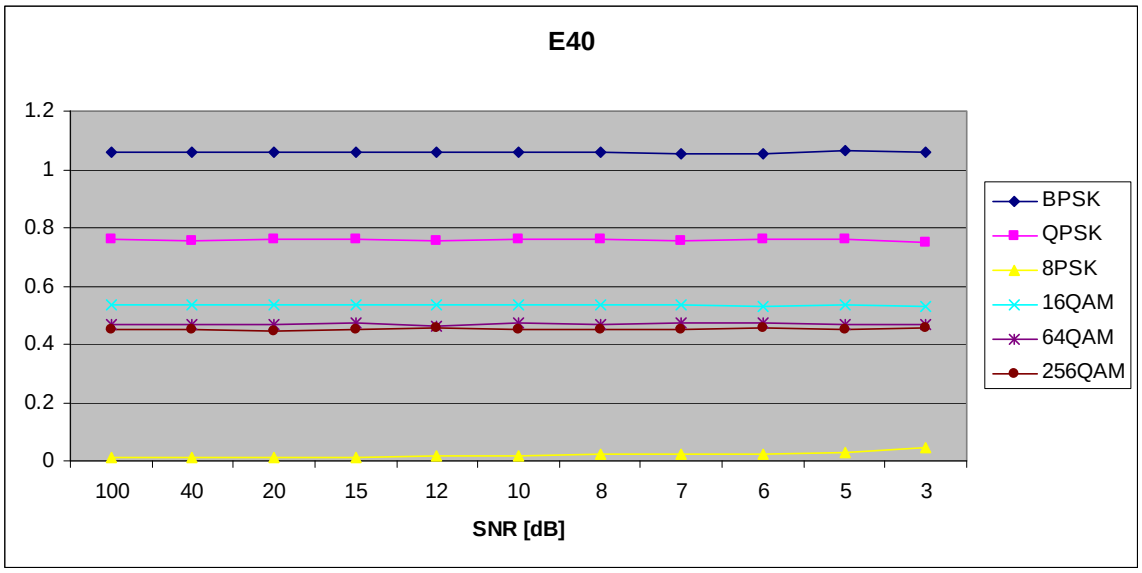

Figure 97. $E_{x,4,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

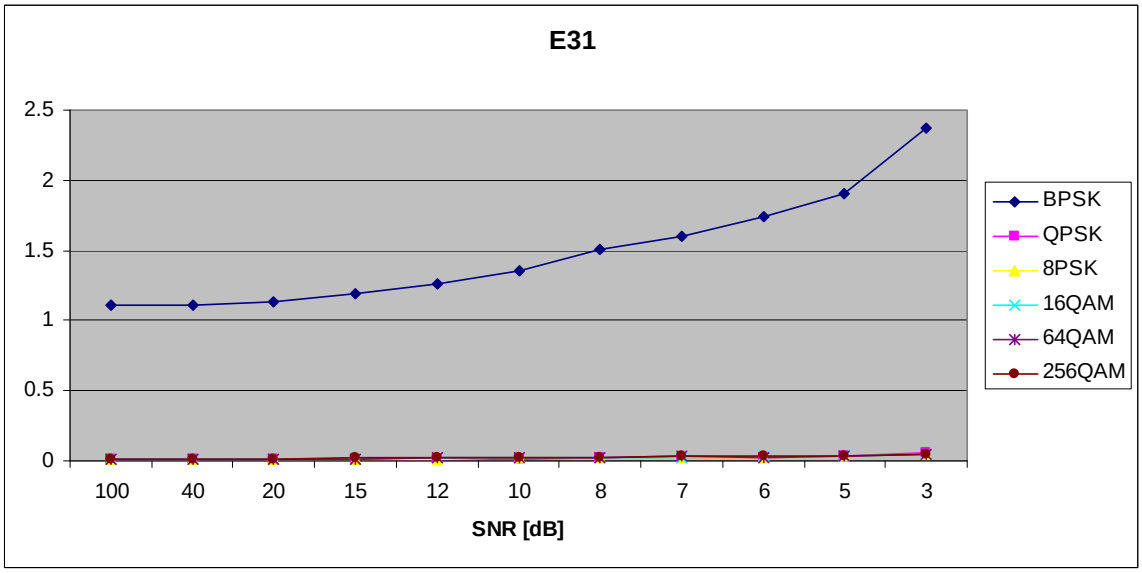

Figure 98. $E_{x,3,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

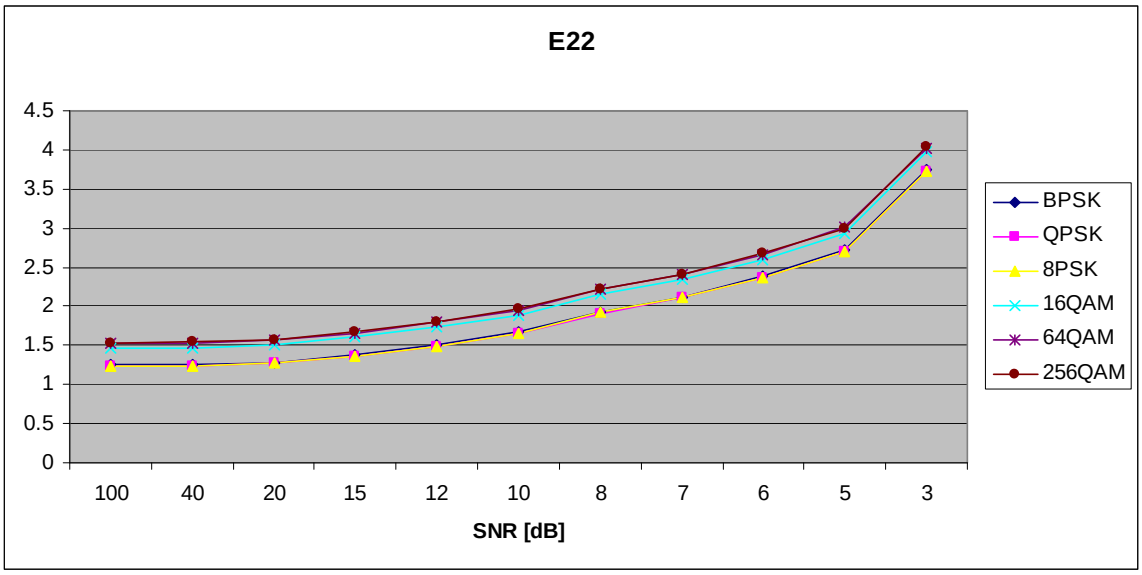

Figure 99. $E_{x,2,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

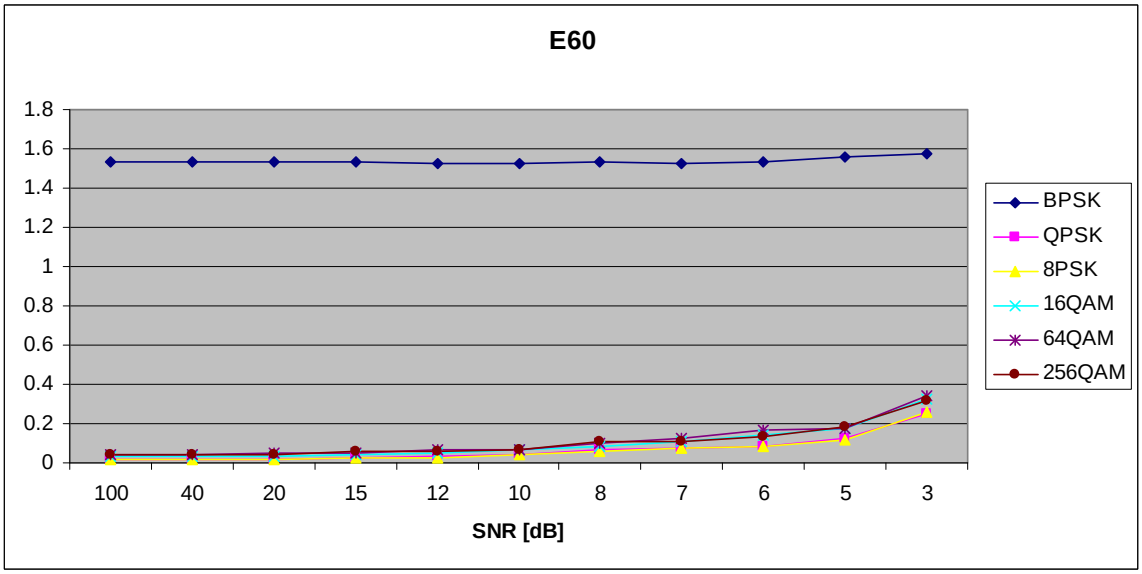

Figure 100. $E_{x,6,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

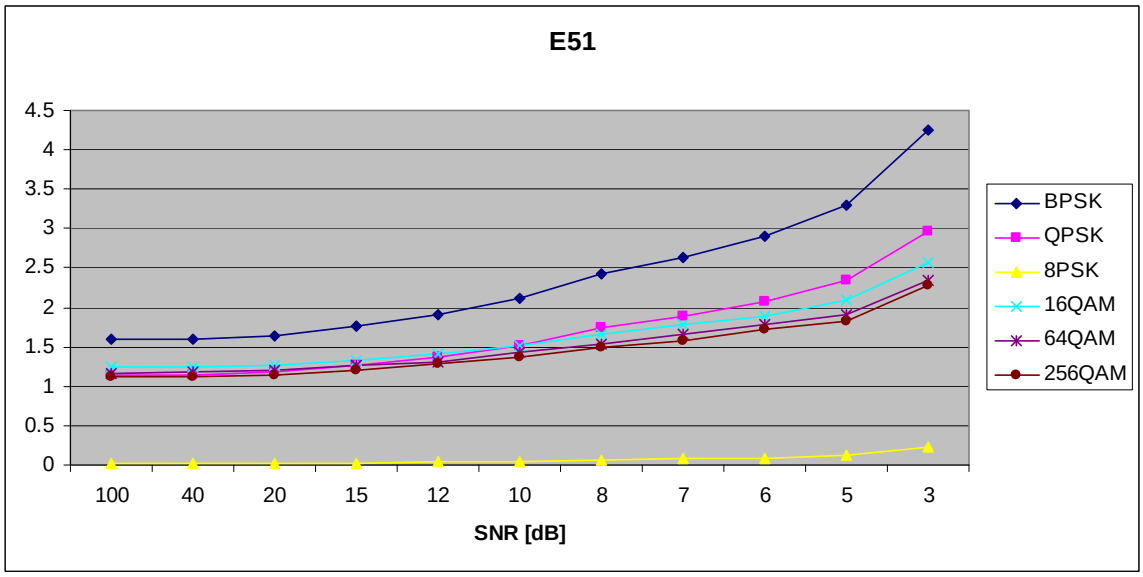

Figure 101. $E_{x,5,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

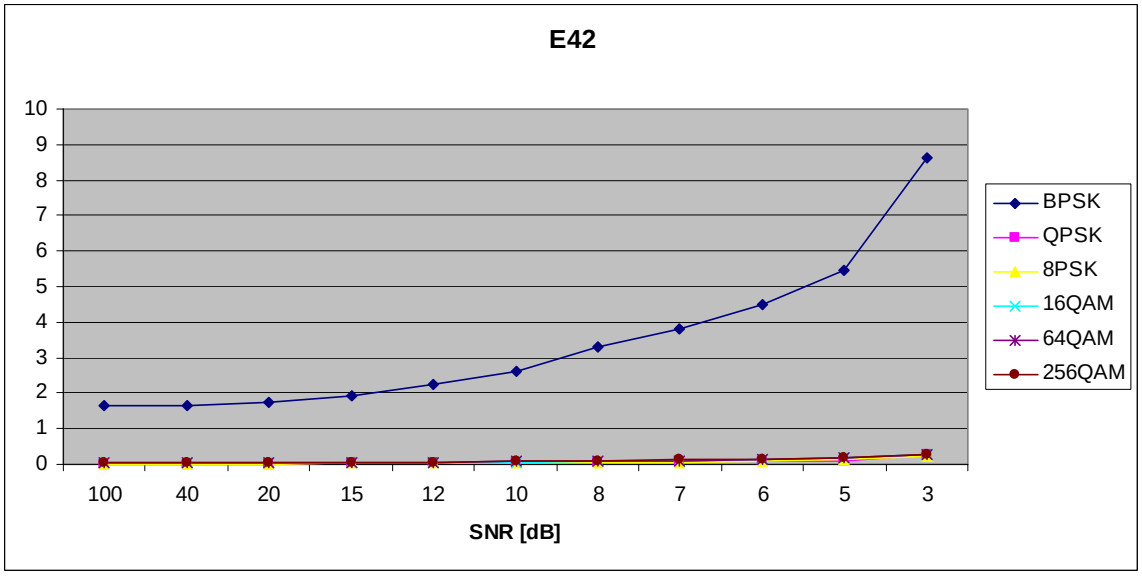

Figure 102. $E_{x,4,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

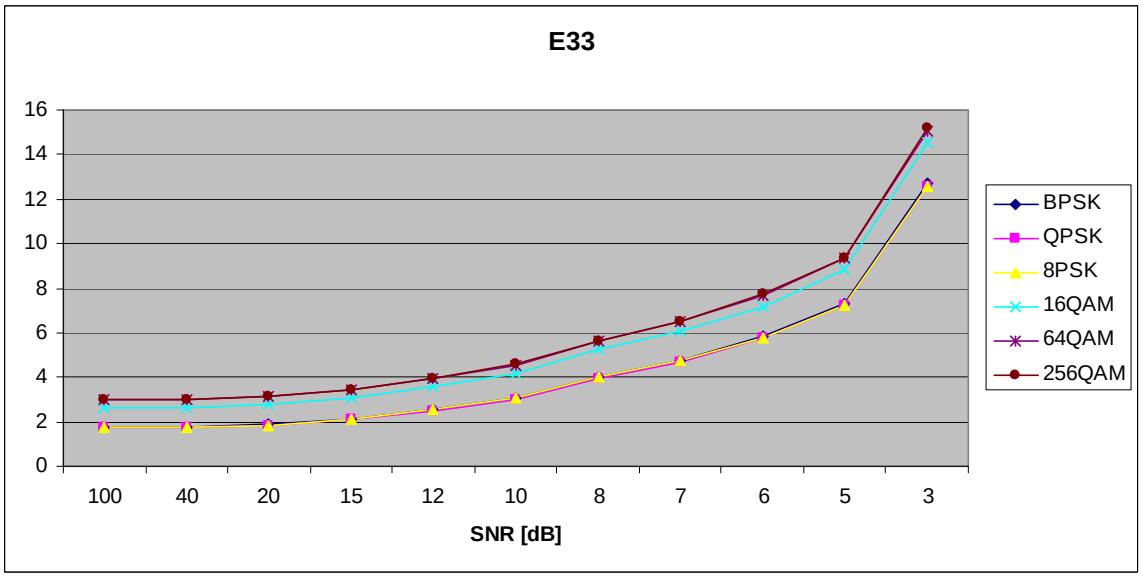

Figure 103. $E_{x,3,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

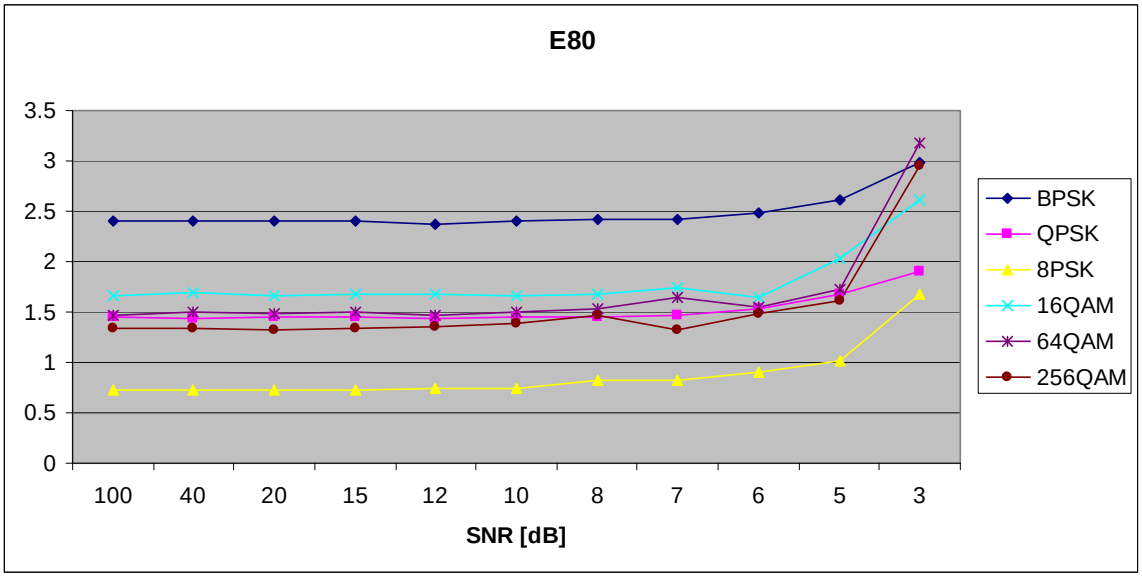

Figure 104. $E_{x,8,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

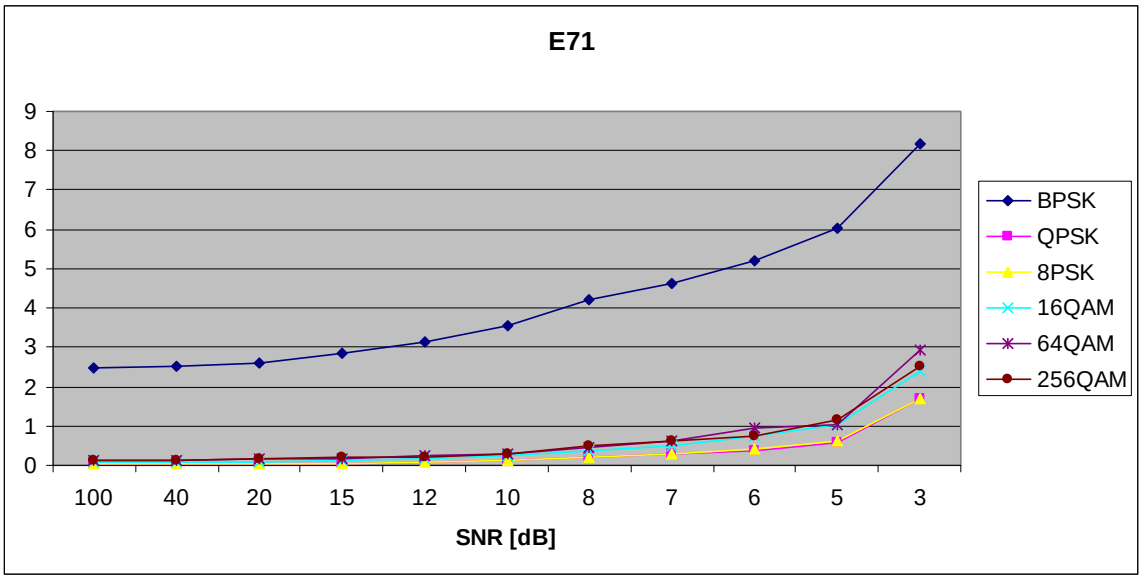

Figure 105. $E_{x,7,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

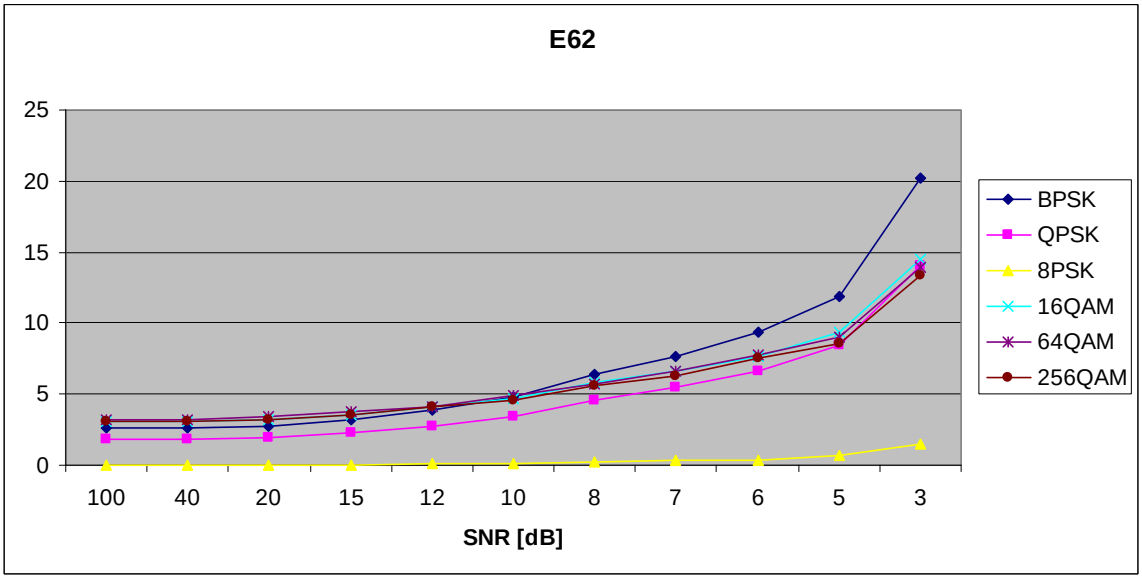

Figure 106. $E_{x,6,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

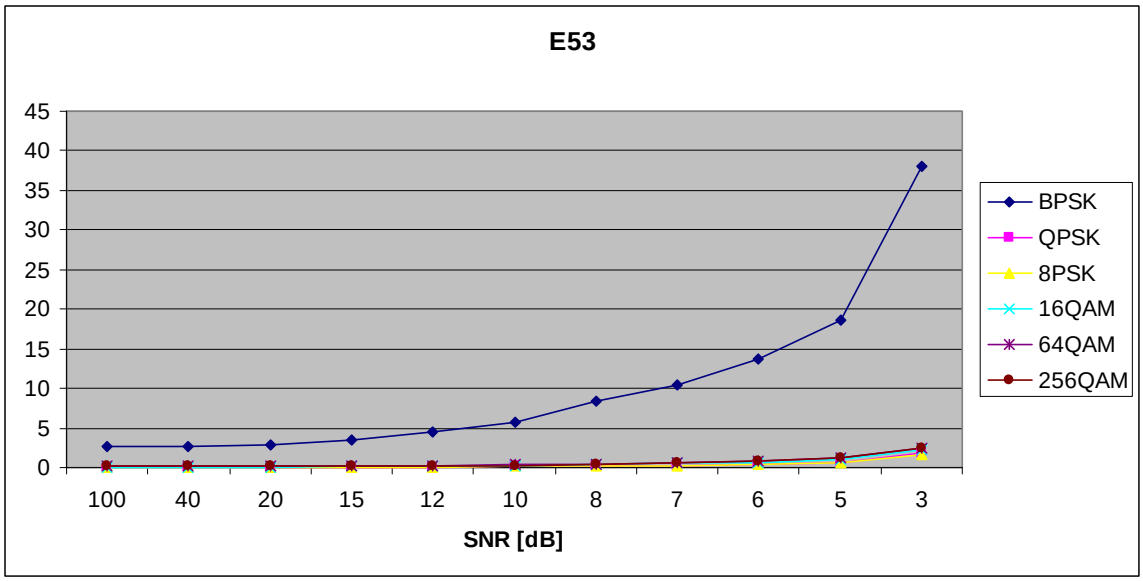

Figure 107. $E_{x,5,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

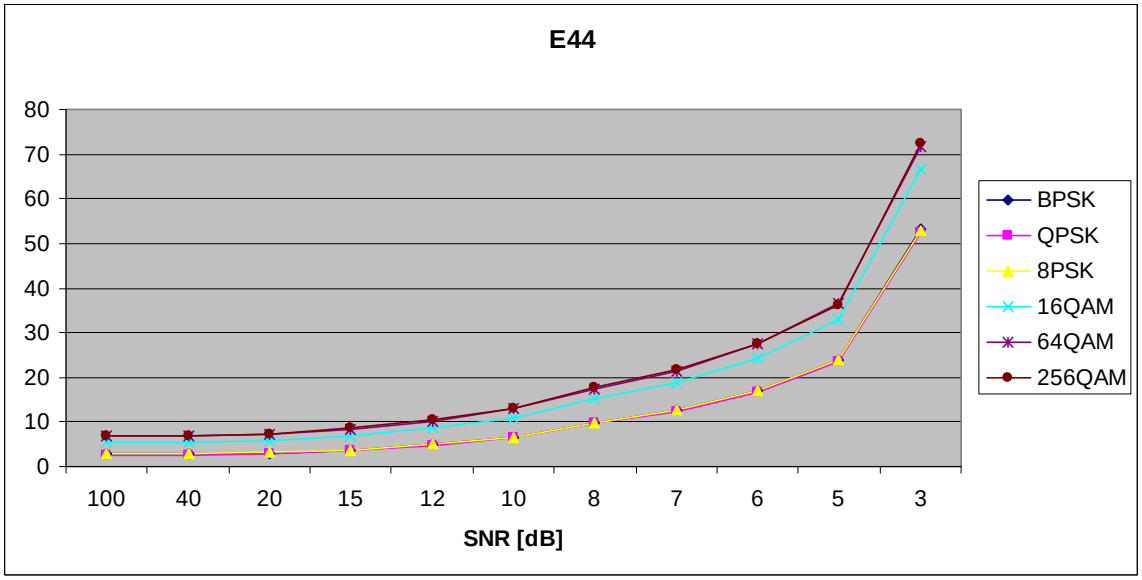

Figure 108. $E_{x,4,4}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

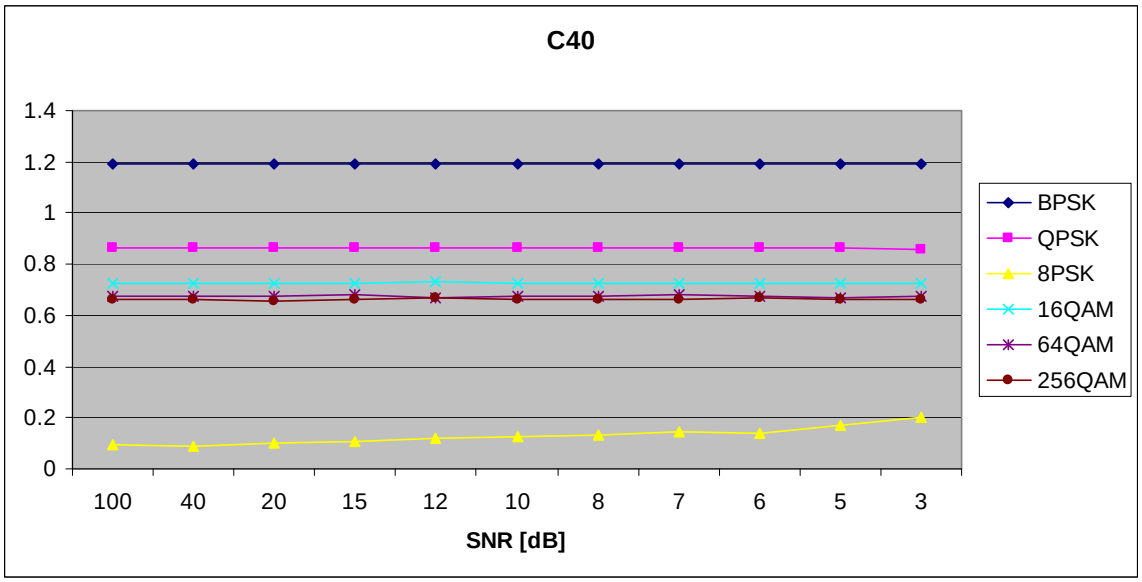

Figure 109. $C_{x,4,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

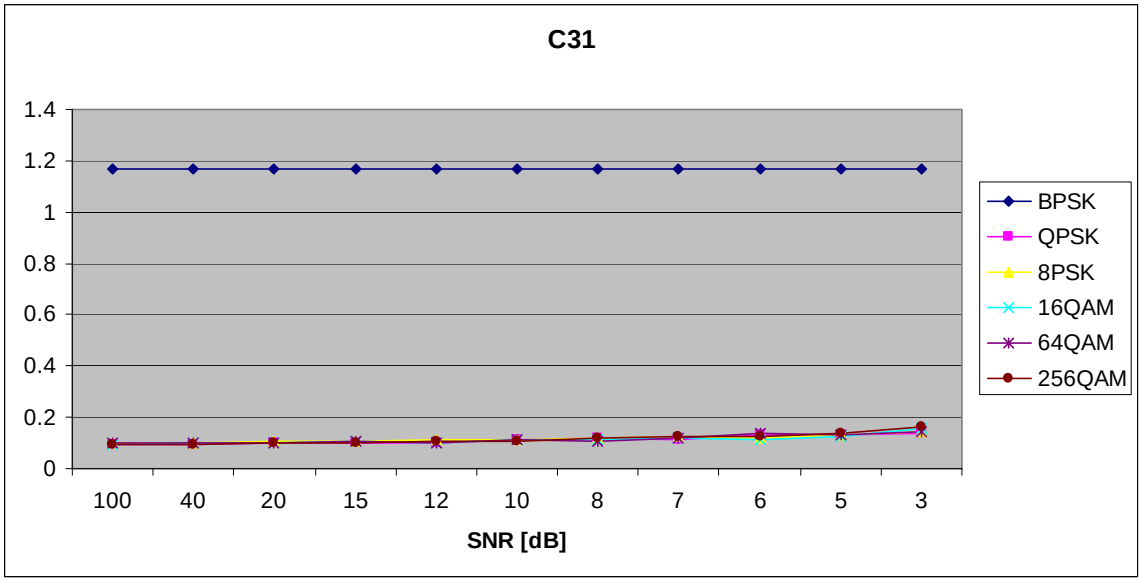

Figure 110. $C_{x,3,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

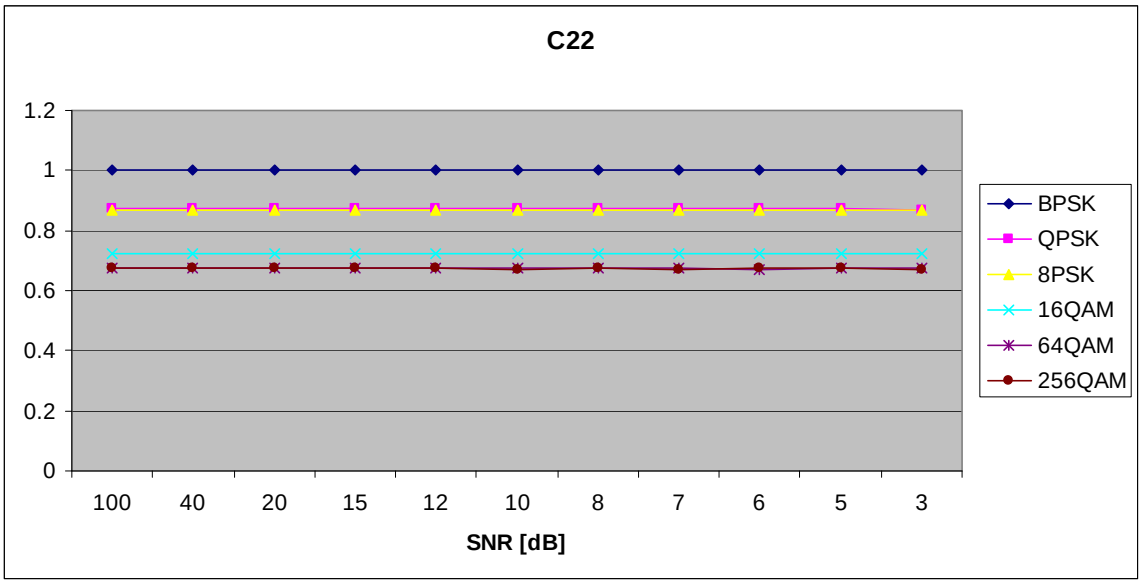

Figure 111. $C_{x,2,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

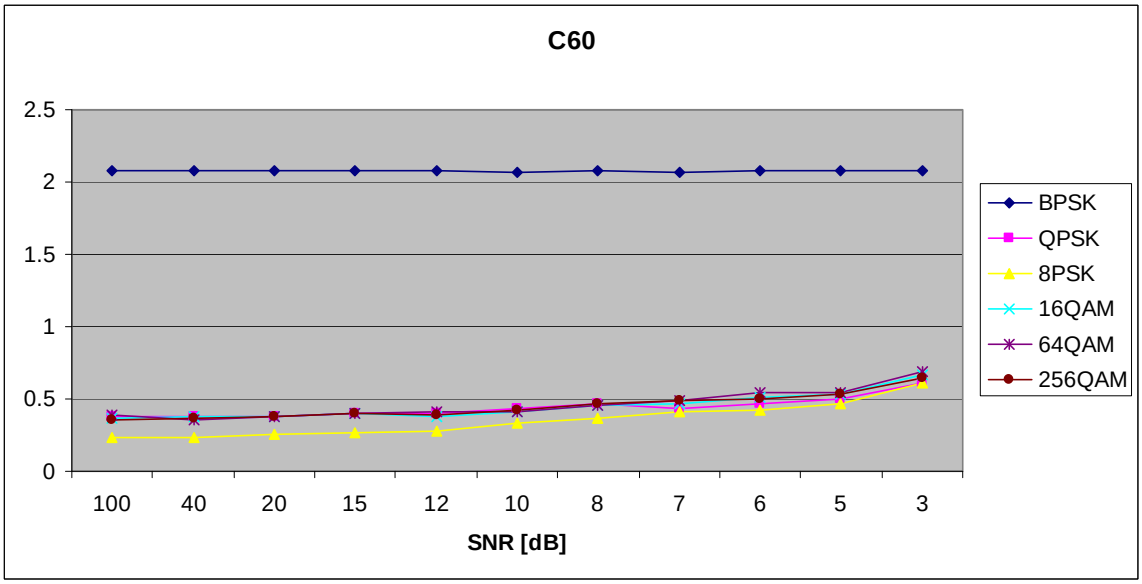

Figure 112. $C_{x,6,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

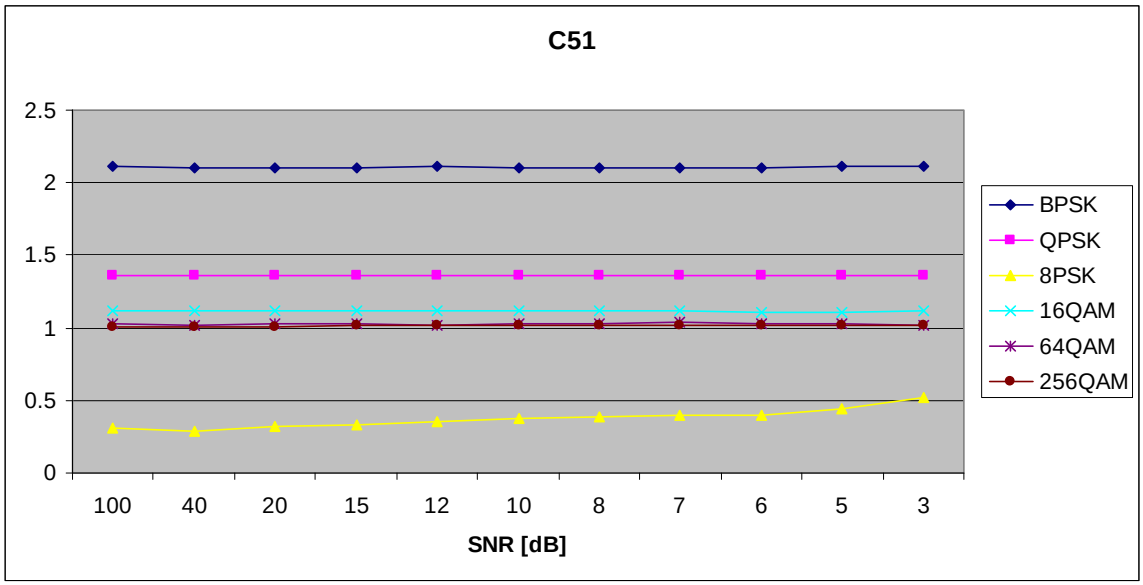

Figure 113. $C_{x,5,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

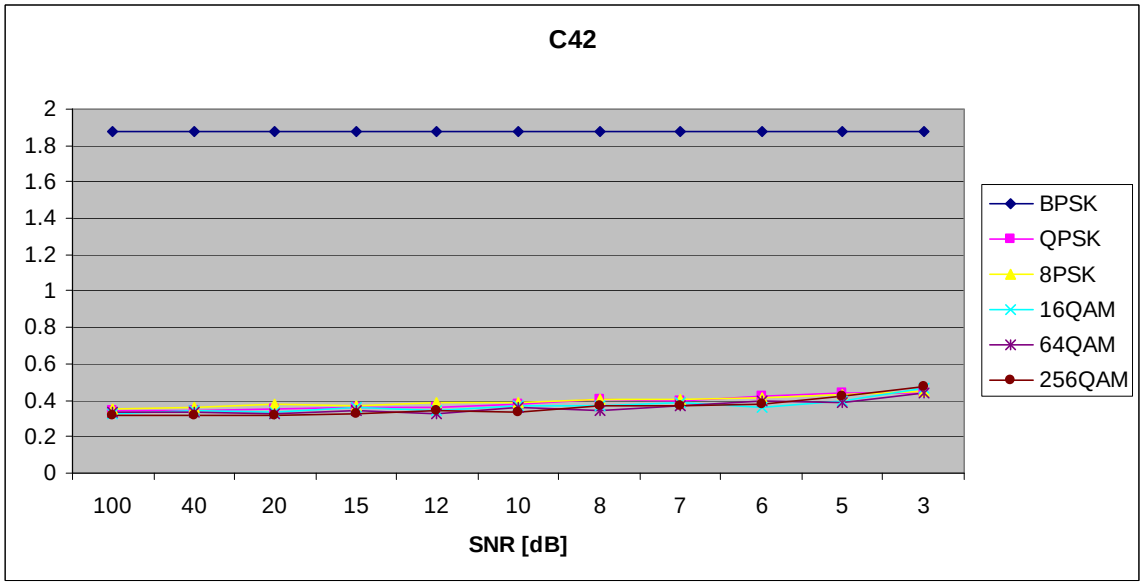

Figure 114. $C_{x,4,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

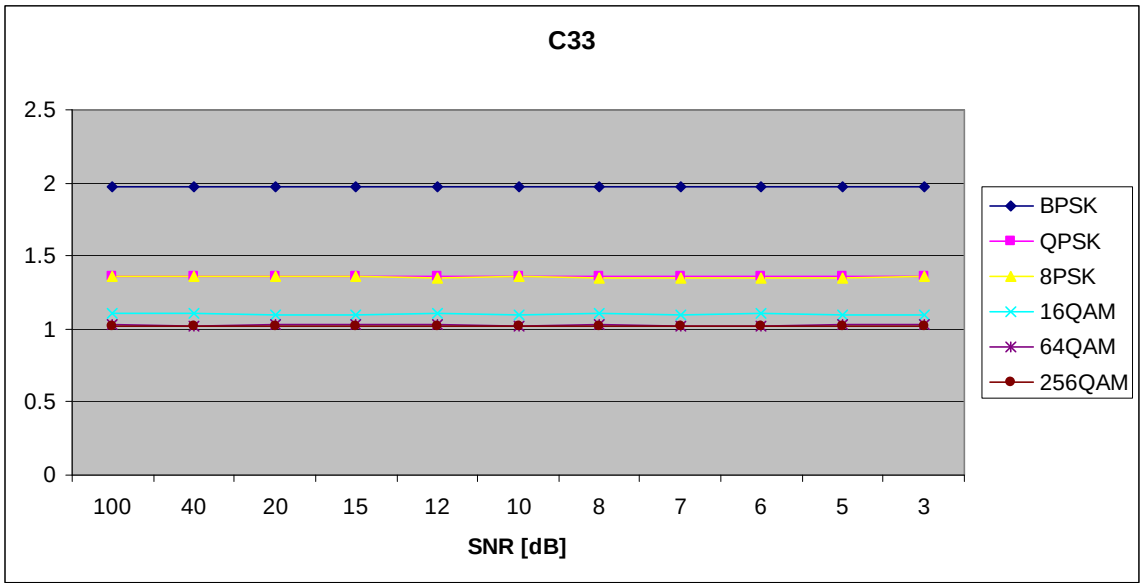

Figure 115. $C_{x,3,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

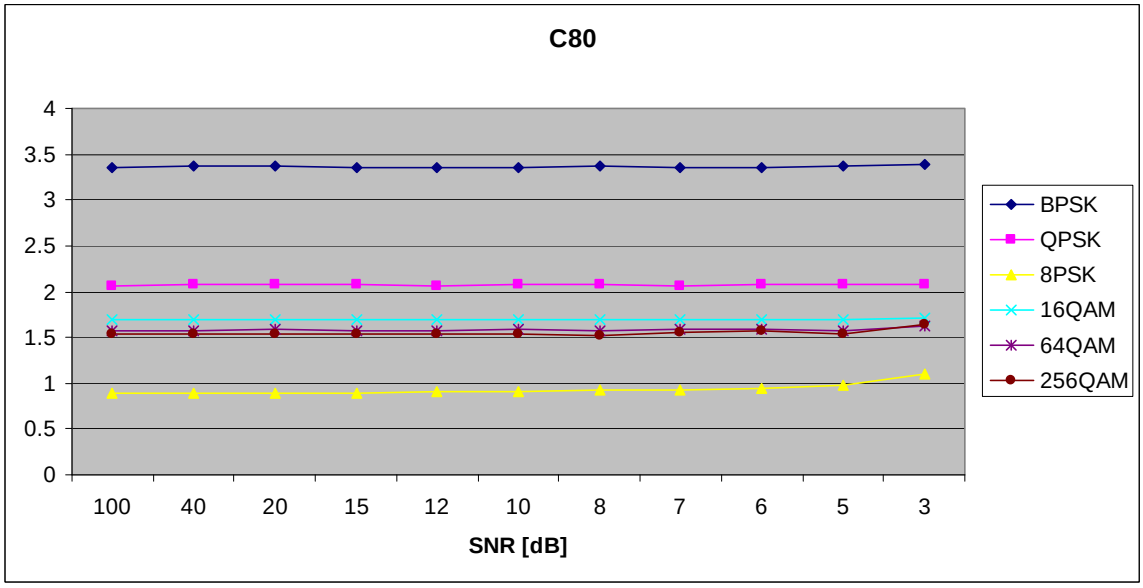

Figure 116. $C_{x,8,0}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

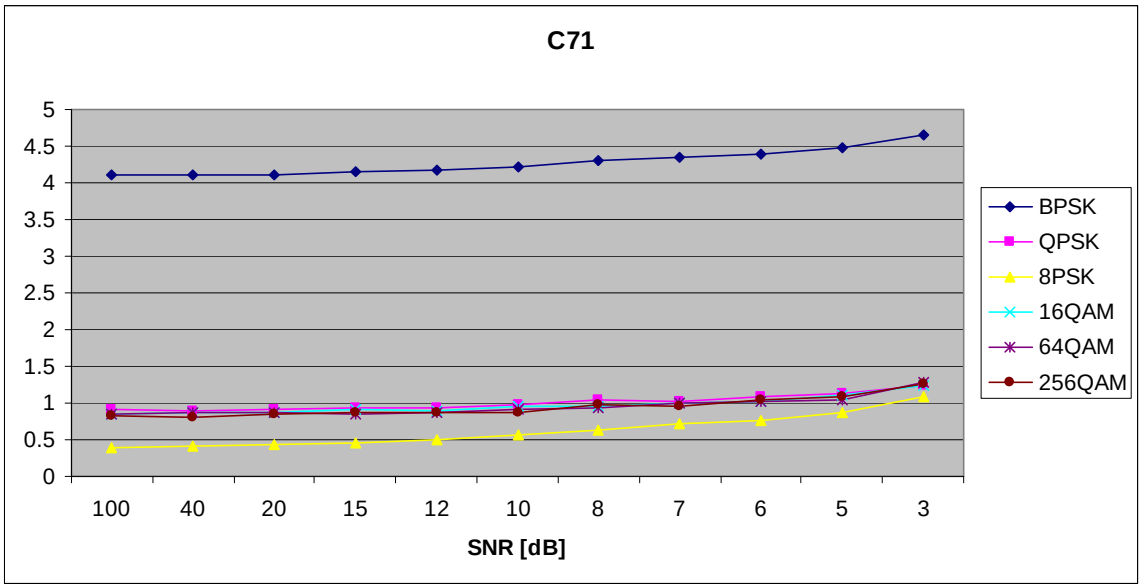

Figure 117. $C_{x,7,1}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

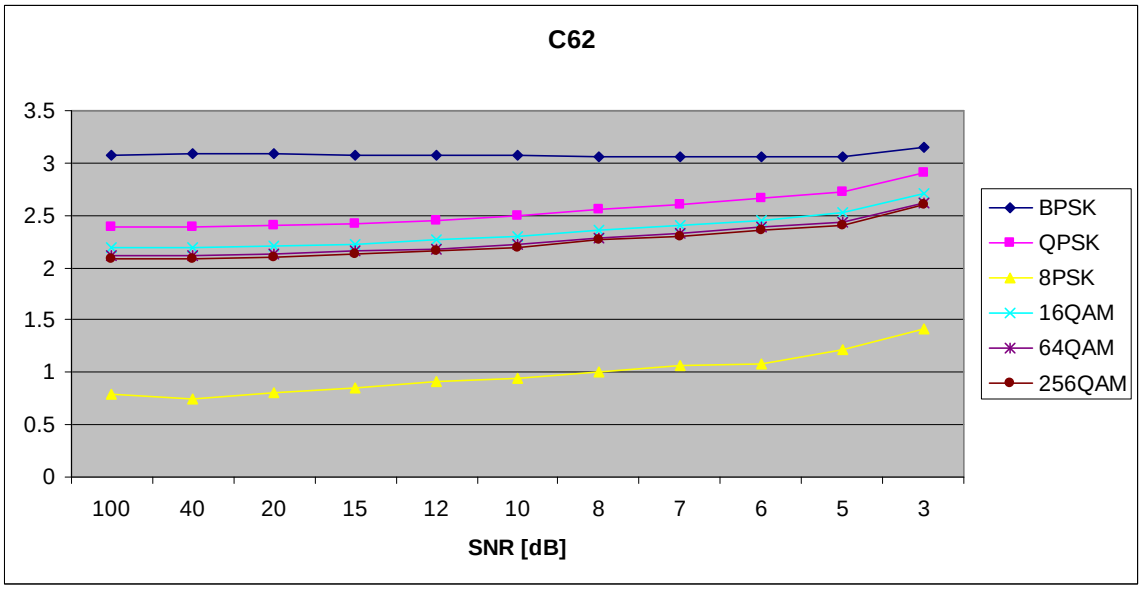

Figure 118. $C_{x,6,2}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

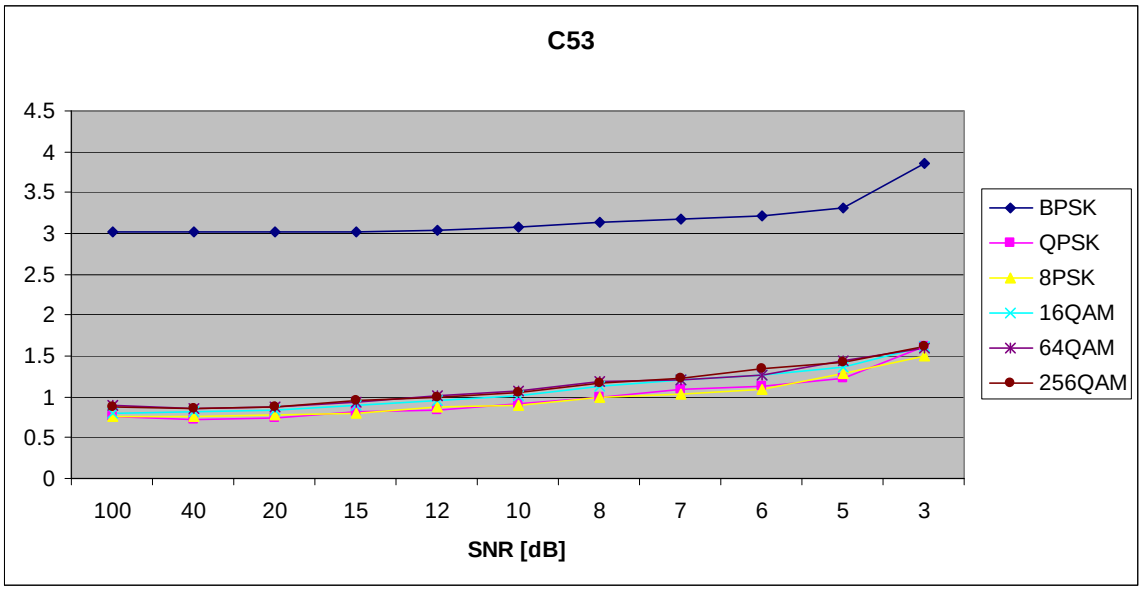

Figure 119. $C_{x,5,3}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

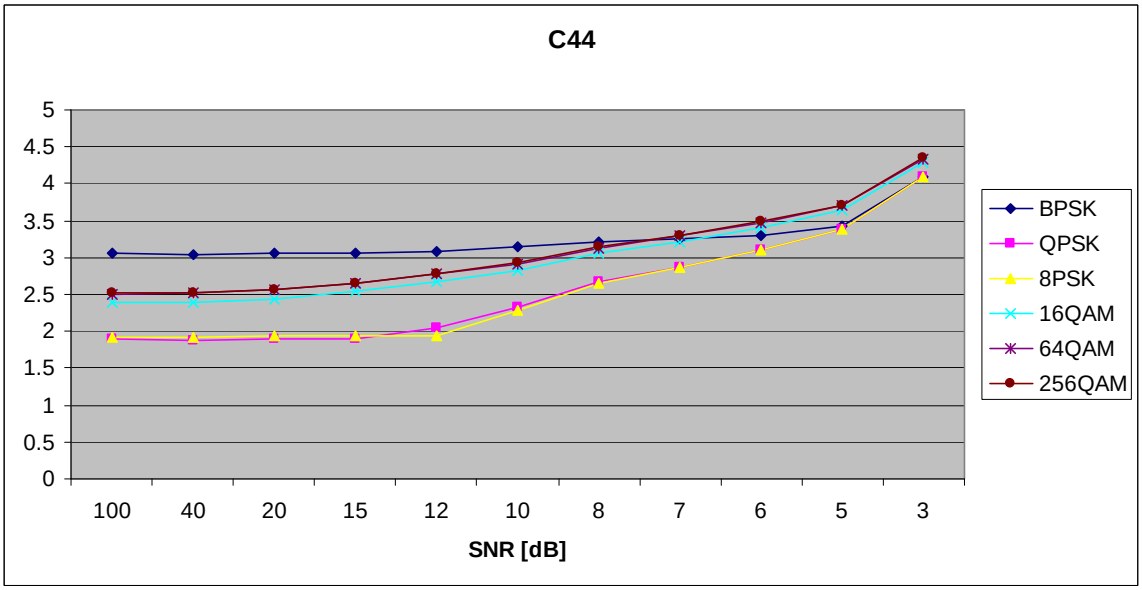

Figure 120. $C_{x,4,4}$ in AWGN and Slow, Frequency-Selective Rayleigh Fading.

E. AWGN PLUS SLOW, FREQUENCY-SELECTIVE RICEAN FADING

Parameters for the ricianchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 3.5 Hz
- K-factor: 3
- Path Delays: $[0,\ 2\times10^{-6}]$
- Average Path Gains: [0, -10]

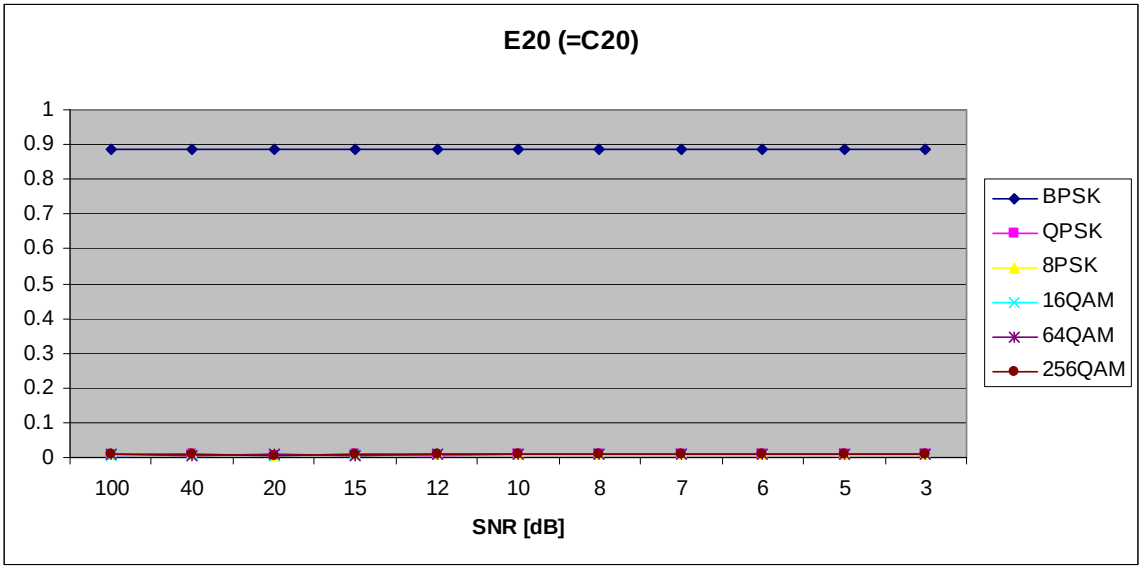

Figure 121. $E_{x,2,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

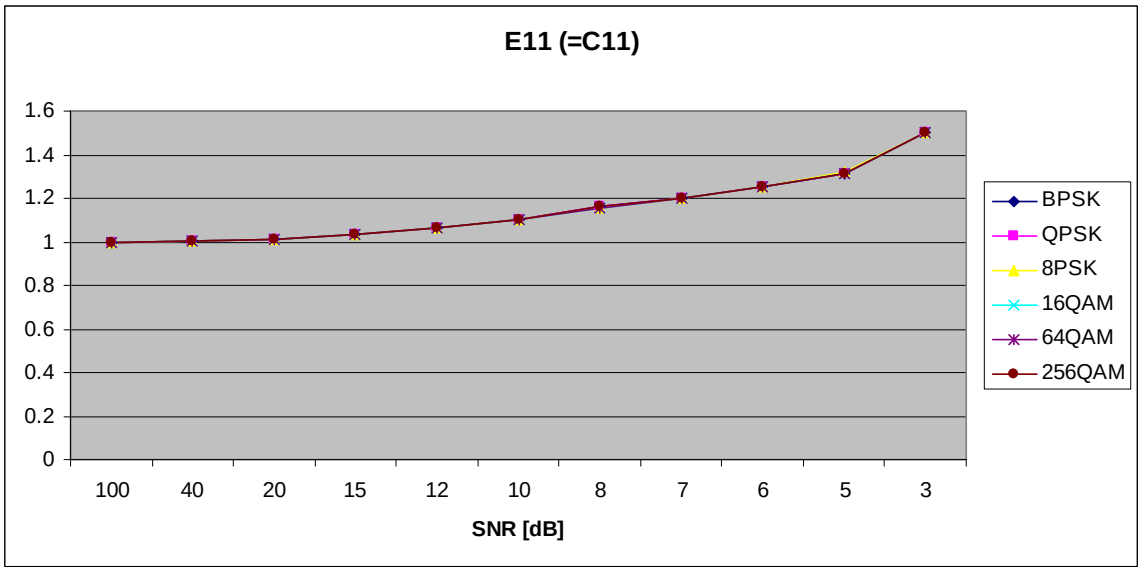

Figure 122. $E_{x,1,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

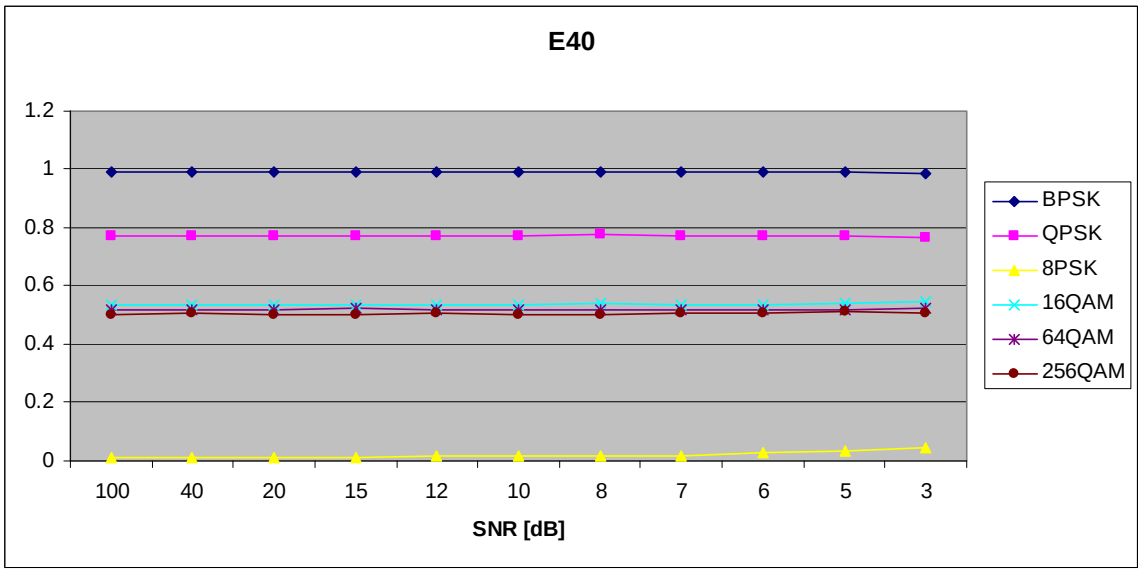

Figure 123. $E_{x,4,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

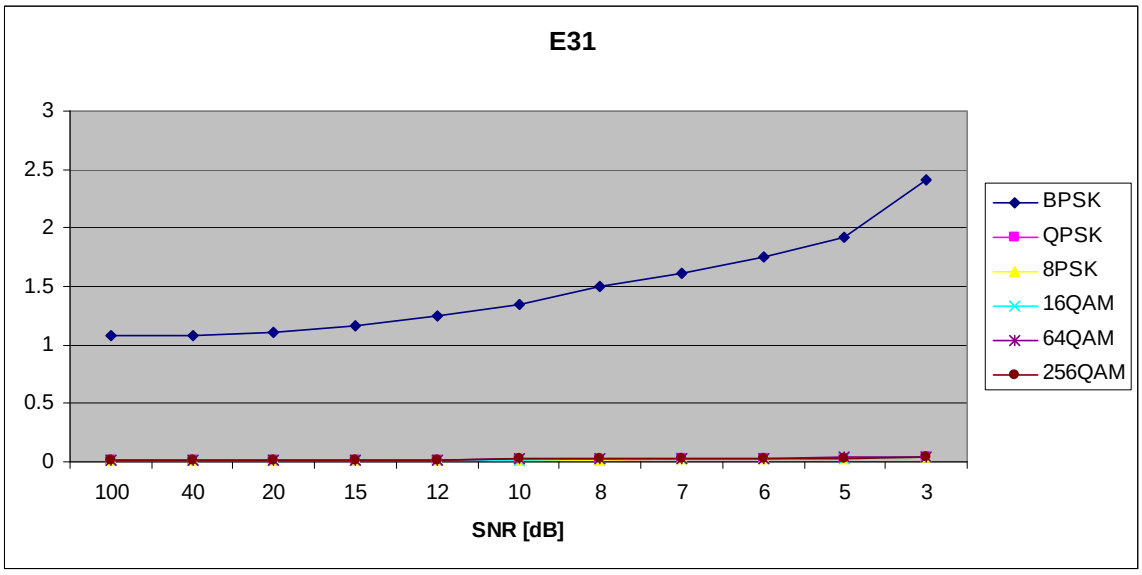

Figure 124. $E_{x,3,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

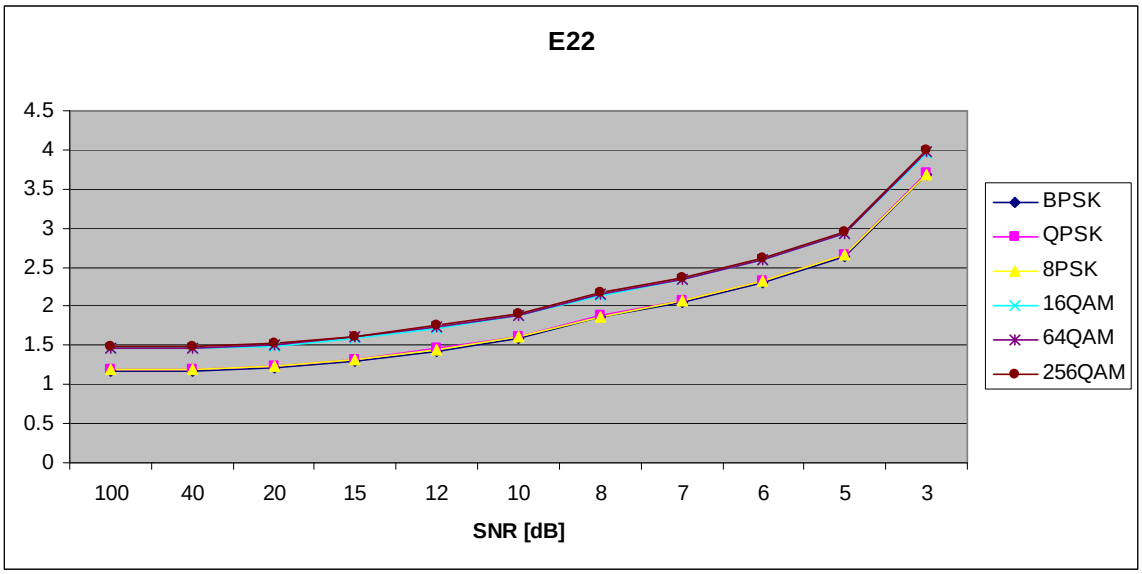

Figure 125. $E_{x,2,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

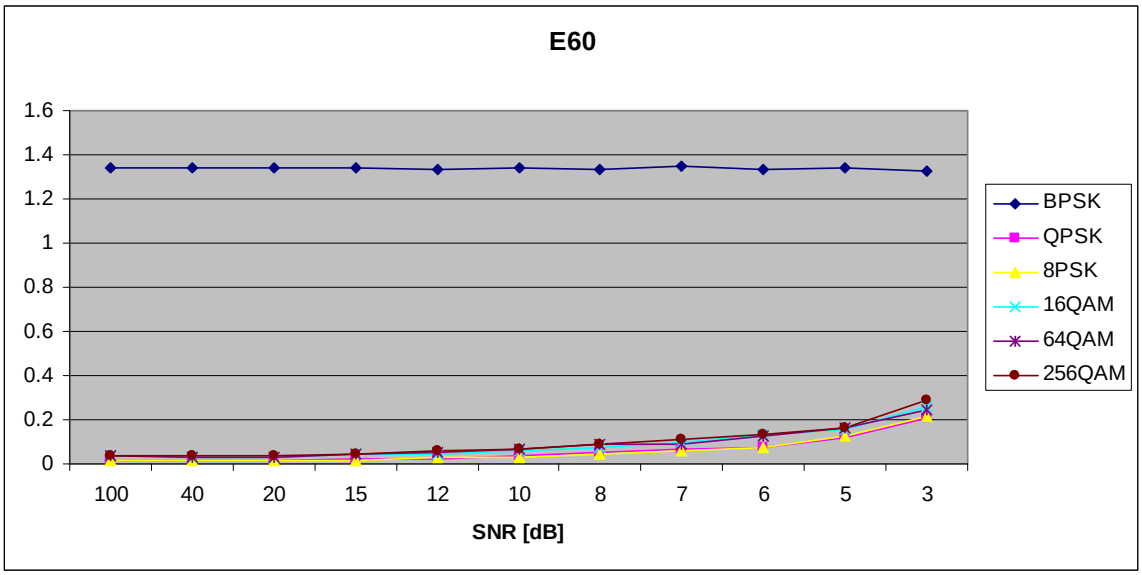

Figure 126. $E_{x,6,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

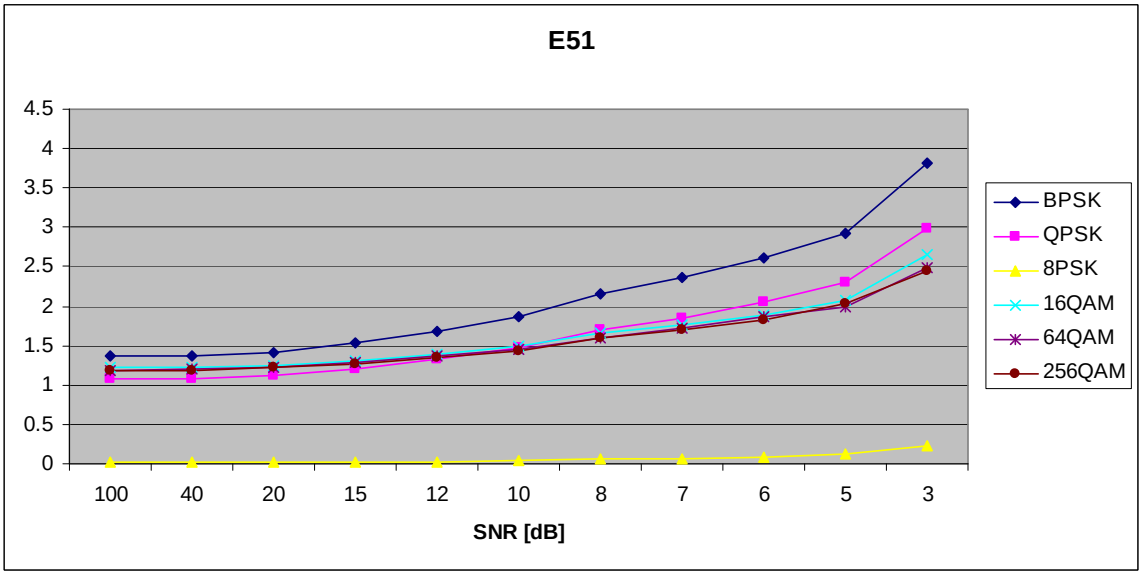

Figure 127. $E_{x,5,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

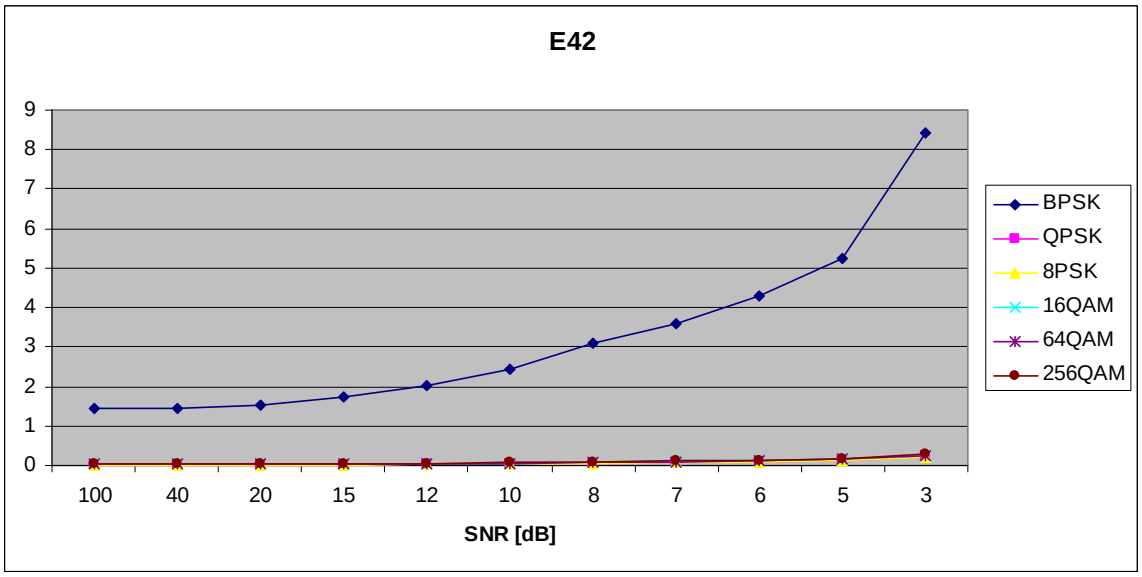

Figure 128. $E_{x,4,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

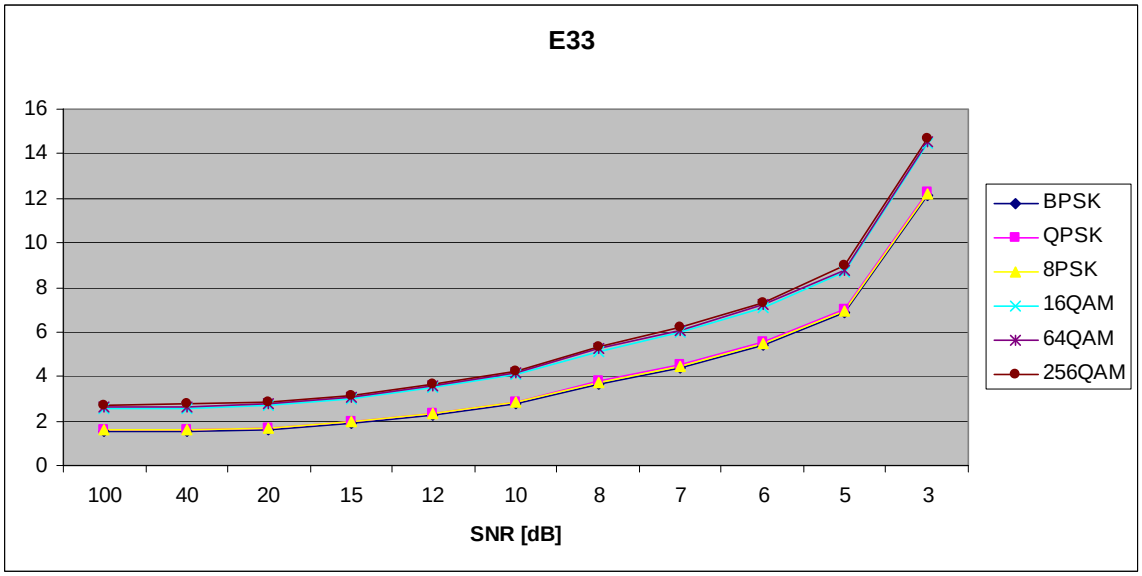

Figure 129. $E_{x,3,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

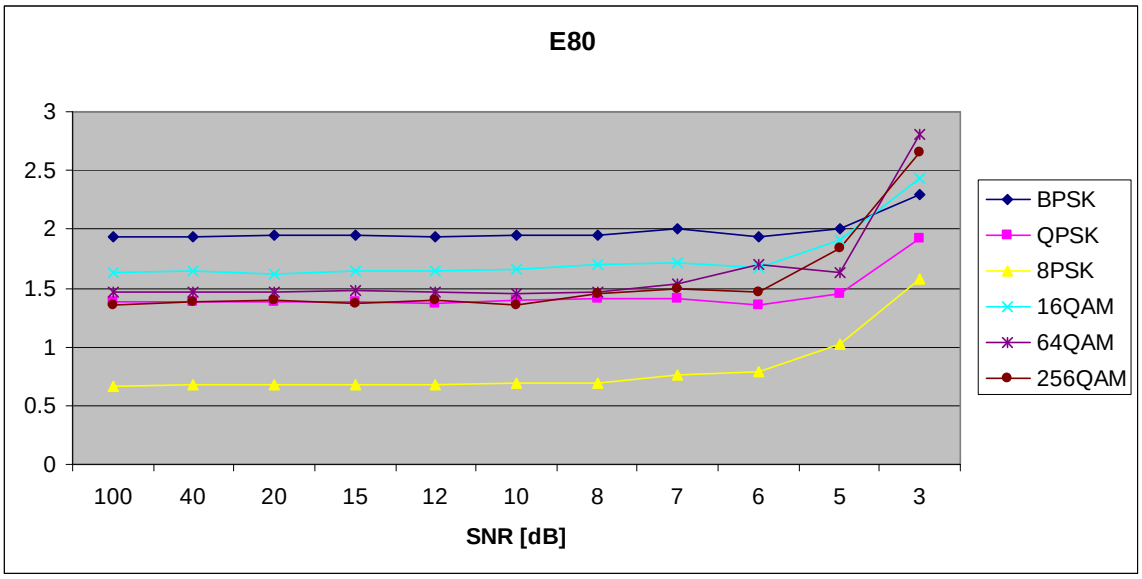

Figure 130. $E_{x,8,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

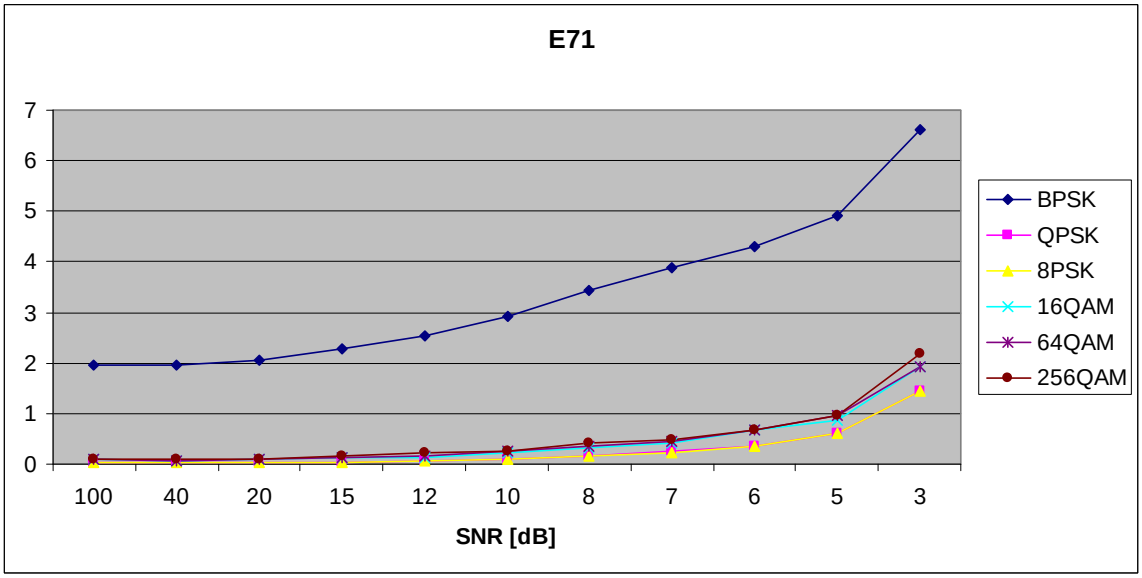

Figure 131. $E_{x,7,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

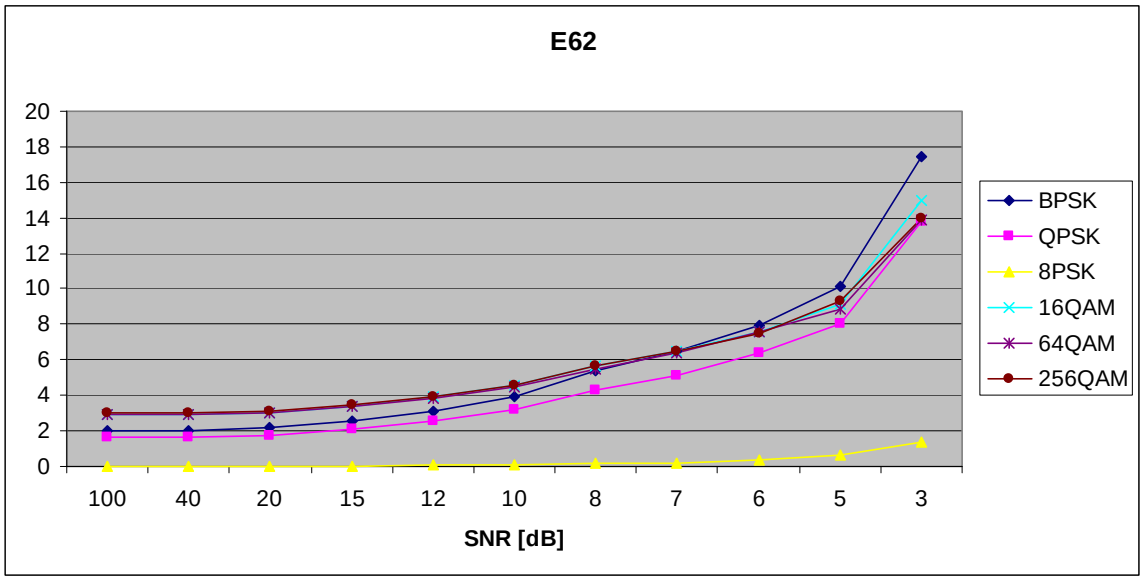

Figure 132. $E_{x,6,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

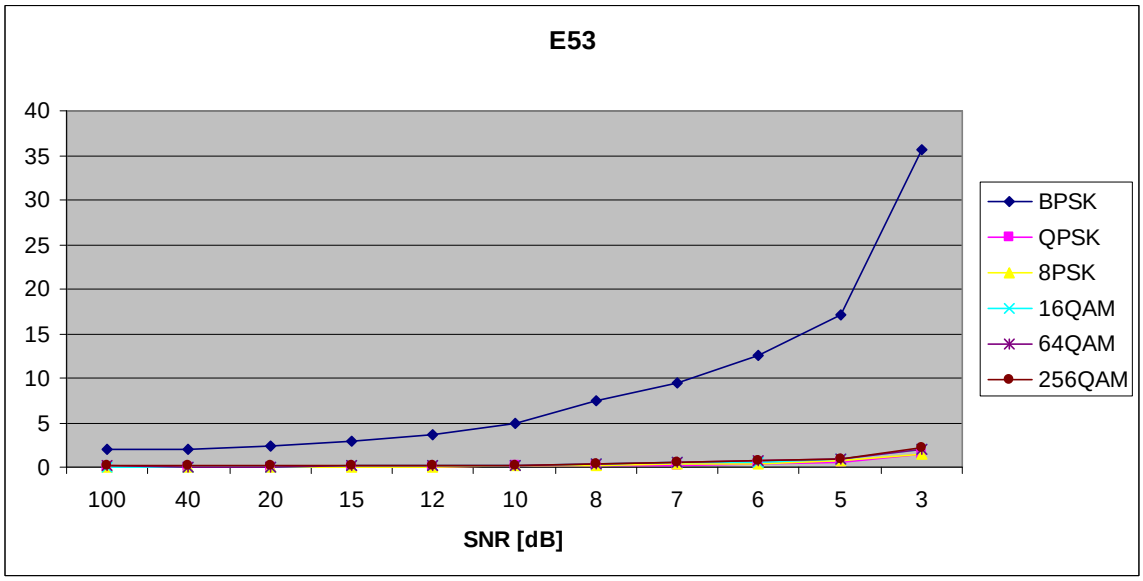

Figure 133. $E_{x,5,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

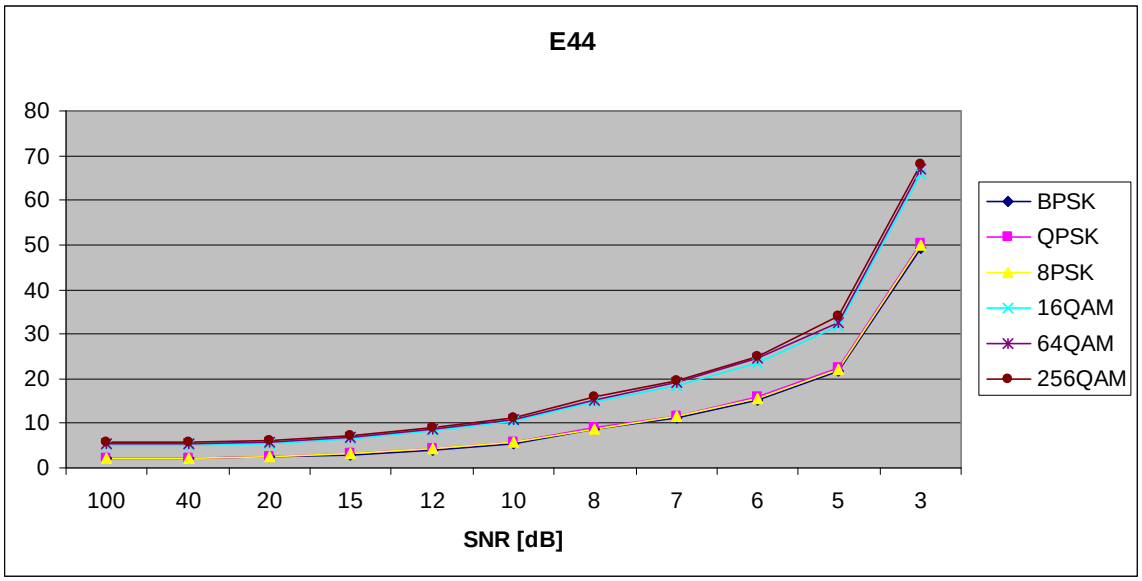

Figure 134. $E_{x,4,4}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

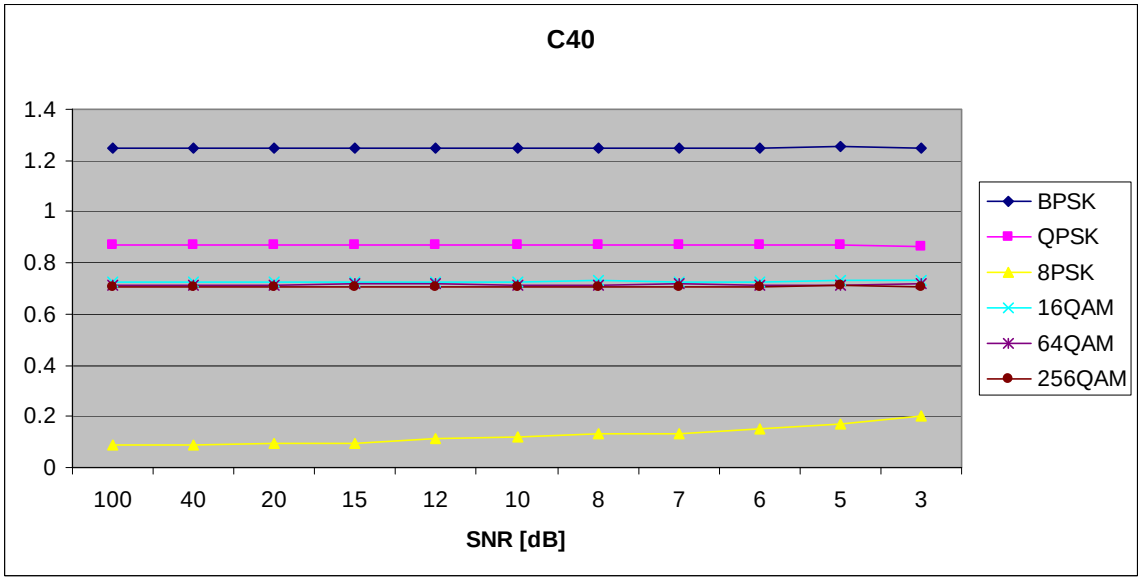

Figure 135. $C_{x,4,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

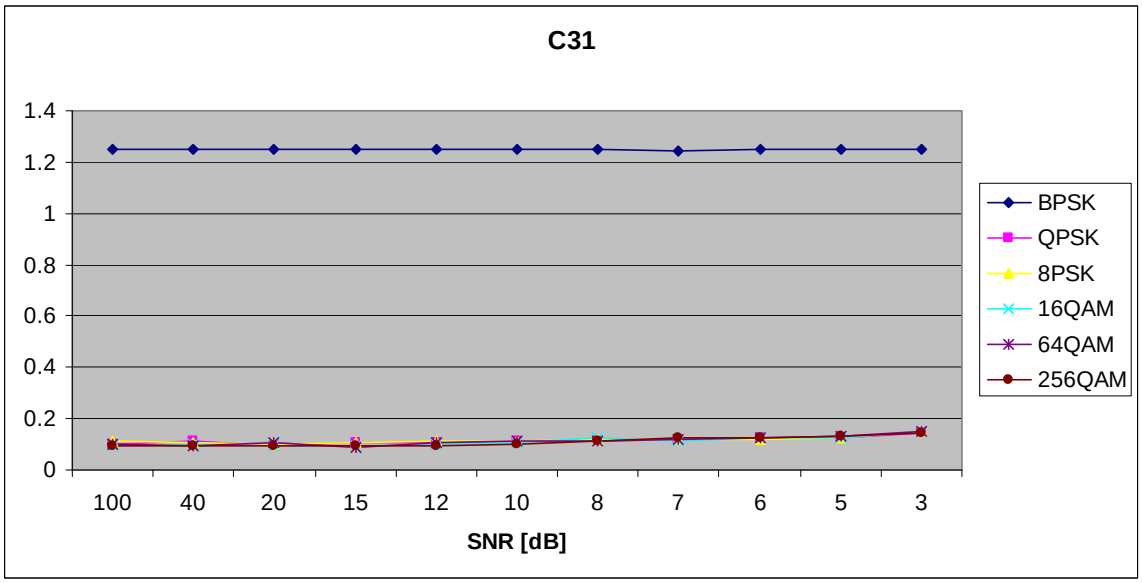

Figure 136. $C_{x,3,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

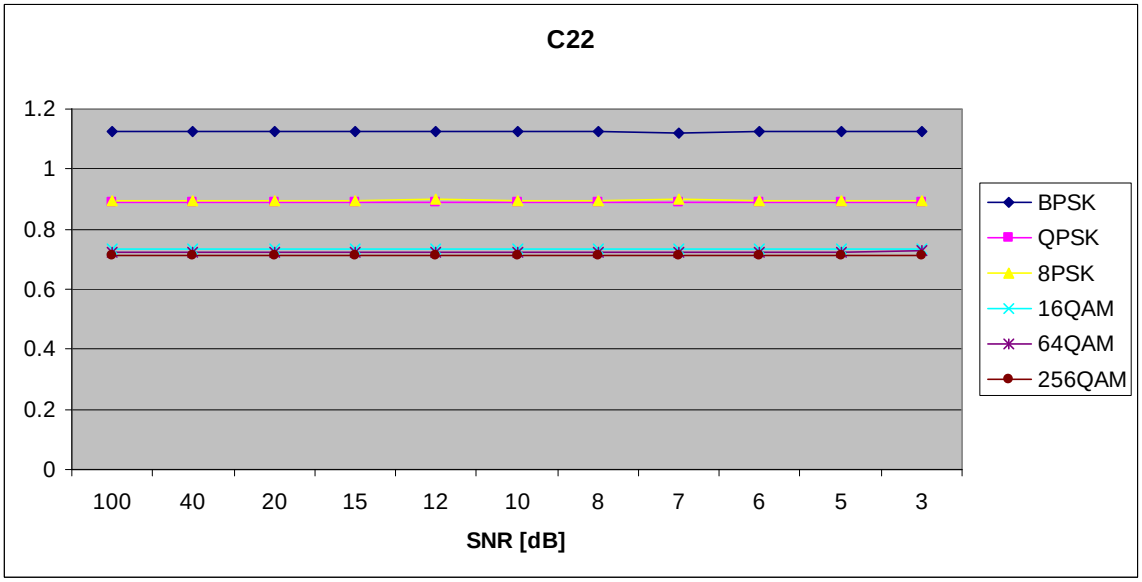

Figure 137. $C_{x,2,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

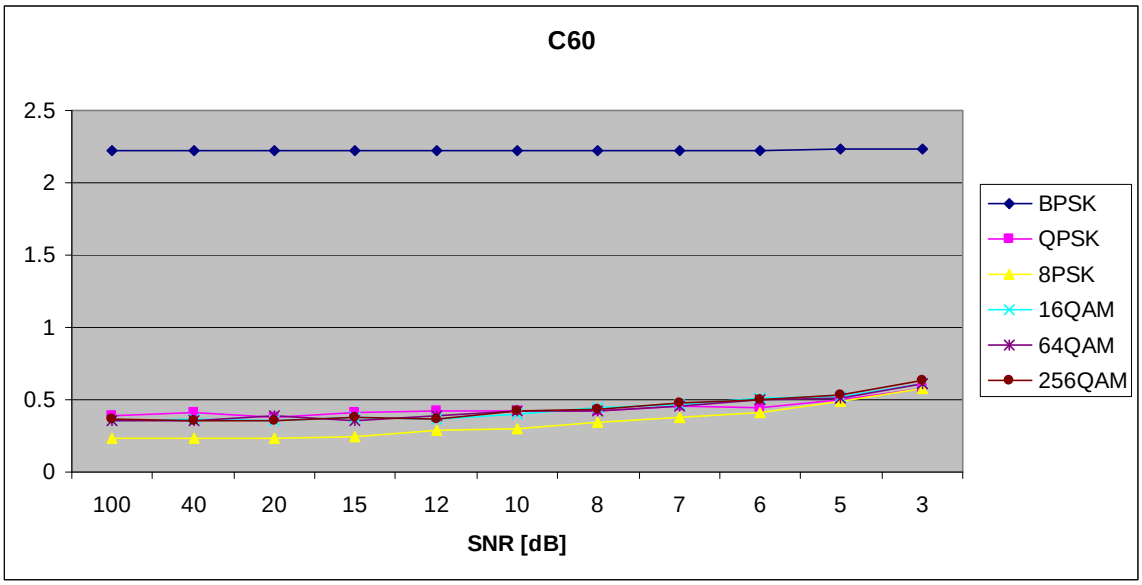

Figure 138. $C_{x,6,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

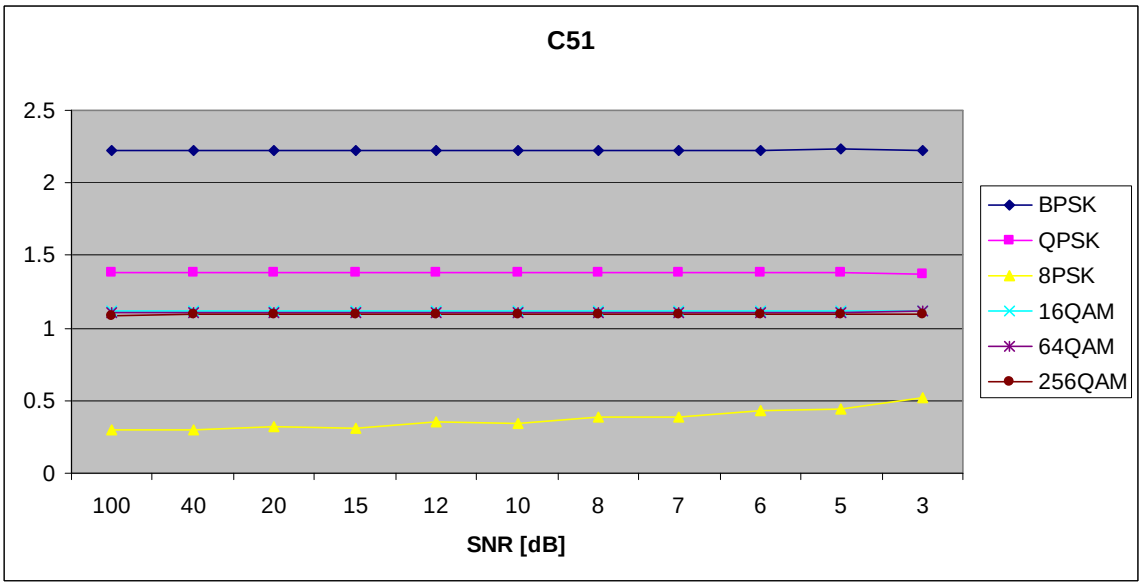

Figure 139. $C_{x,5,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

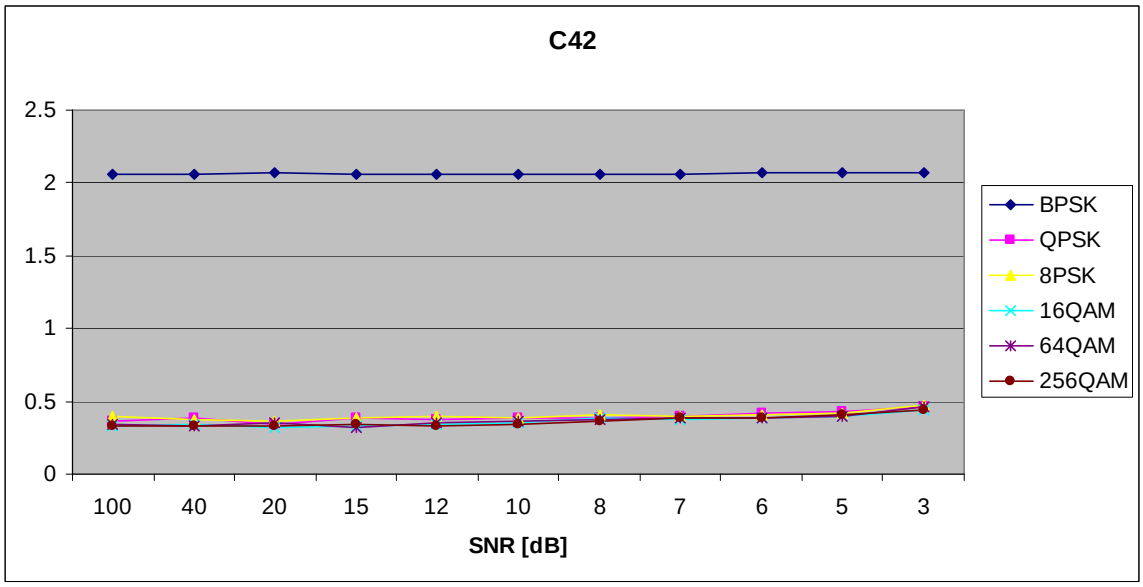

Figure 140. $C_{x,4,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

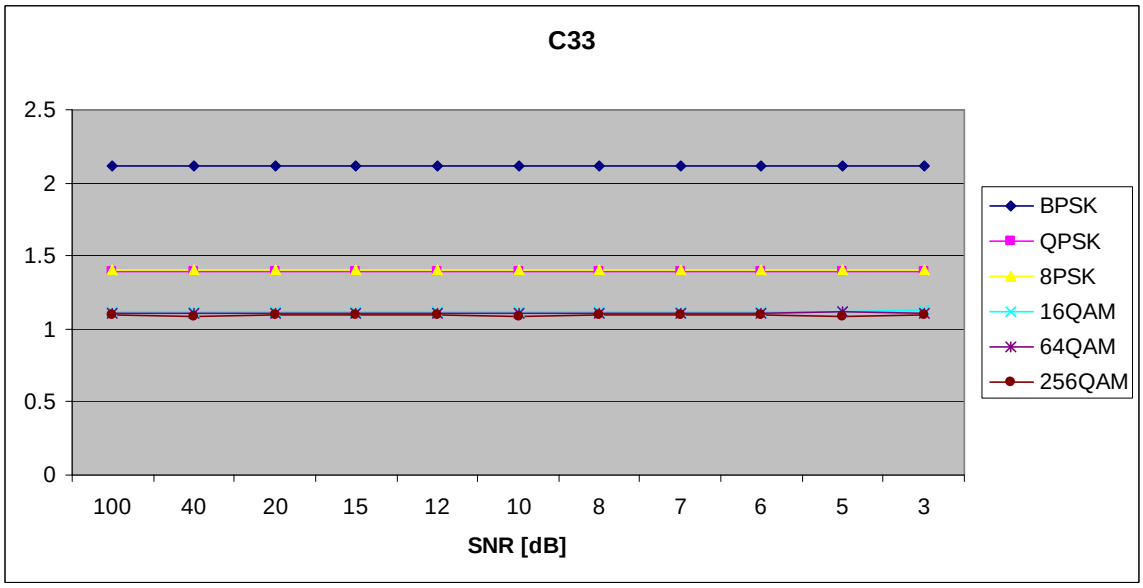

Figure 141. $C_{x,3,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

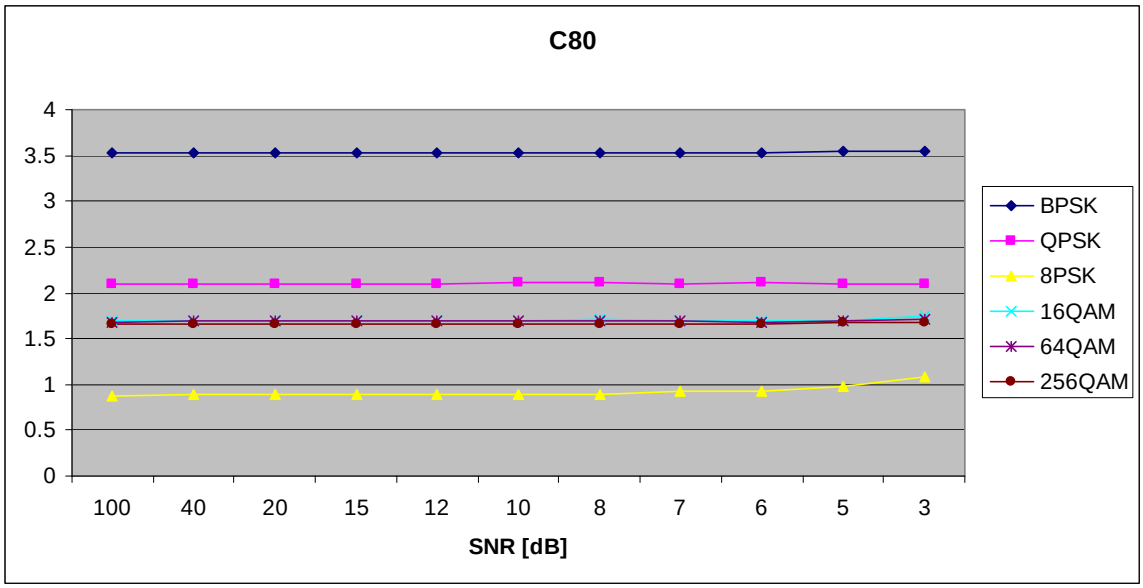

Figure 142. $C_{x,8,0}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

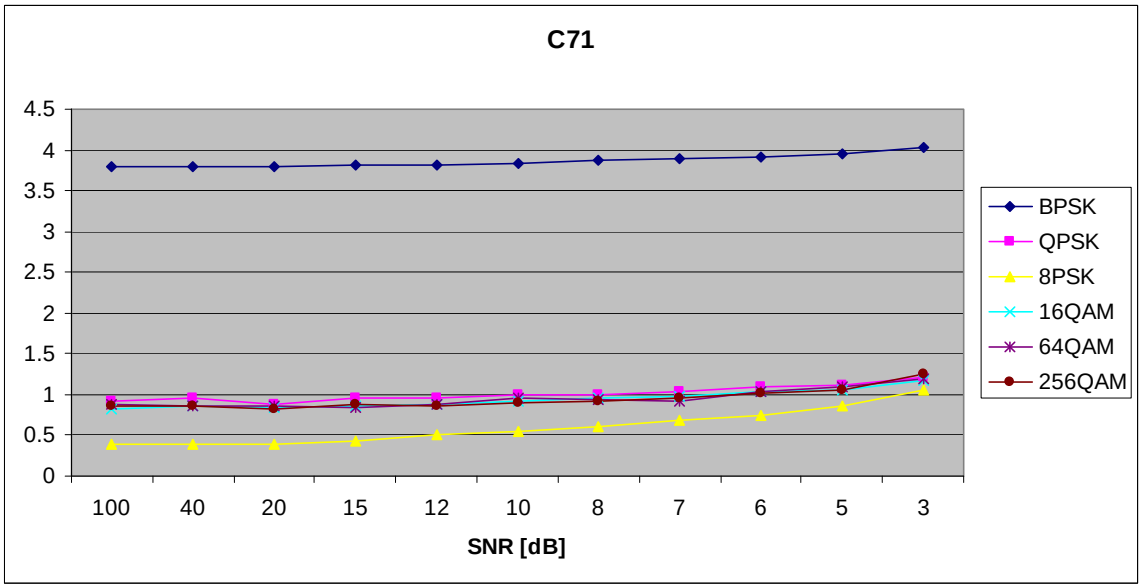

Figure 143. $C_{x,7,1}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

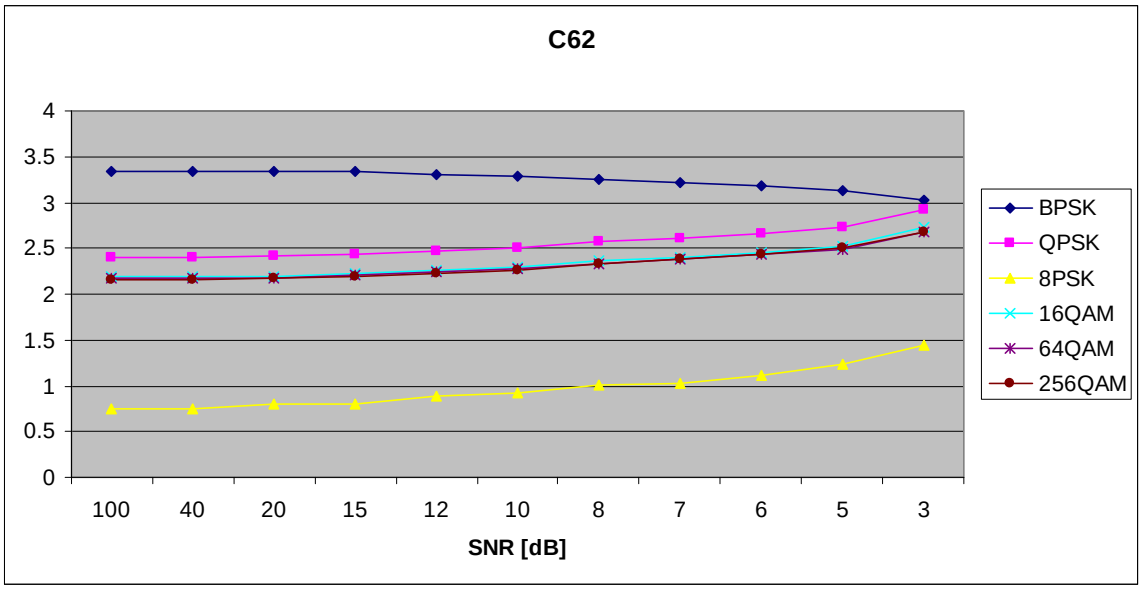

Figure 144. $C_{x,6,2}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

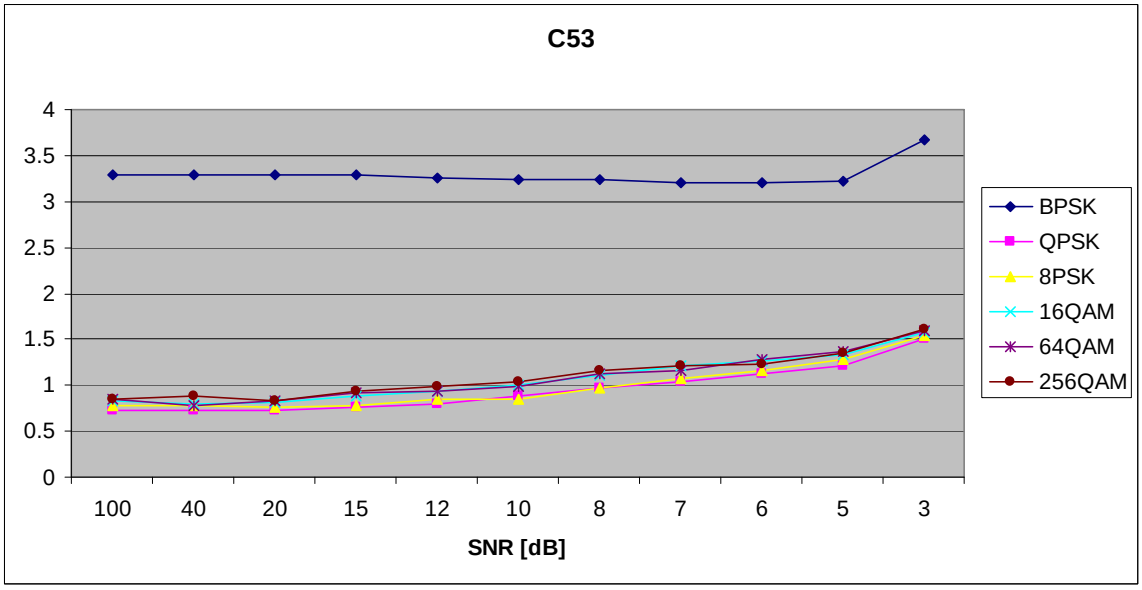

Figure 145. $C_{x,5,3}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

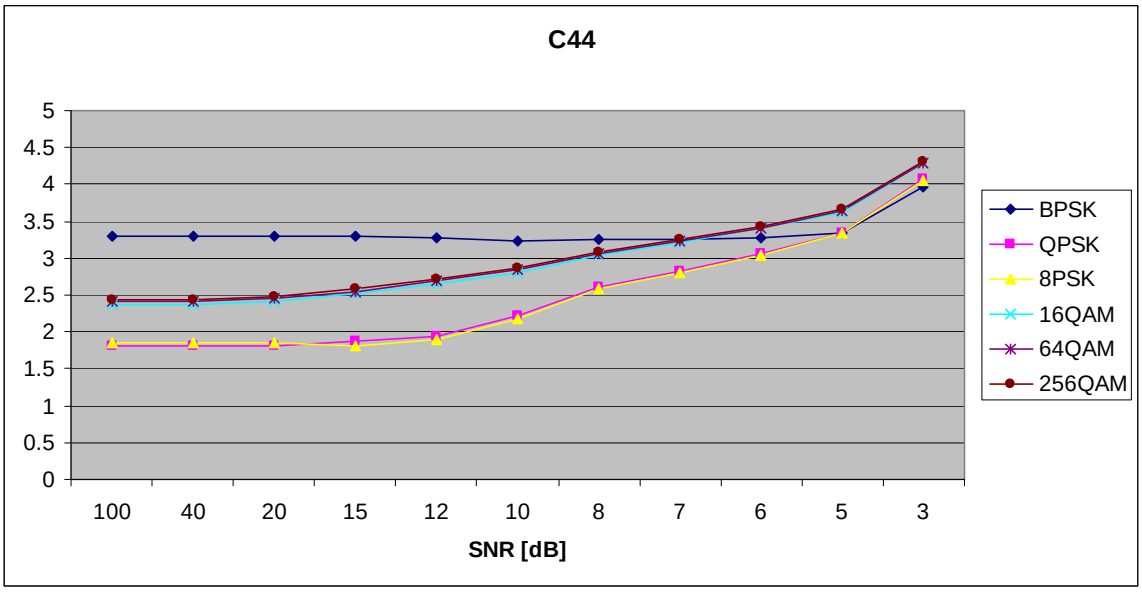

Figure 146. $C_{x,4,4}$ in AWGN and Slow, Frequency-Selective Ricean Fading.

F. AWGN PLUS FAST, FREQUENCY-FLAT RAYLEIGH FADING

Parameters for the rayleighchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 5000 Hz
- Path Delays: [0, 1×10^{-7}]
- Average Path Gains: [0, -10]

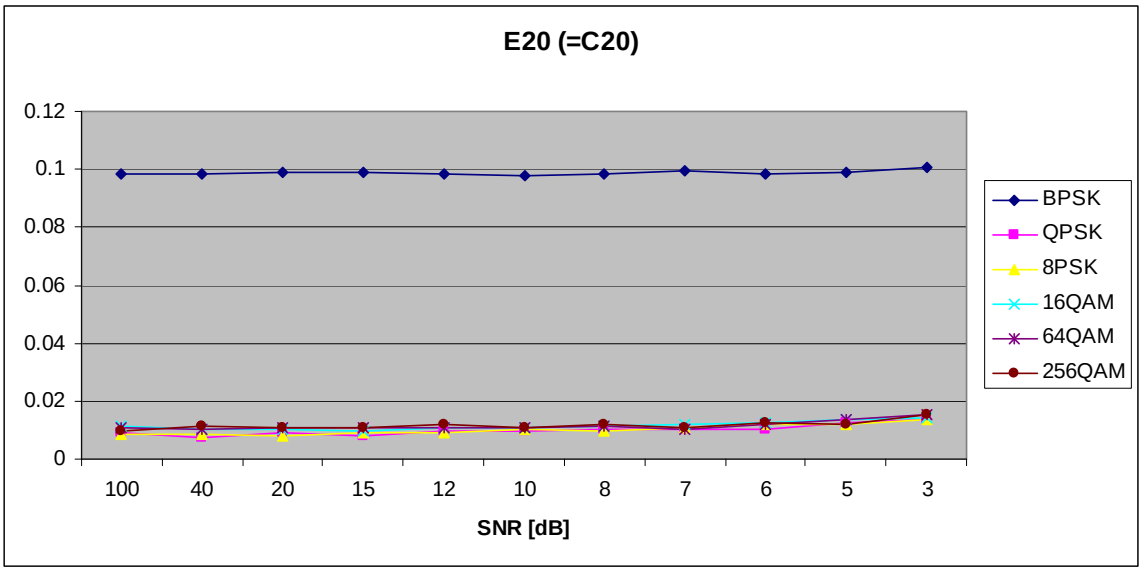

Figure 147. $E_{x,2,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

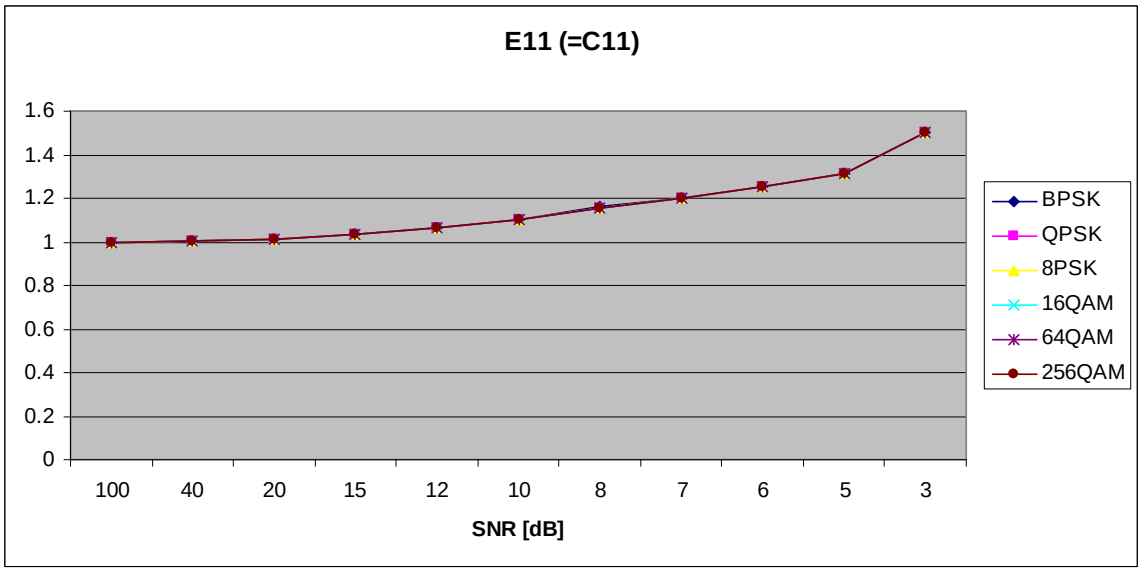

Figure 148. $E_{x,1,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

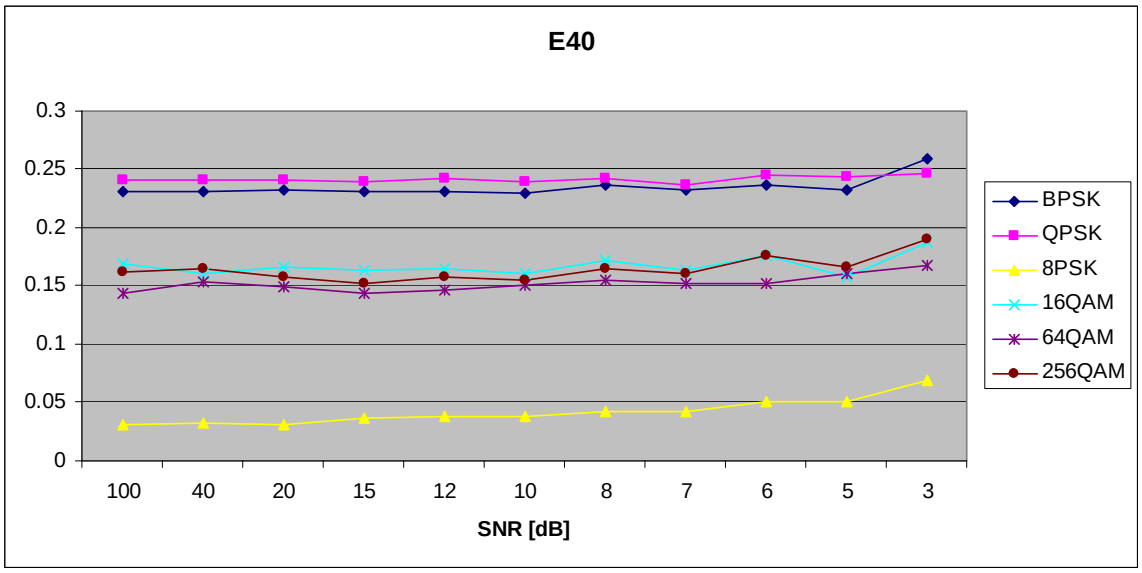

Figure 149. $E_{x,4,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

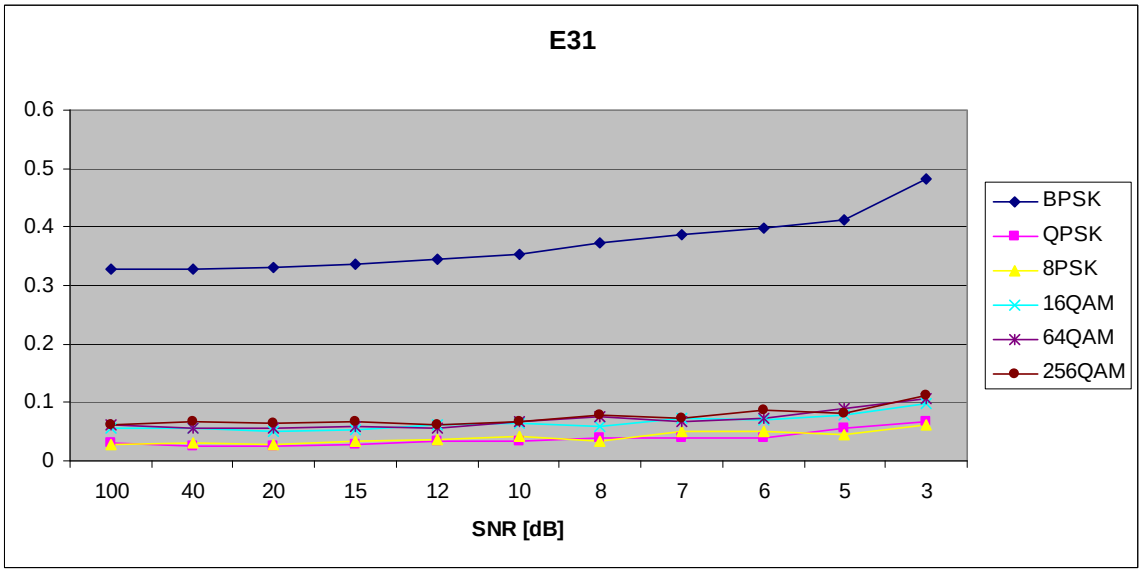

Figure 150. $E_{x,3,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

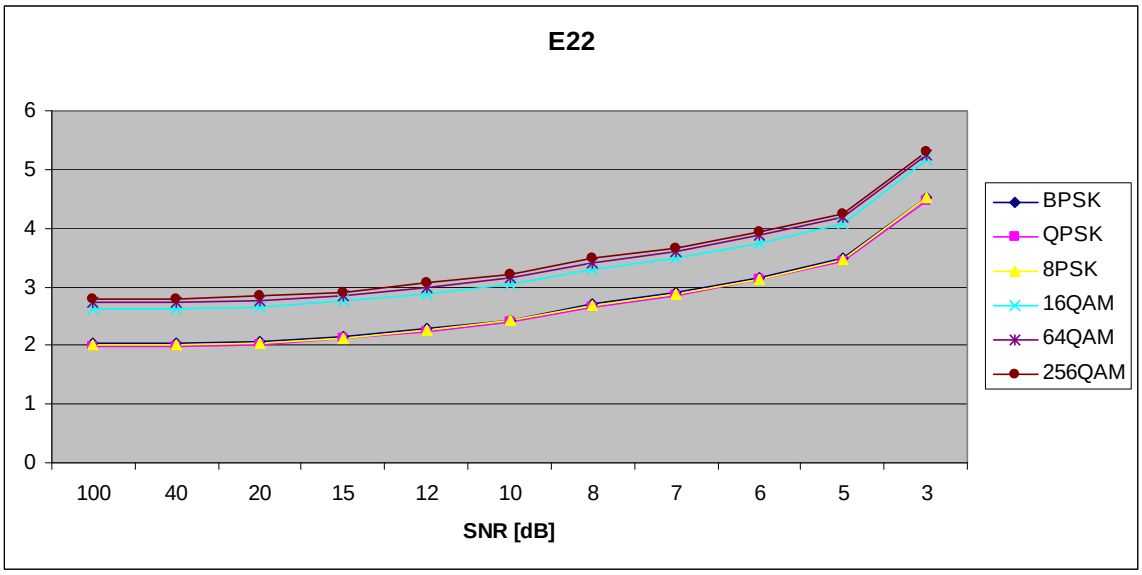

Figure 151. $E_{x,2,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

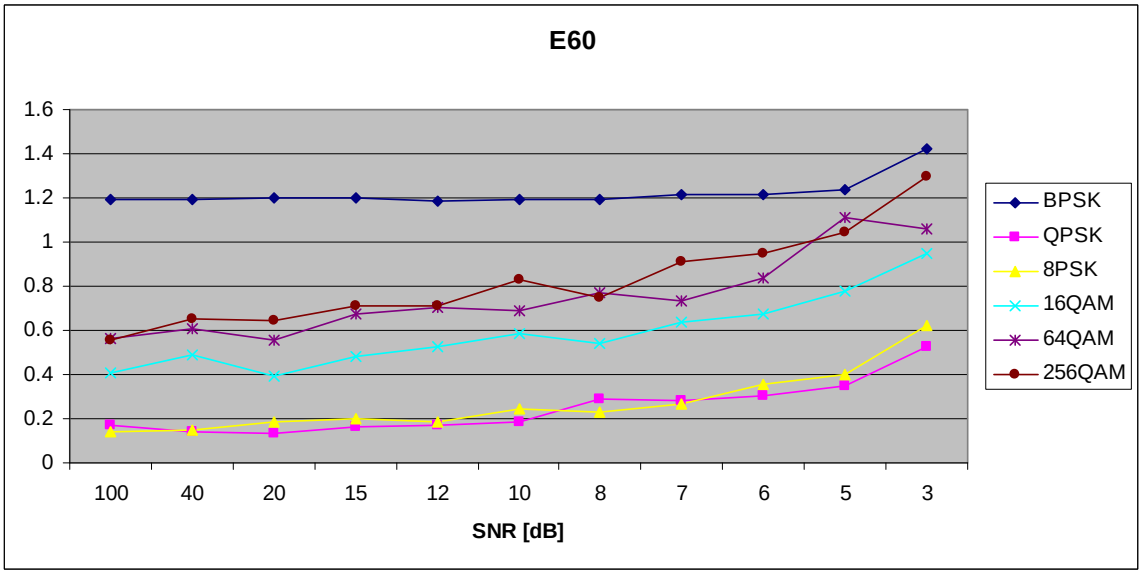

Figure 152. $E_{x,6,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

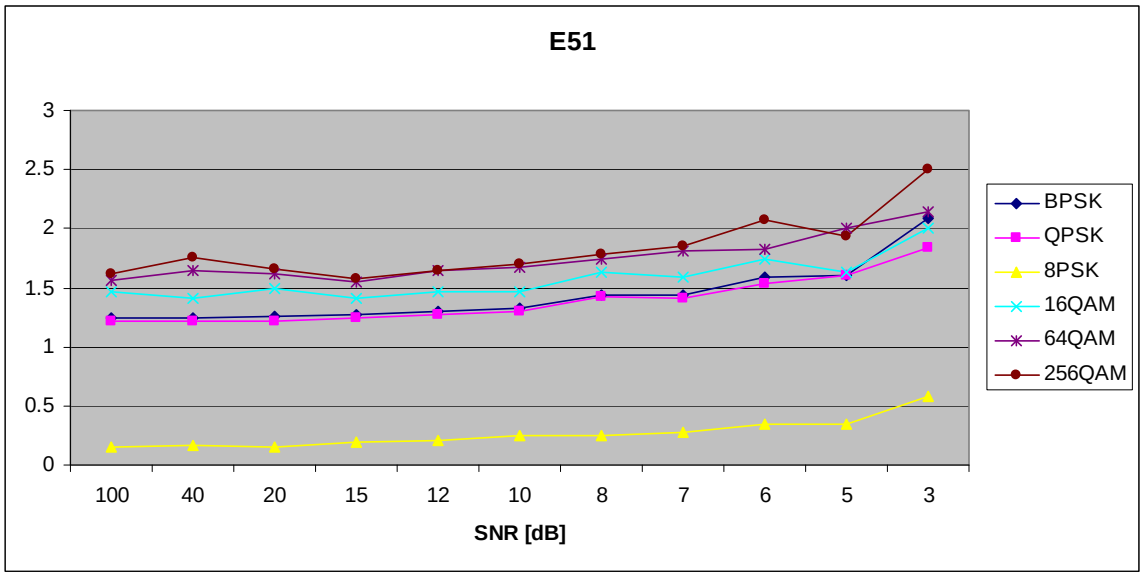

Figure 153. $E_{x,5,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

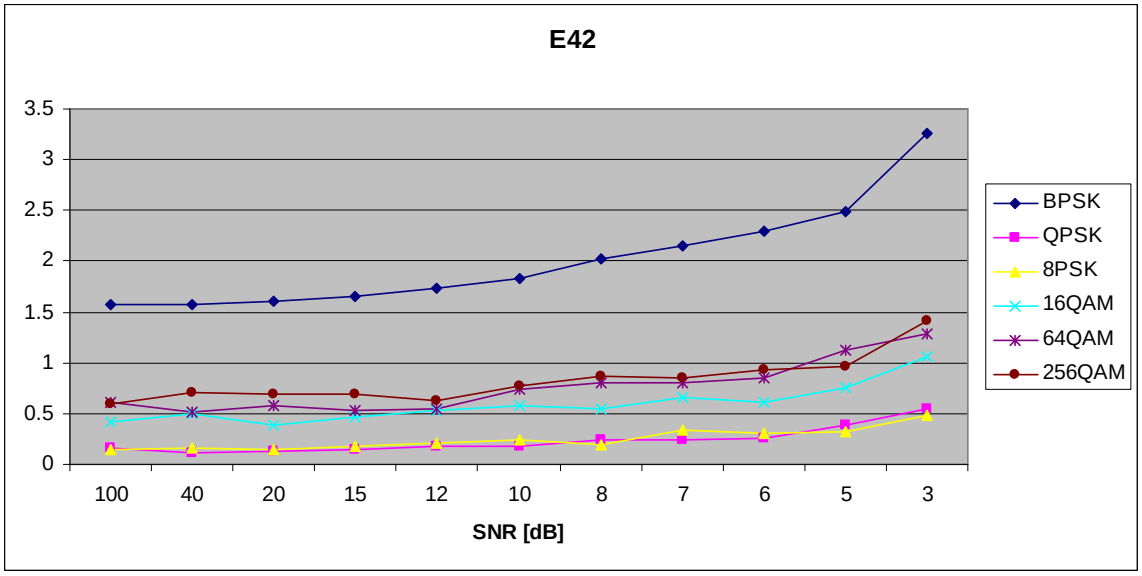

Figure 154. $E_{x,4,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

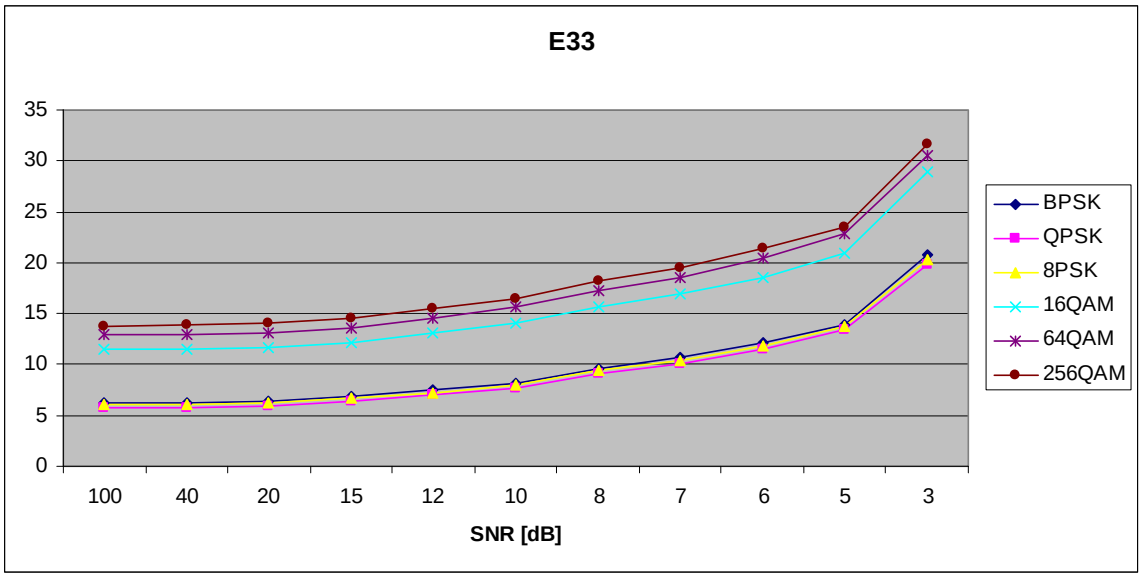

Figure 155. $E_{x,3,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

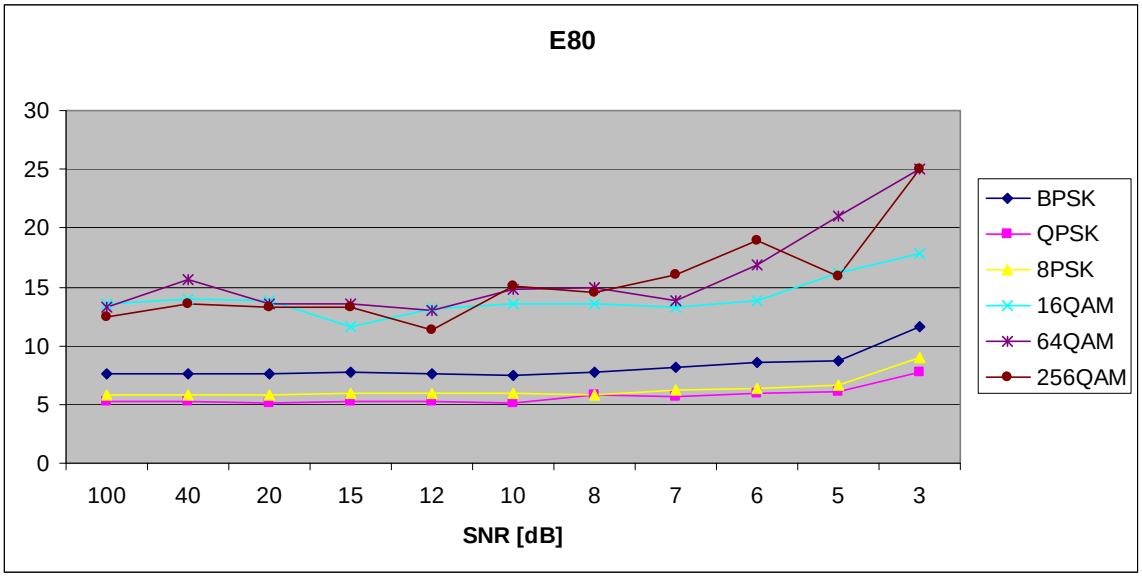

Figure 156. $E_{x,8,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

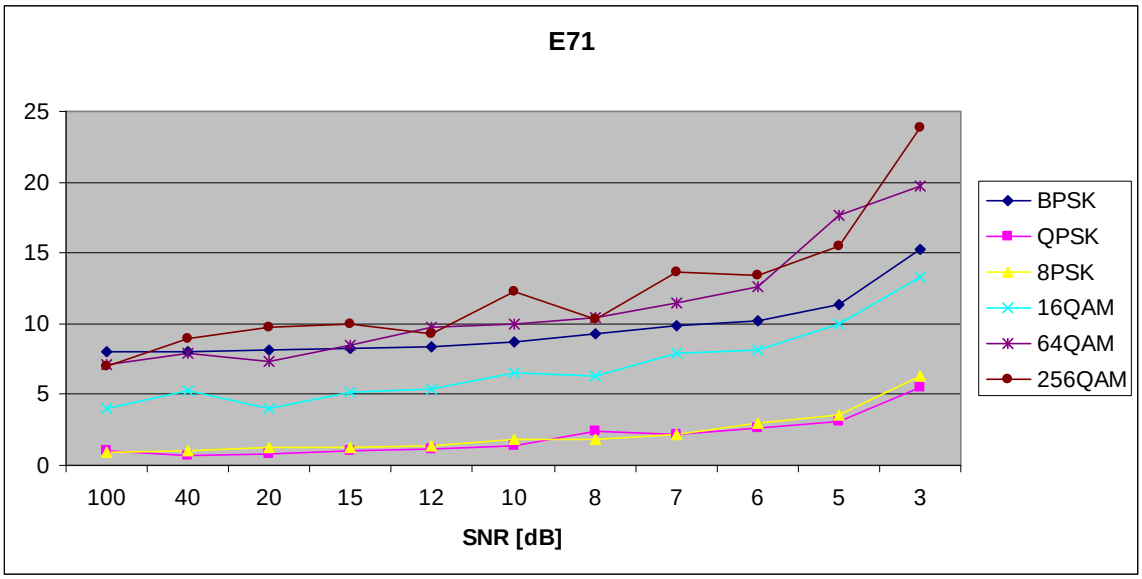

Figure 157. $E_{x,7,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

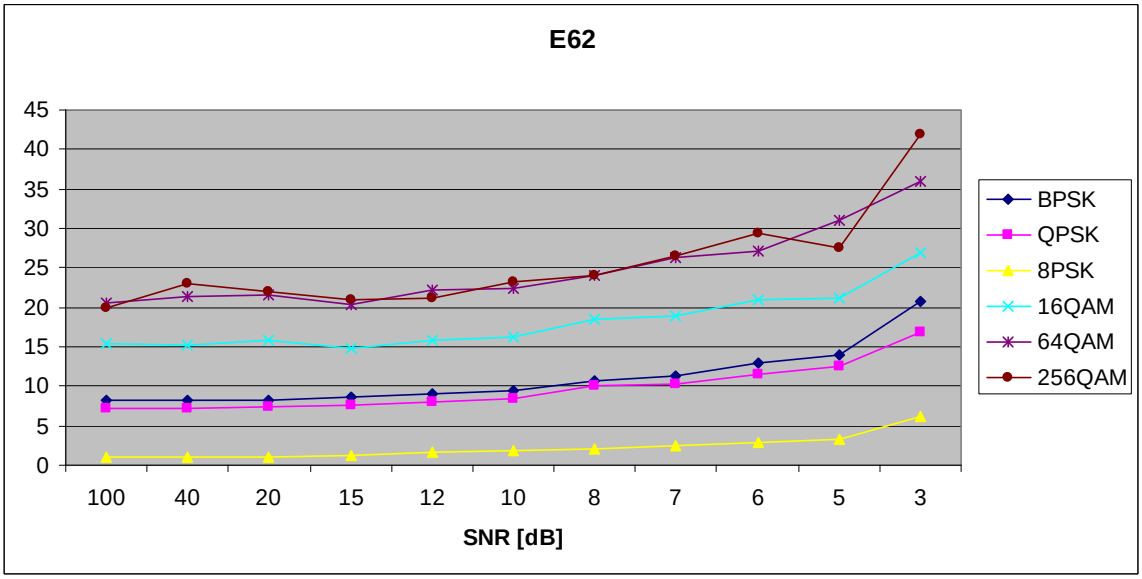

Figure 158. $E_{x,6,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

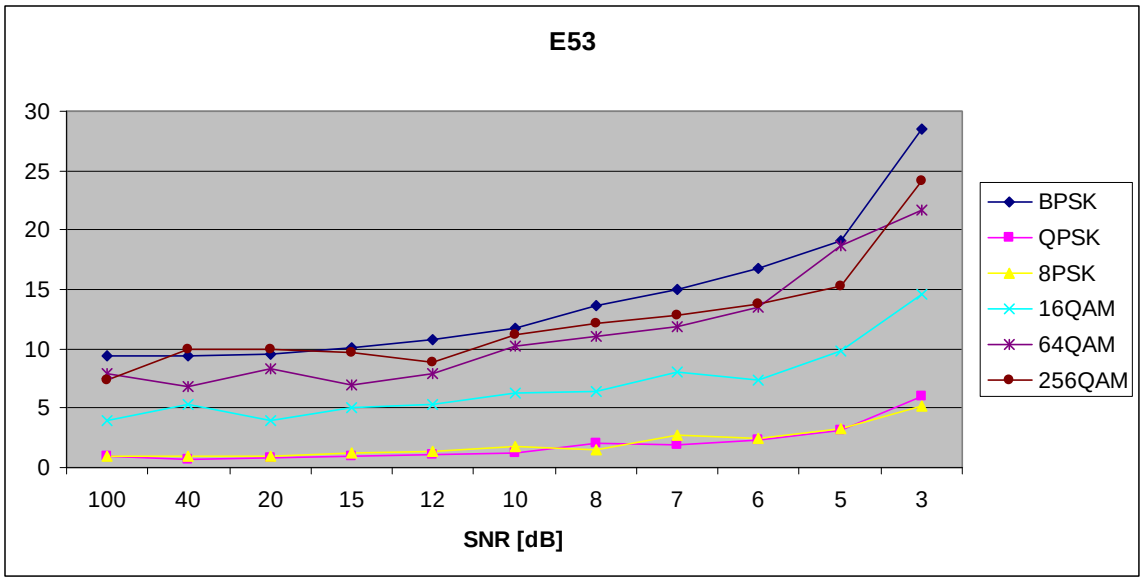

Figure 159. $E_{x,5,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

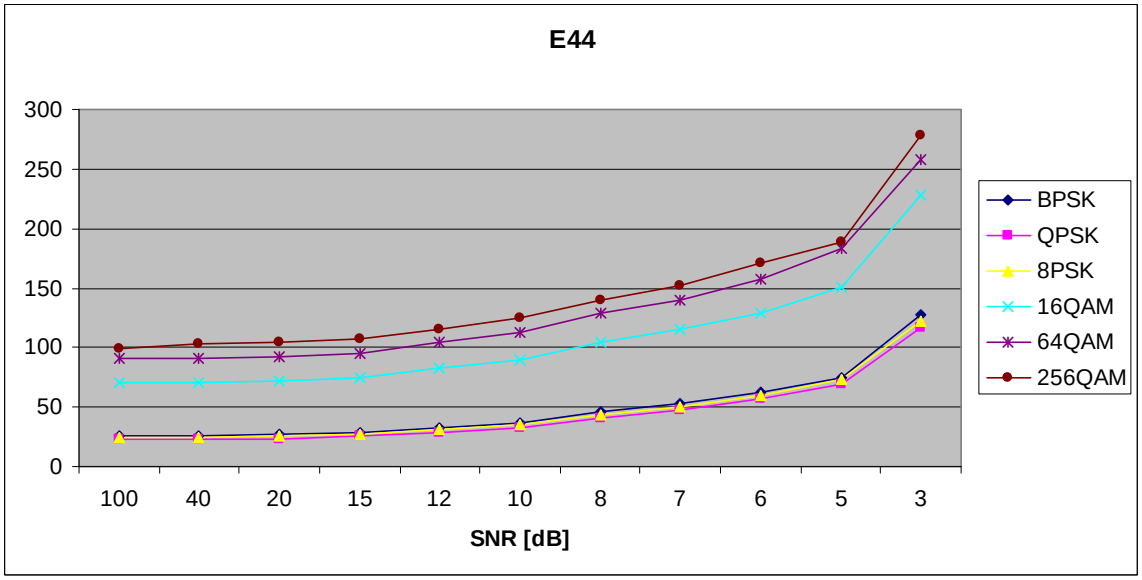

Figure 160. $E_{x,4,4}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

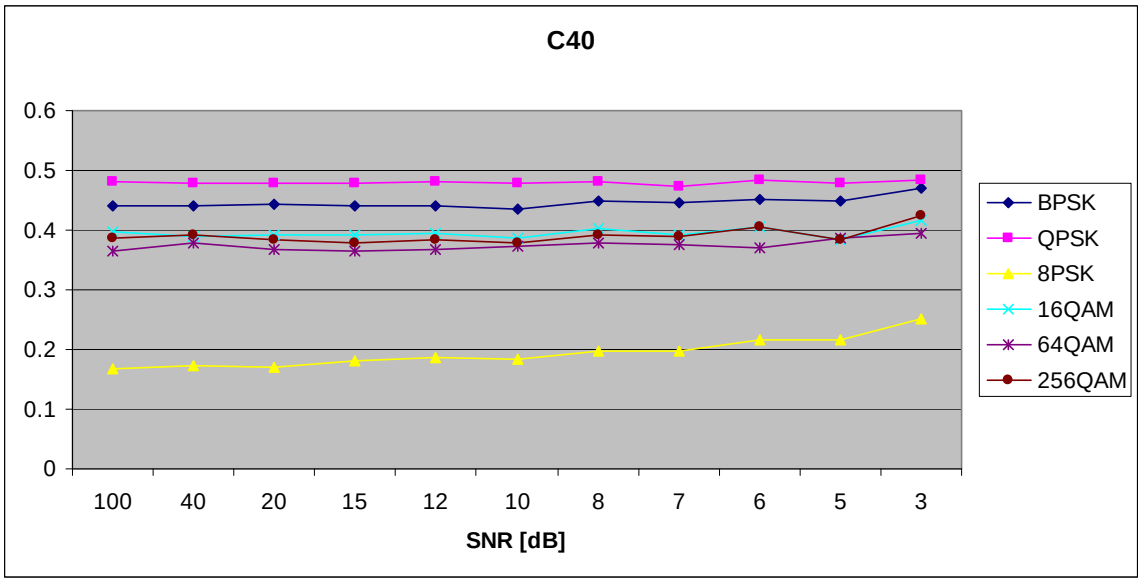

Figure 161. $C_{x,4,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

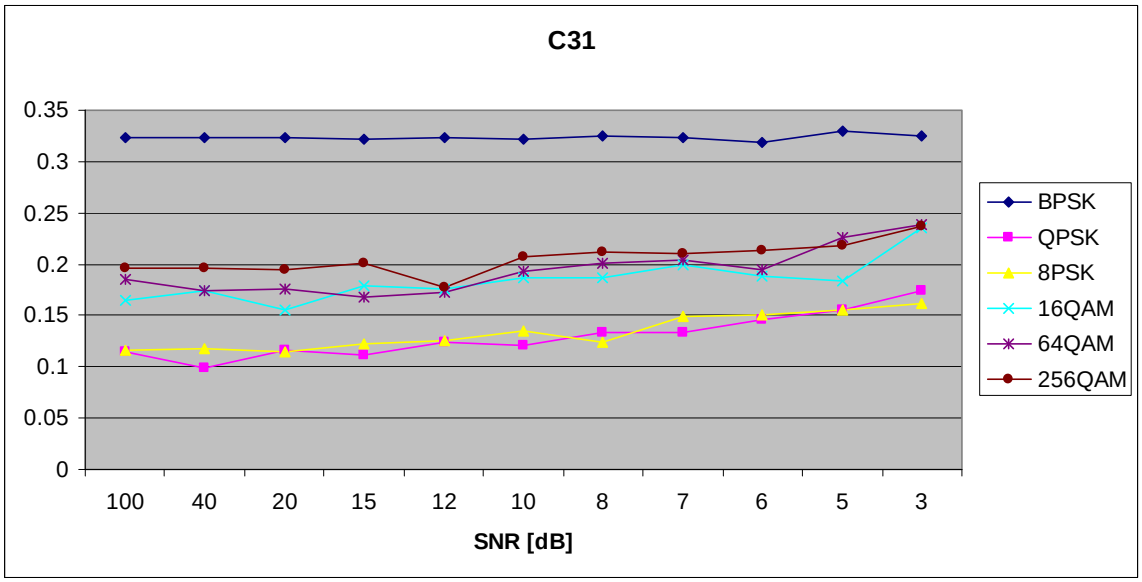

Figure 162. $C_{x,3,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

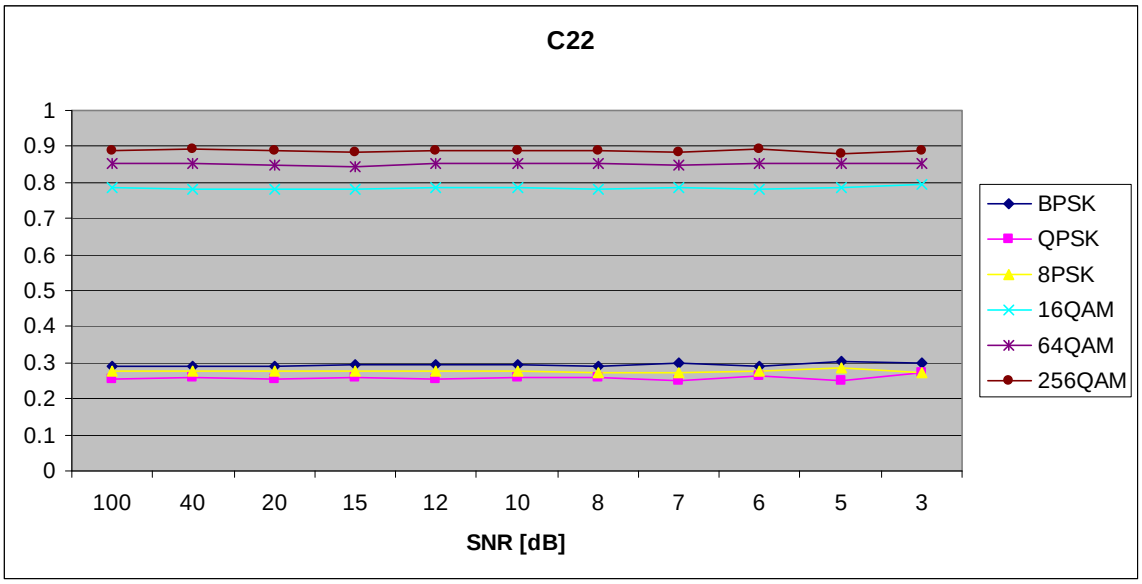

Figure 163. $C_{x,2,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

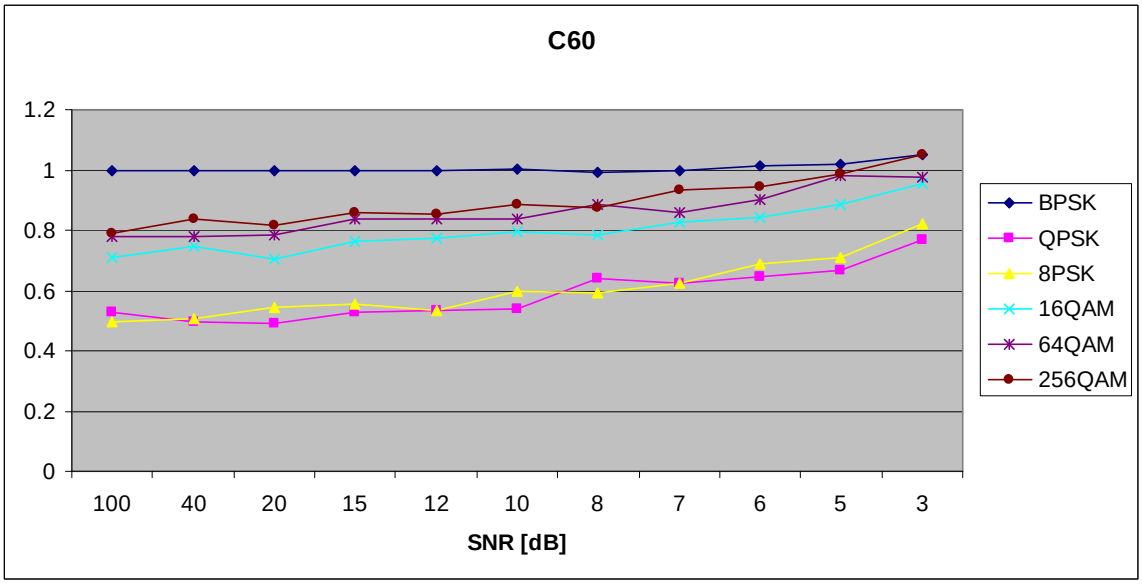

Figure 164. $C_{x,6,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

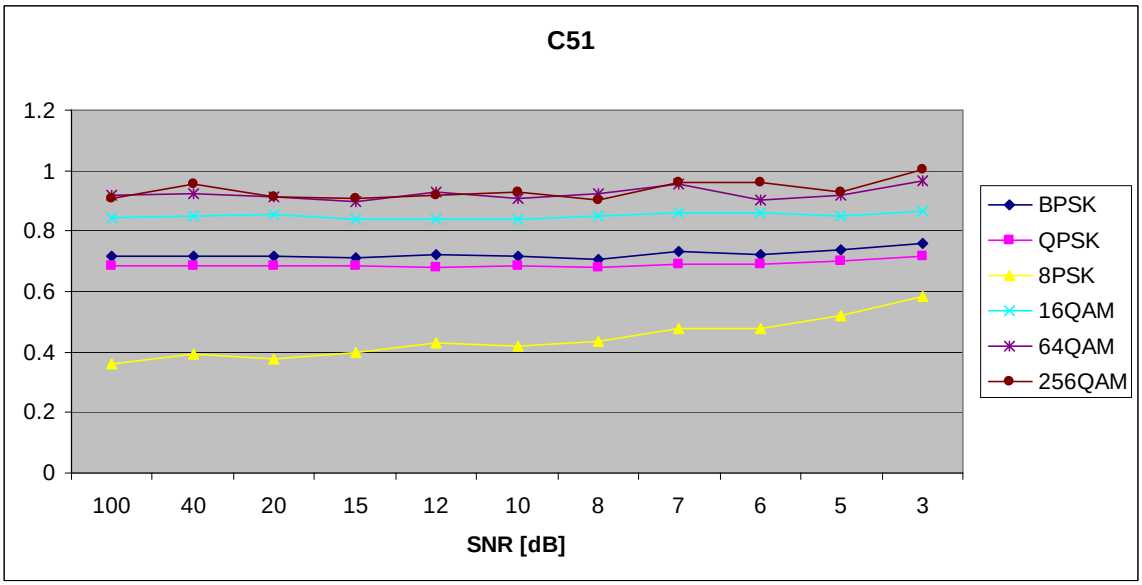

Figure 165. $C_{x,5,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

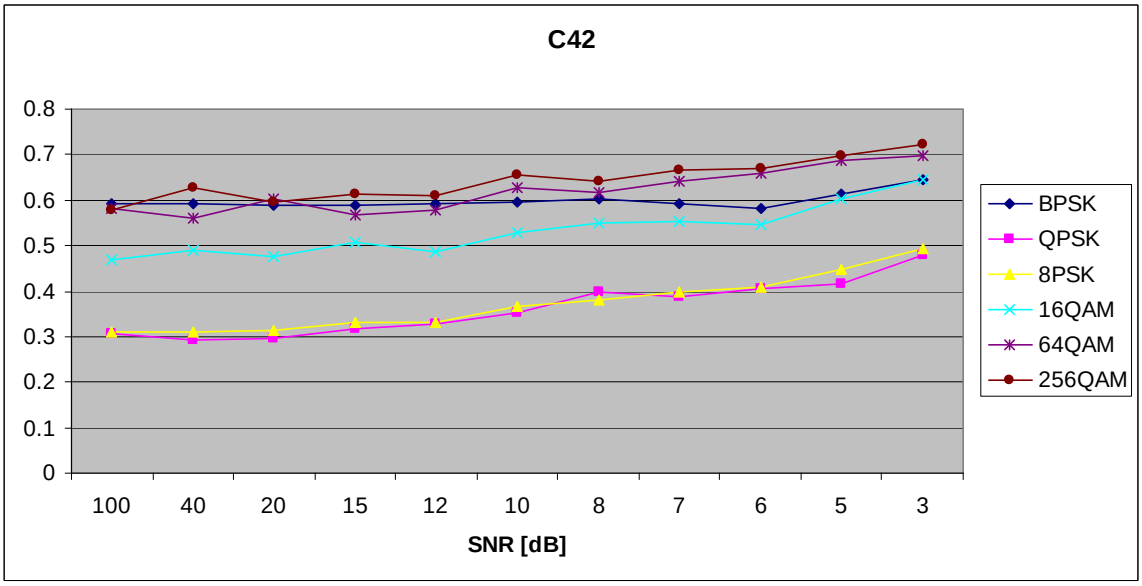

Figure 166. $C_{x,4,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

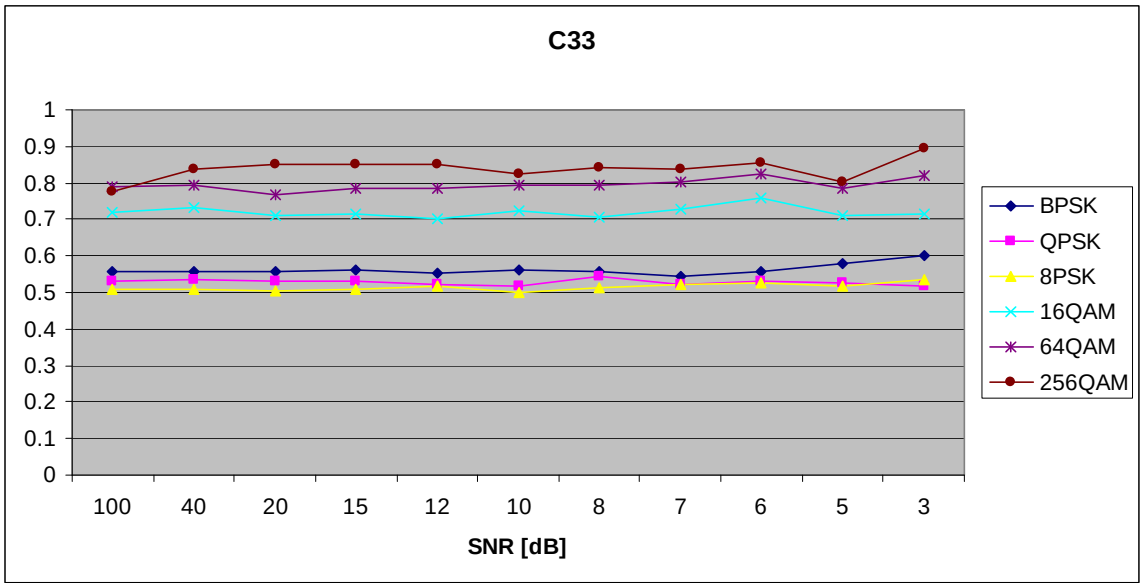

Figure 167. $C_{x,3,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

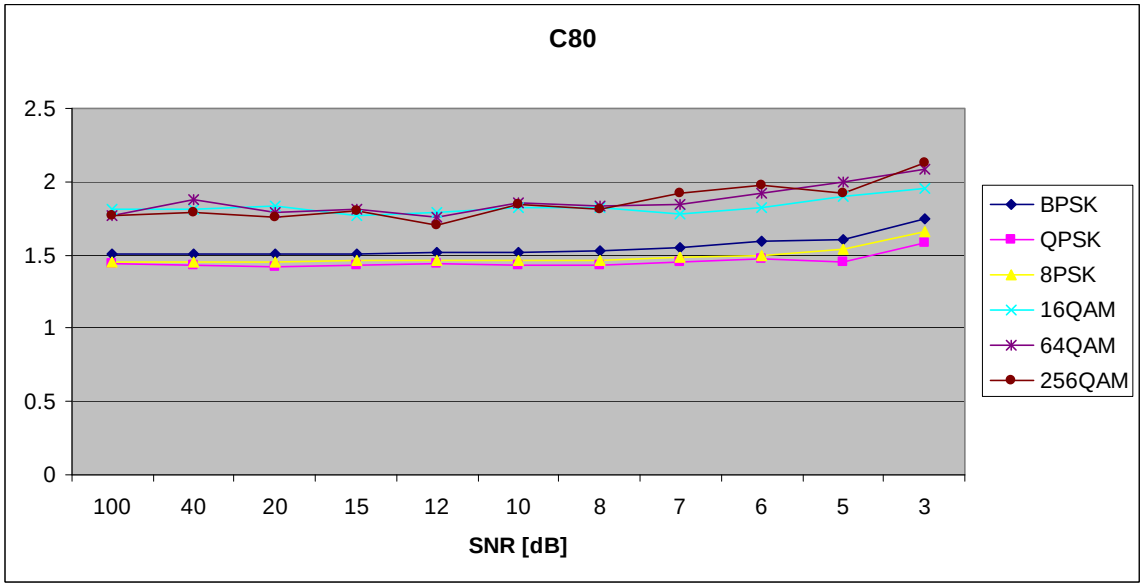

Figure 168. $C_{x,8,0}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

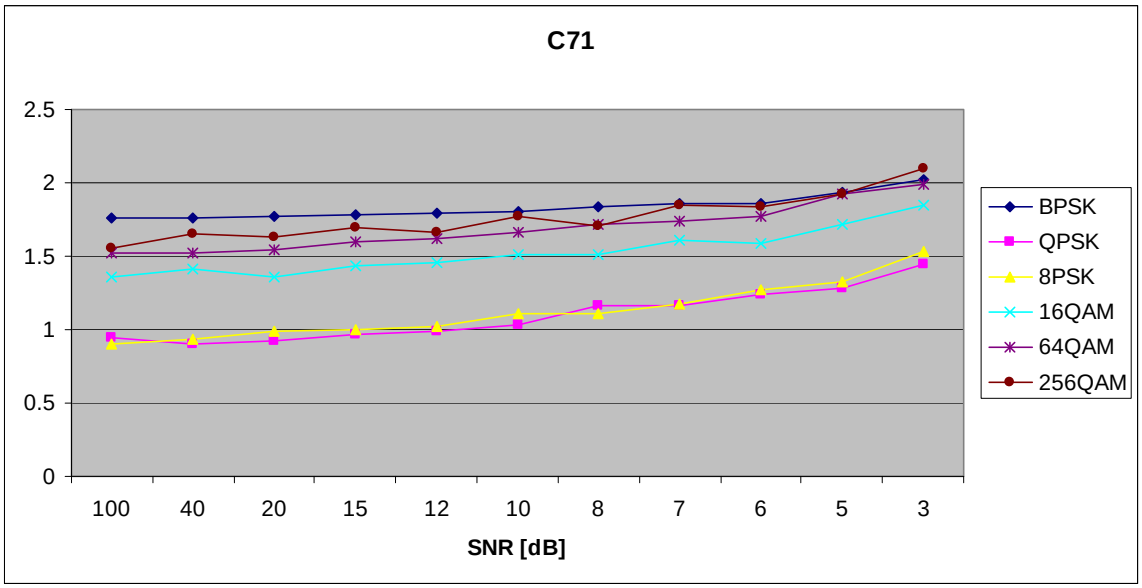

Figure 169. $C_{x,7,1}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

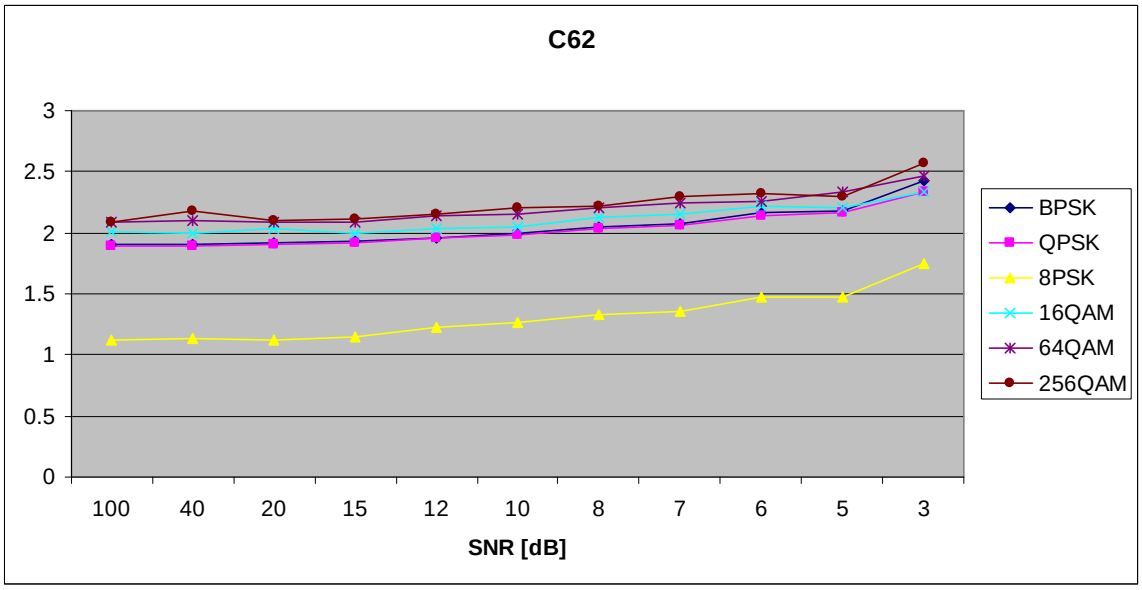

Figure 170. $C_{x,6,2}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

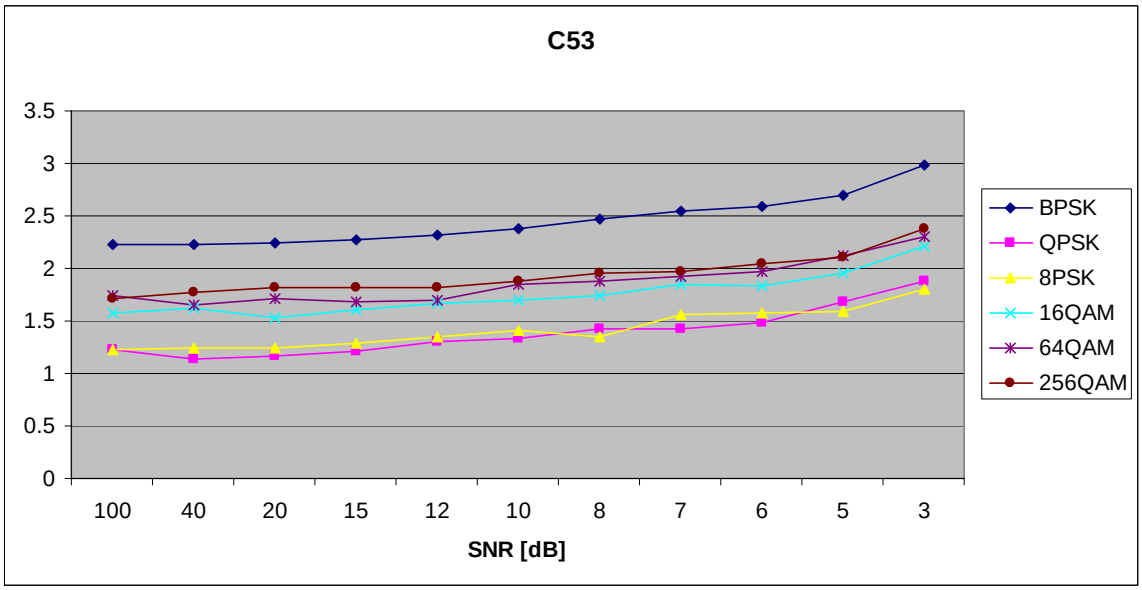

Figure 171. $C_{x,5,3}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

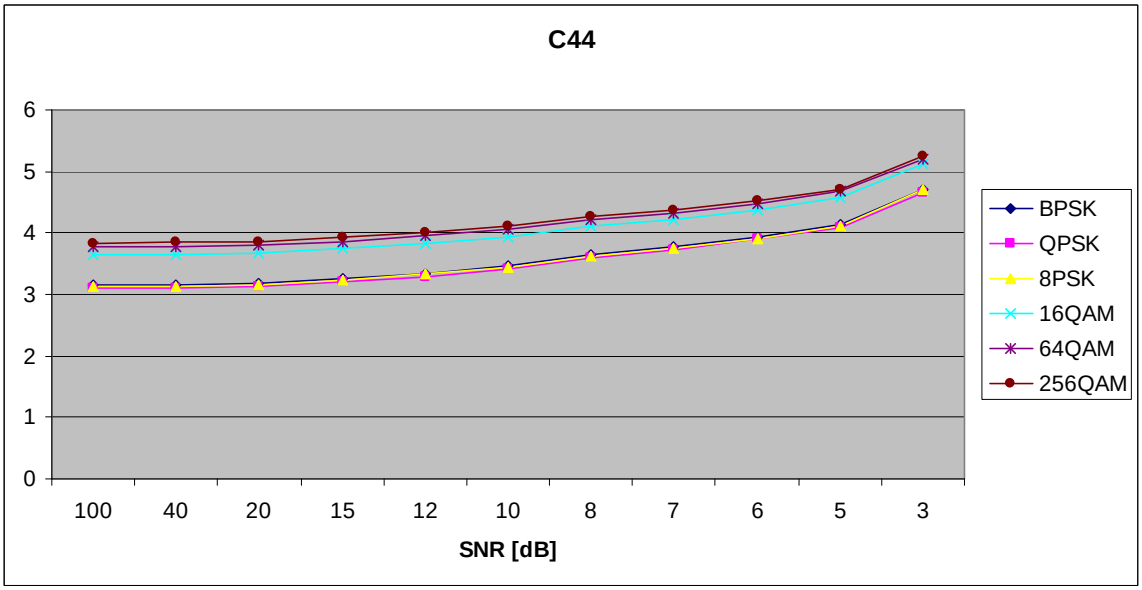

Figure 172. $C_{x,4,4}$ in AWGN and Fast, Frequency-Flat Rayleigh Fading.

G. AWGN PLUS FAST, FREQUENCY-FLAT RICEAN FADING

Parameters for the ricianchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 5000 Hz
- K-factor: 3
- Path Delays: [0, 1×10^{-7}]
- Average Path Gains: [0, -10]

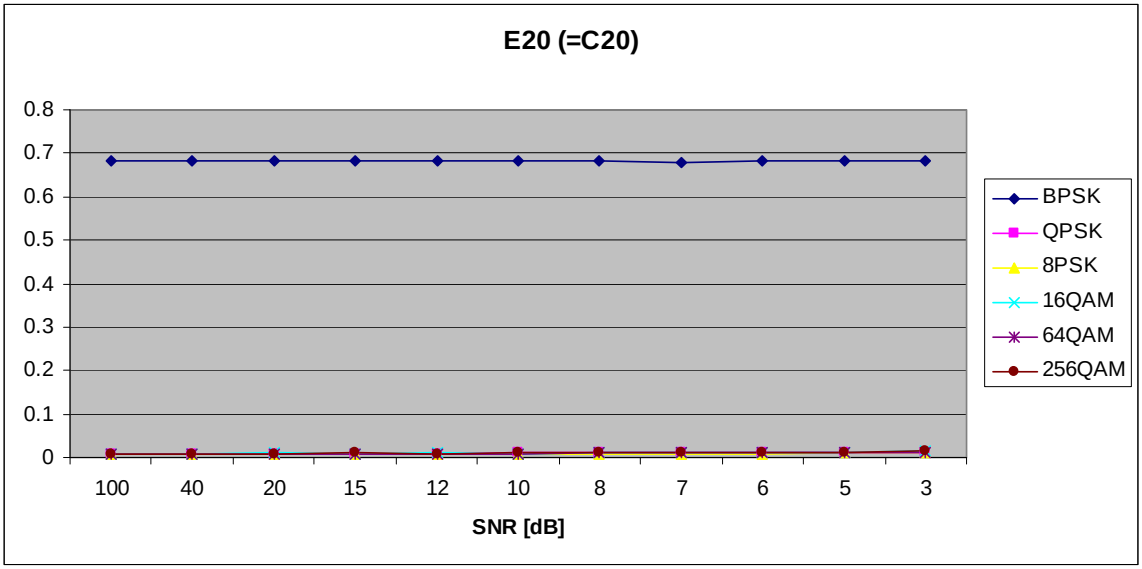

Figure 173. $E_{x,2,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

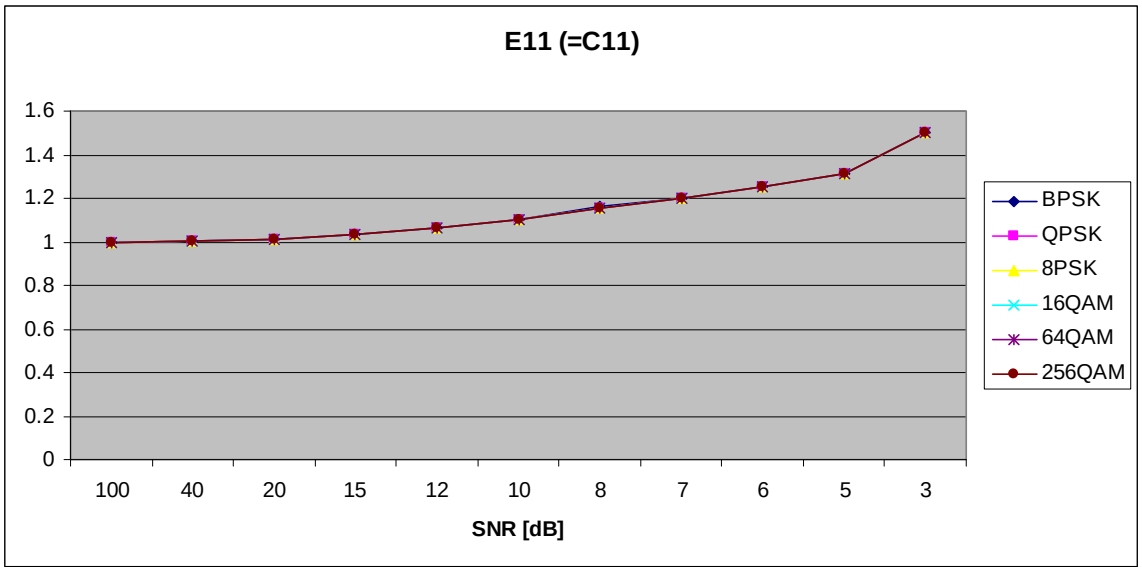

Figure 174. $E_{x,1,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

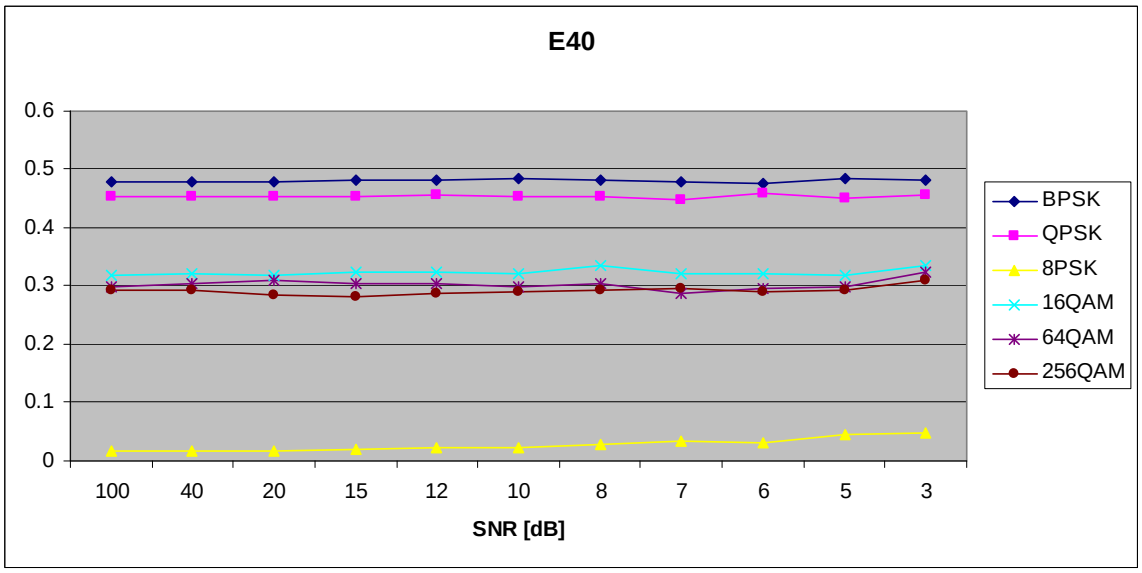

Figure 175. $E_{x,4,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

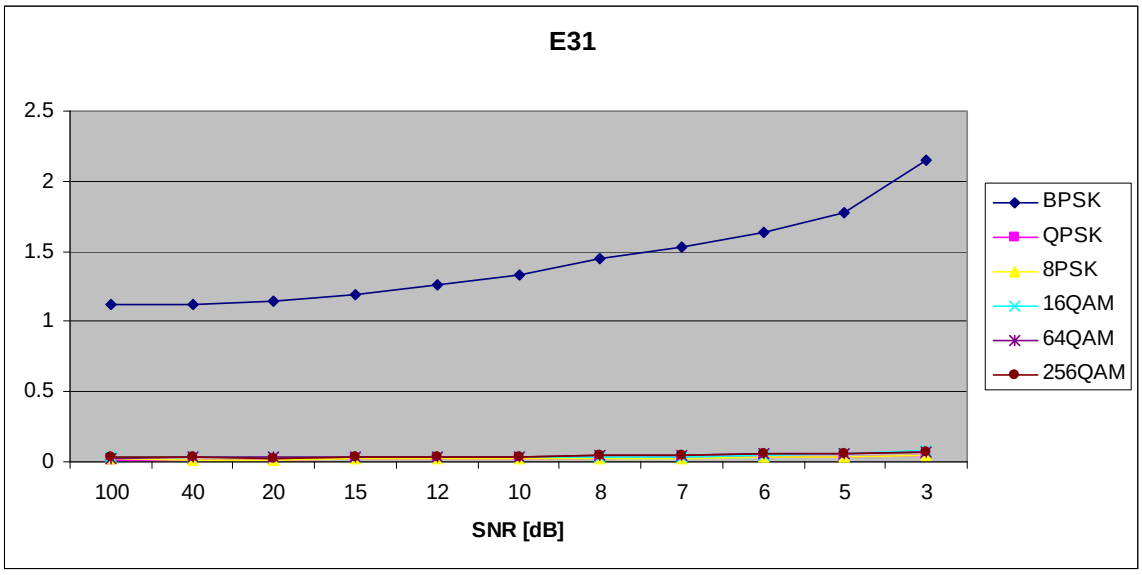

Figure 176. $E_{x,3,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

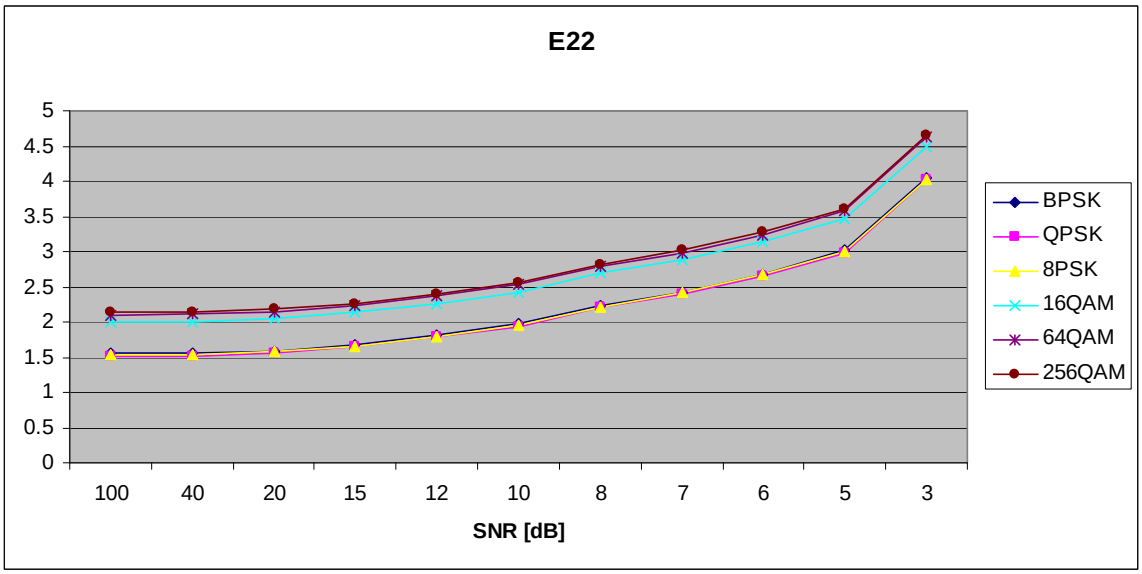

Figure 177. $E_{x,2,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

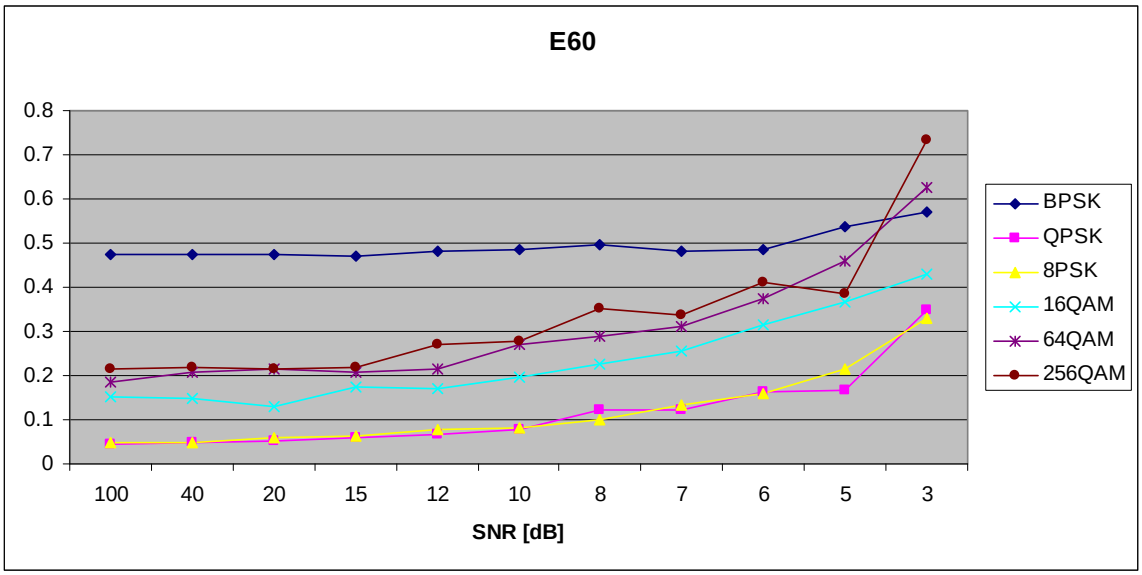

Figure 178. $E_{x,6,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

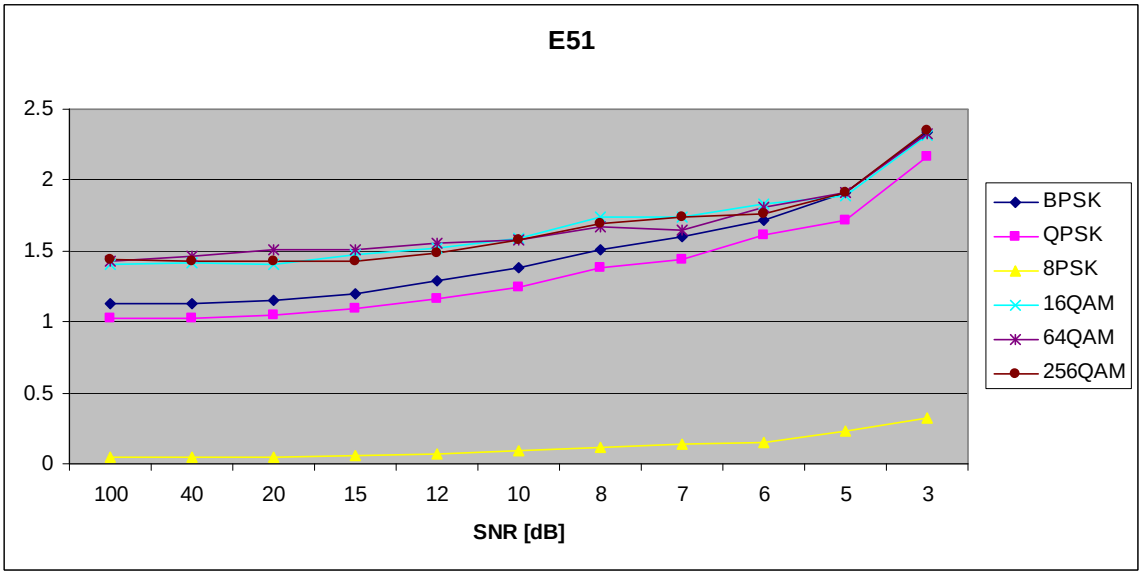

Figure 179. $E_{x,5,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

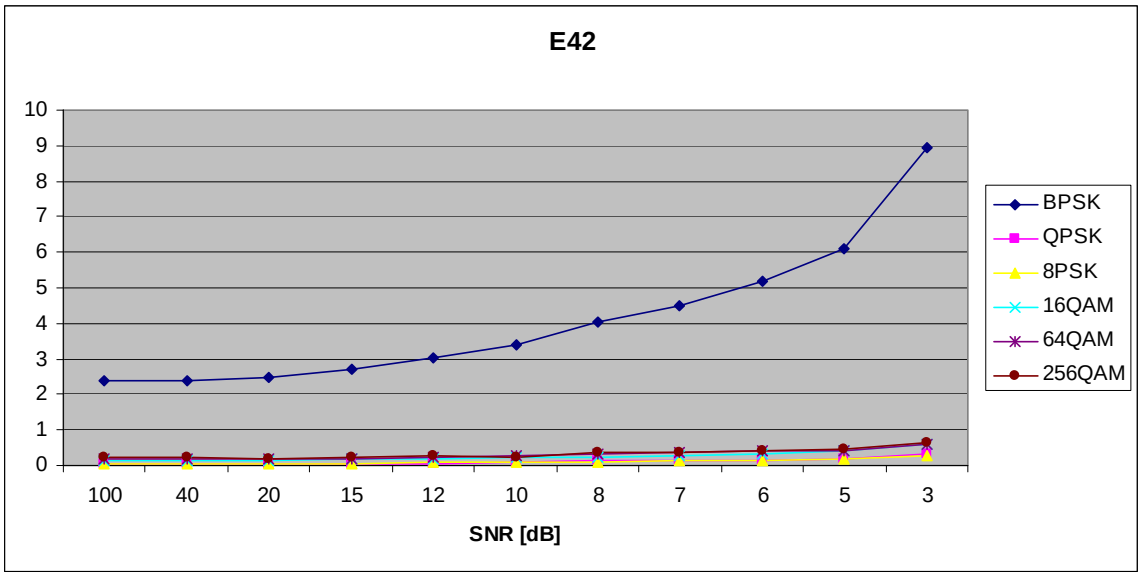

Figure 180. $E_{x,4,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

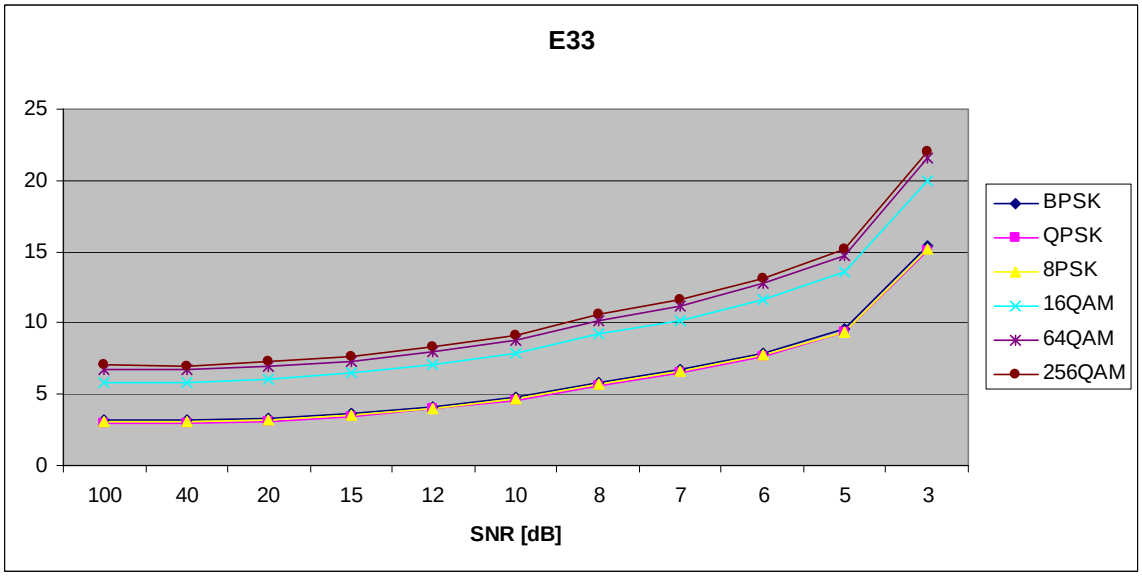

Figure 181. $E_{x,3,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

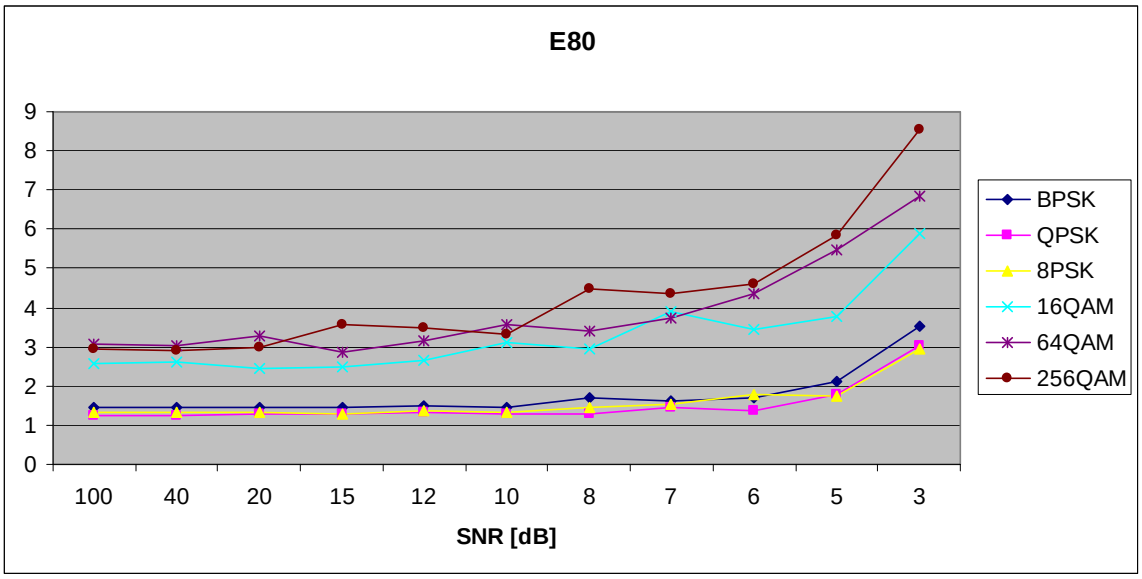

Figure 182. $E_{x,8,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

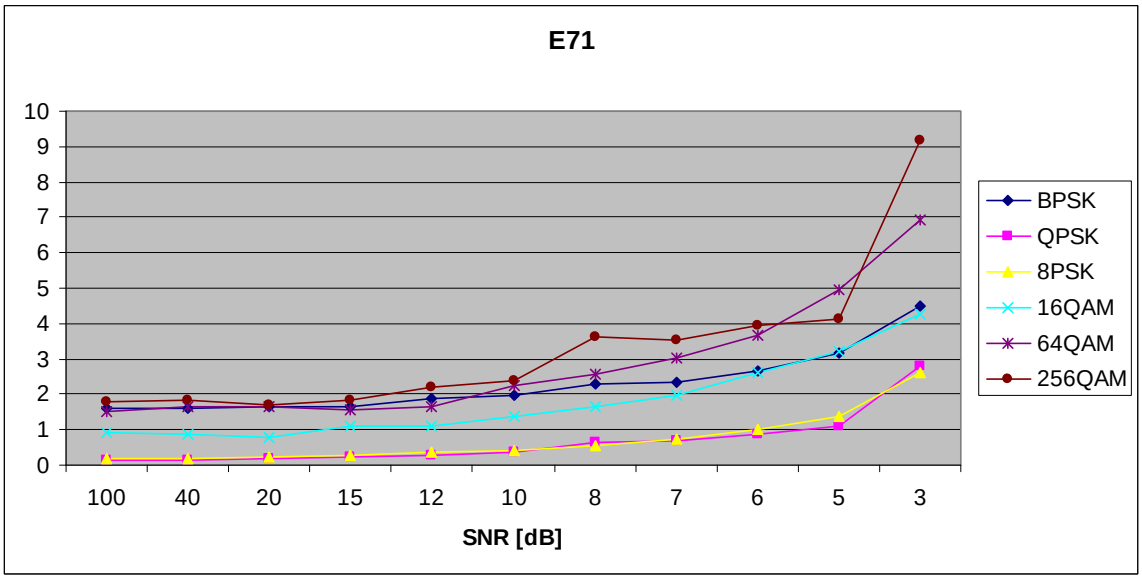

Figure 183. $E_{x,7,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

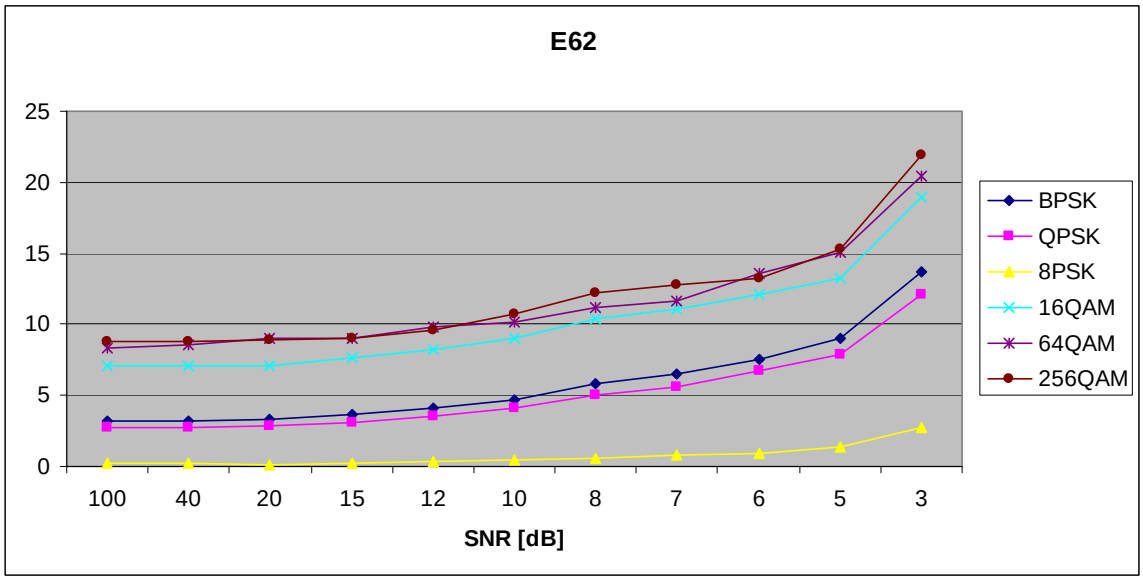

Figure 184. $E_{x,6,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

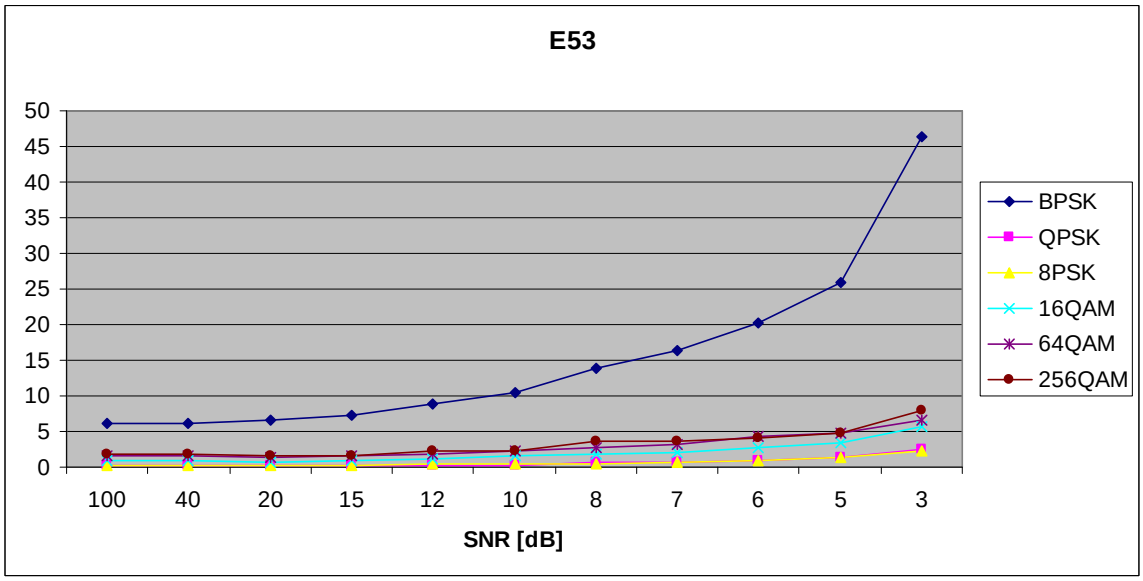

Figure 185. $E_{x,5,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

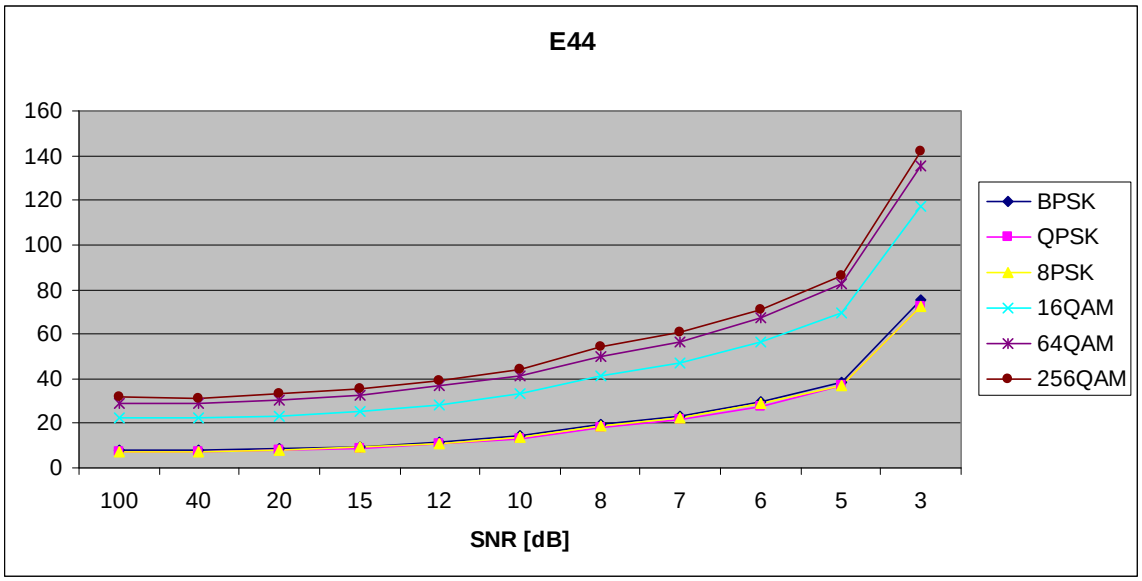

Figure 186. $E_{x,4,4}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

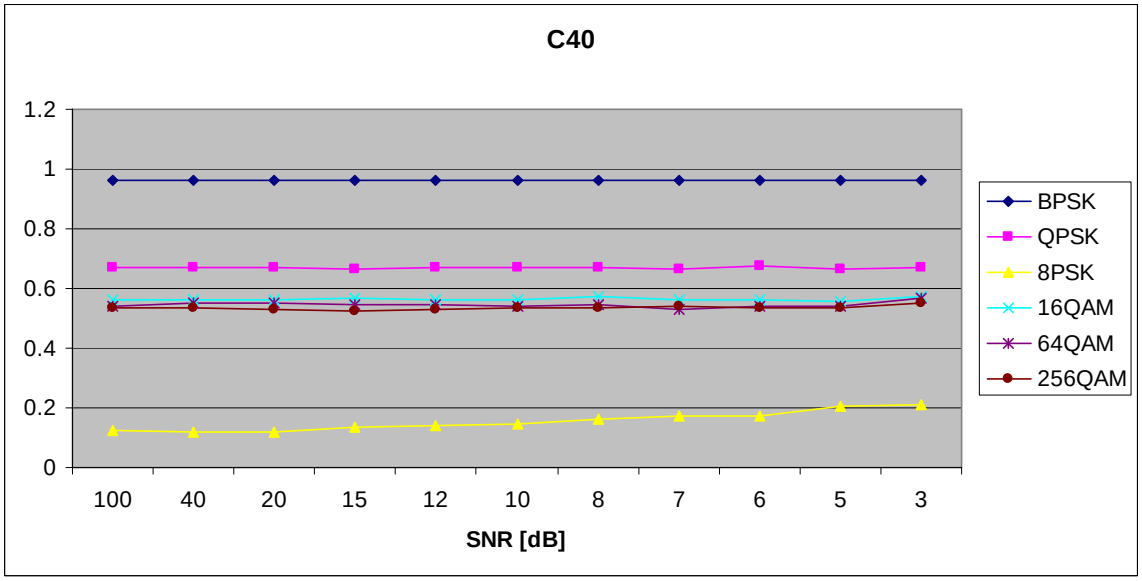

Figure 187. $C_{x,4,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

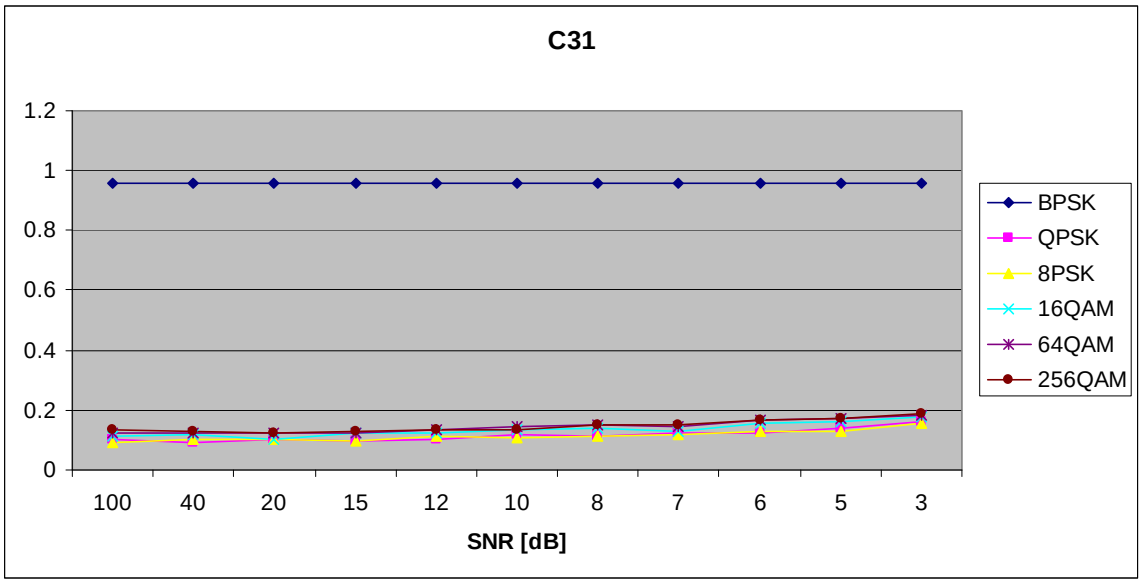

Figure 188. $C_{x,3,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

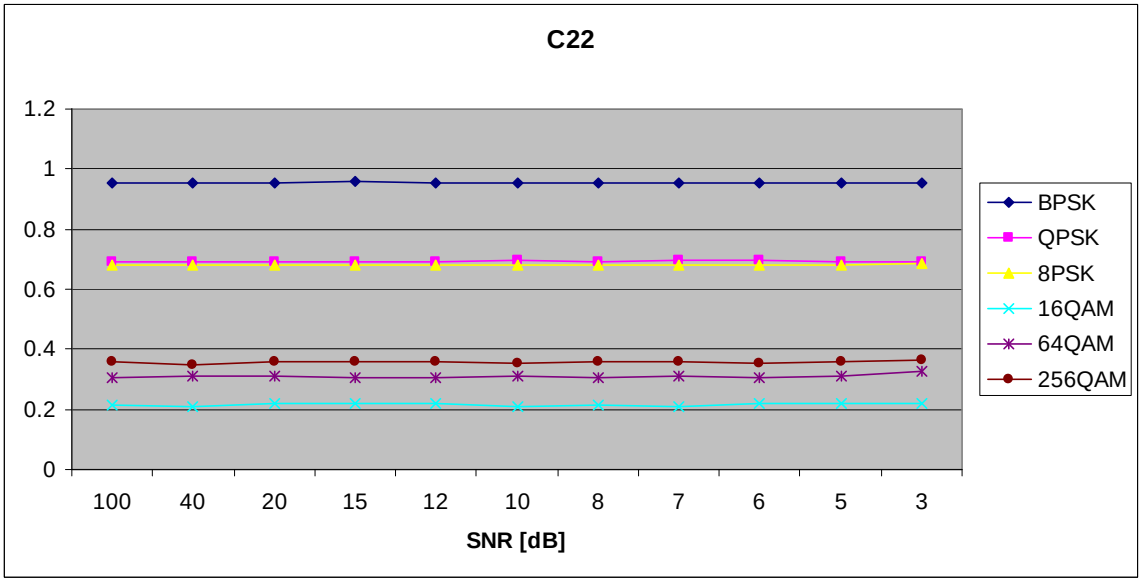

Figure 189. $C_{x,2,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

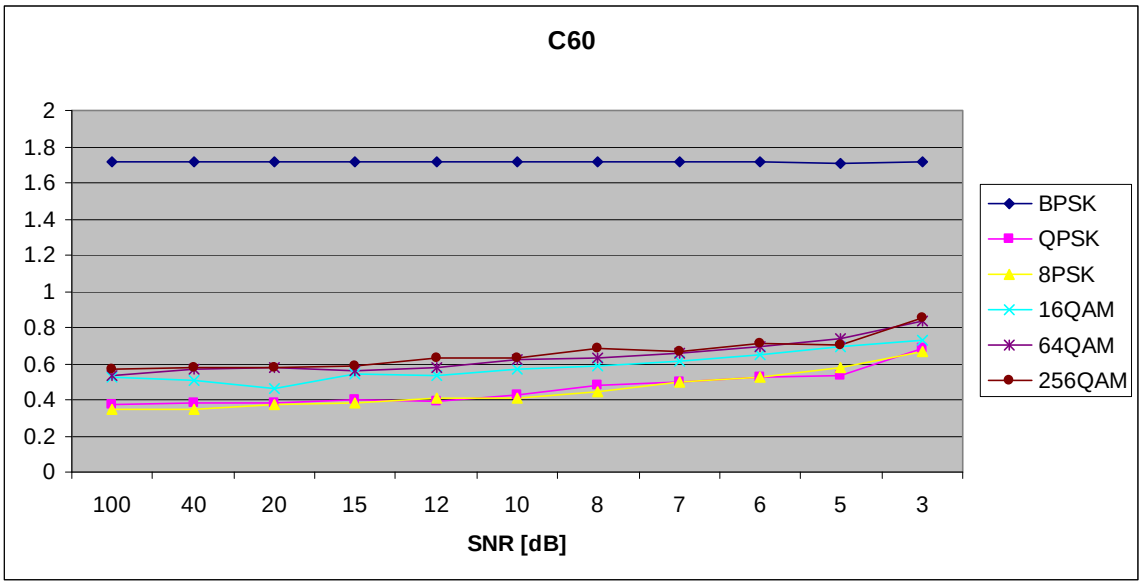

Figure 190. $C_{x,6,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

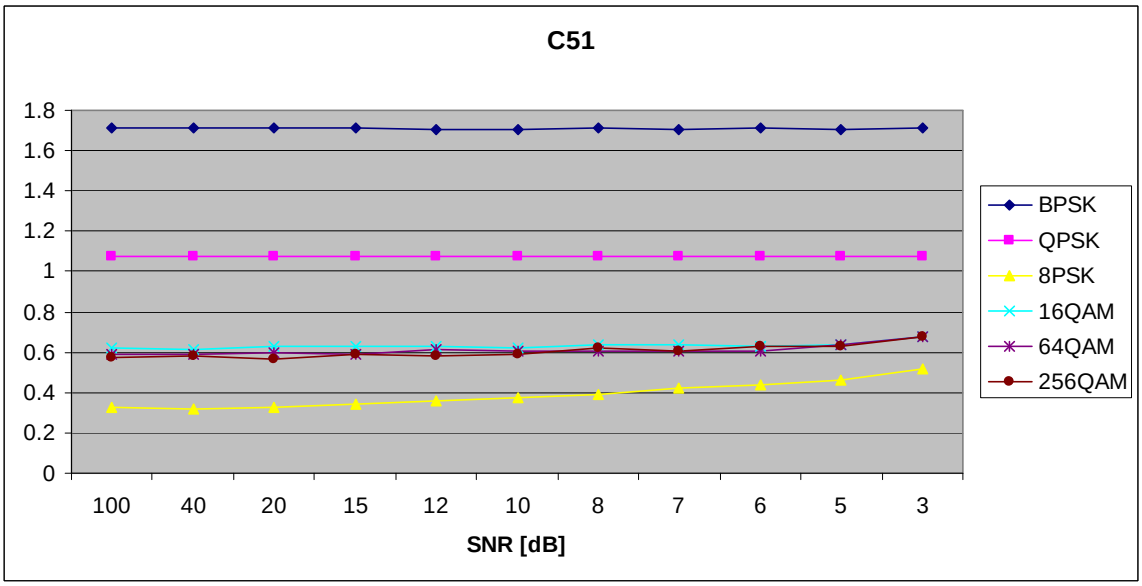

Figure 191. $C_{x,5,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

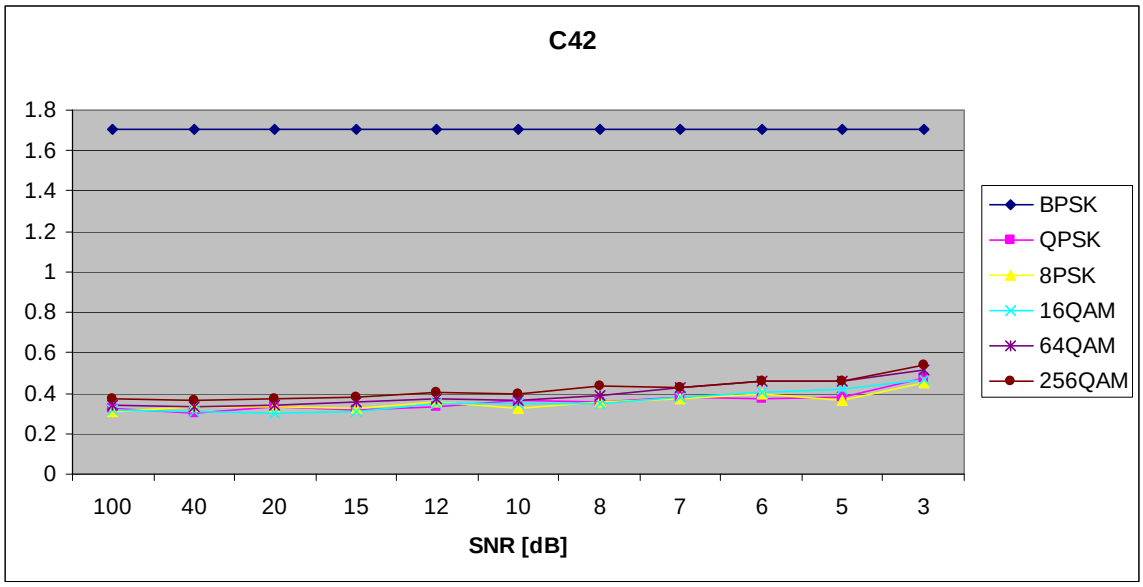

Figure 192. $C_{x,4,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

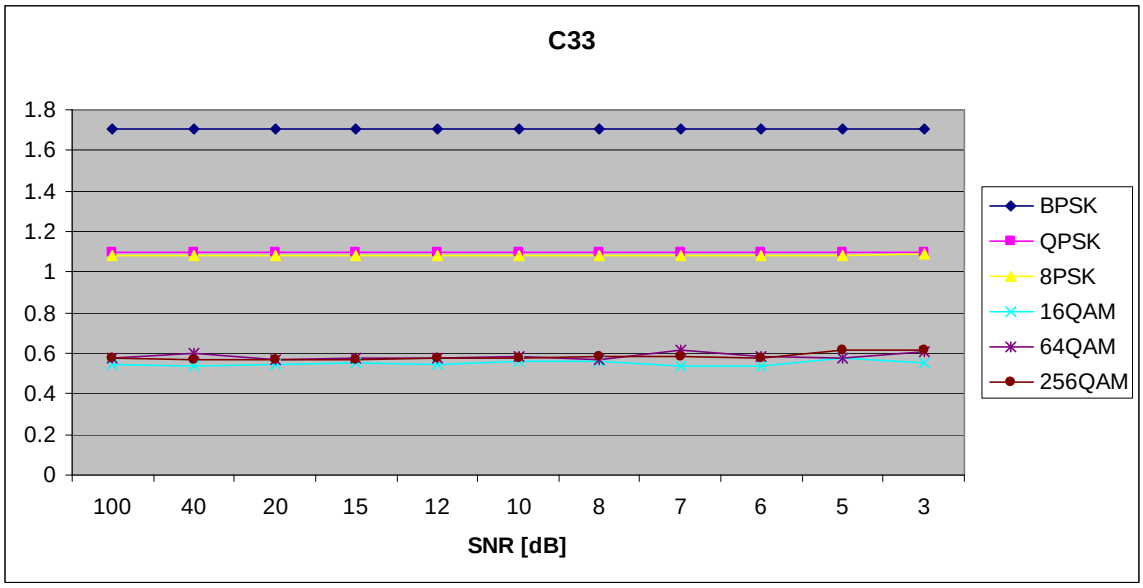

Figure 193. $C_{x,3,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

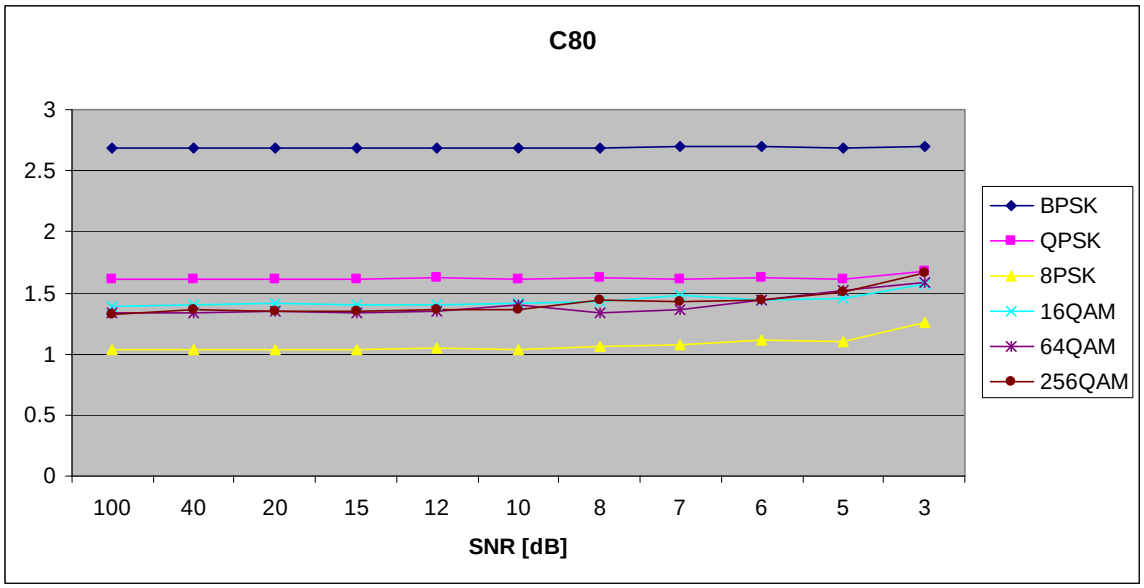

Figure 194. $C_{x,8,0}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

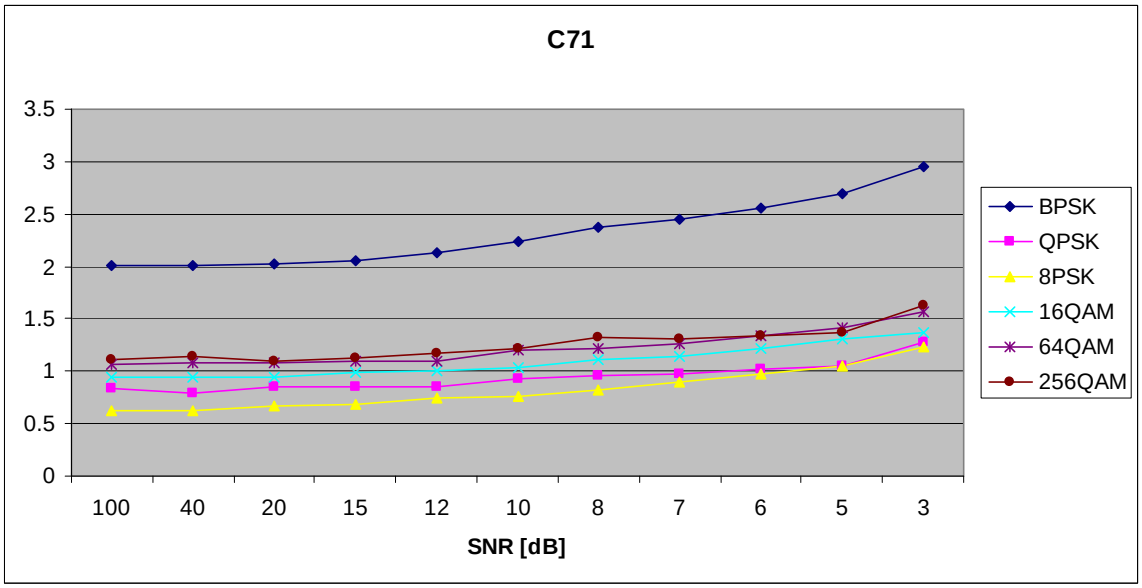

Figure 195. $C_{x,7,1}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

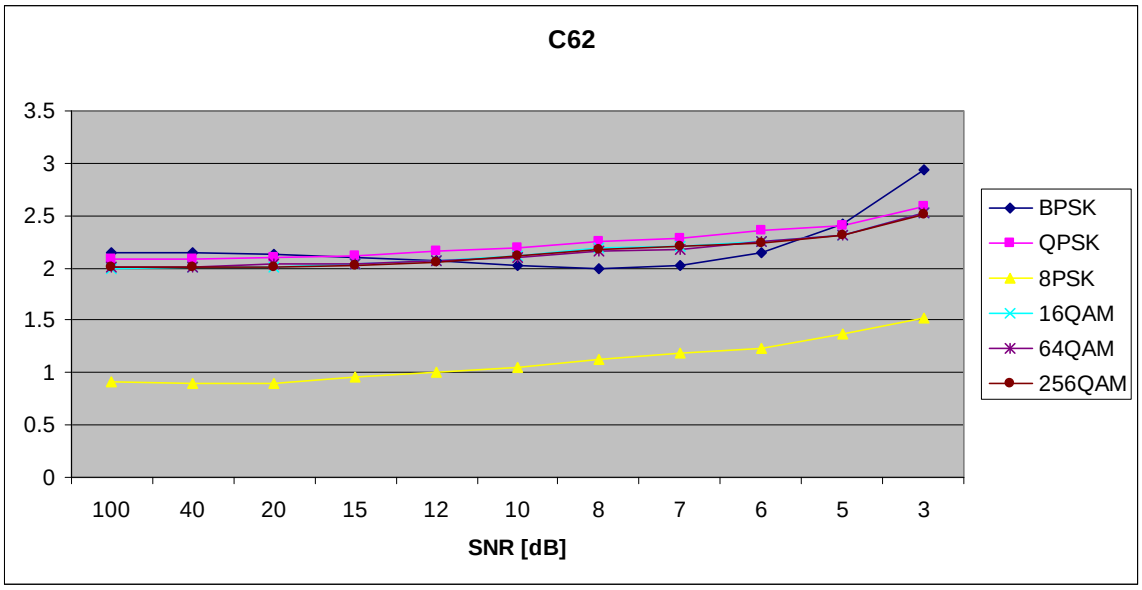

Figure 196. $C_{x,6,2}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

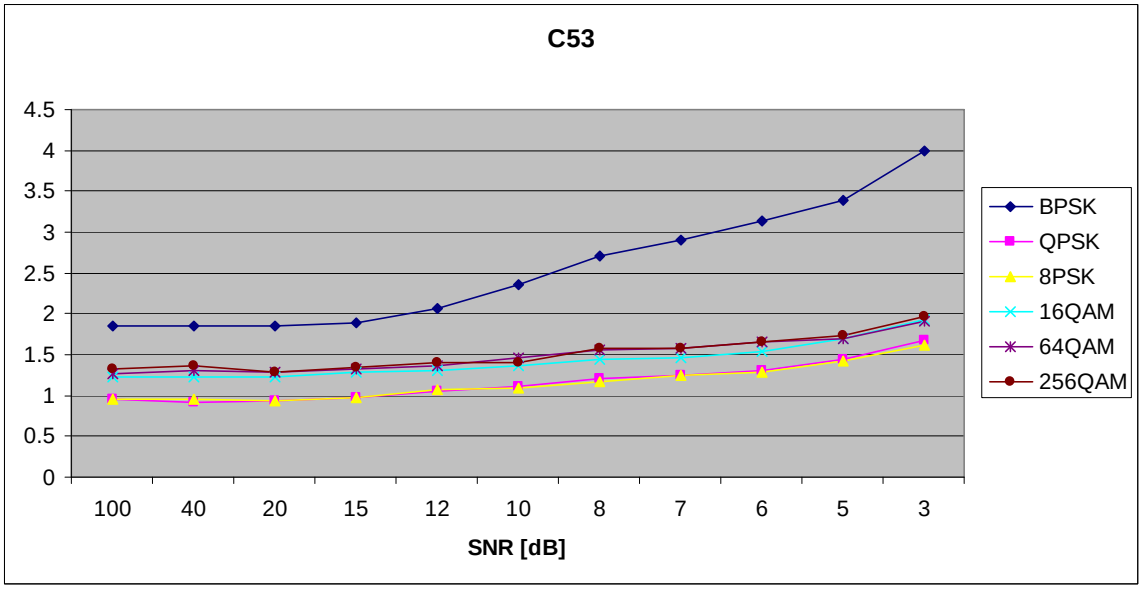

Figure 197. $C_{x,5,3}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

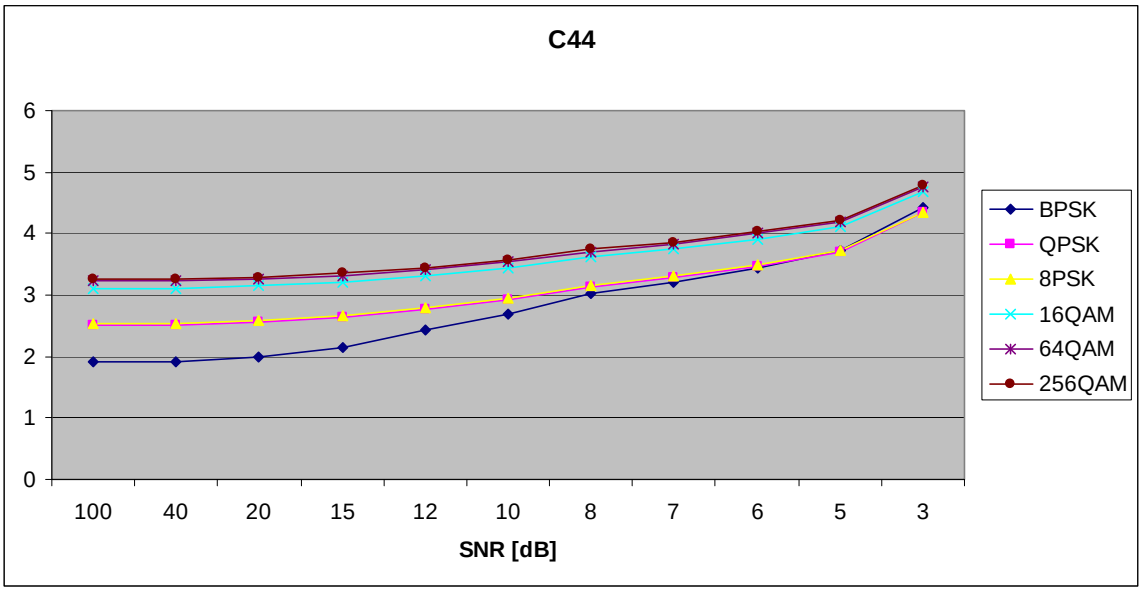

Figure 198. $C_{x,4,4}$ in AWGN and Fast, Frequency-Flat Ricean Fading.

H. AWGN PLUS FAST, FREQUENCY-SELECTIVE RAYLEIGH FADING

Parameters for the rayleighchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}
- Maximum Doppler shift: 5000 Hz
- Path Delays: [0, 2×10^{-6}]
- Average Path Gains: [0, -10]

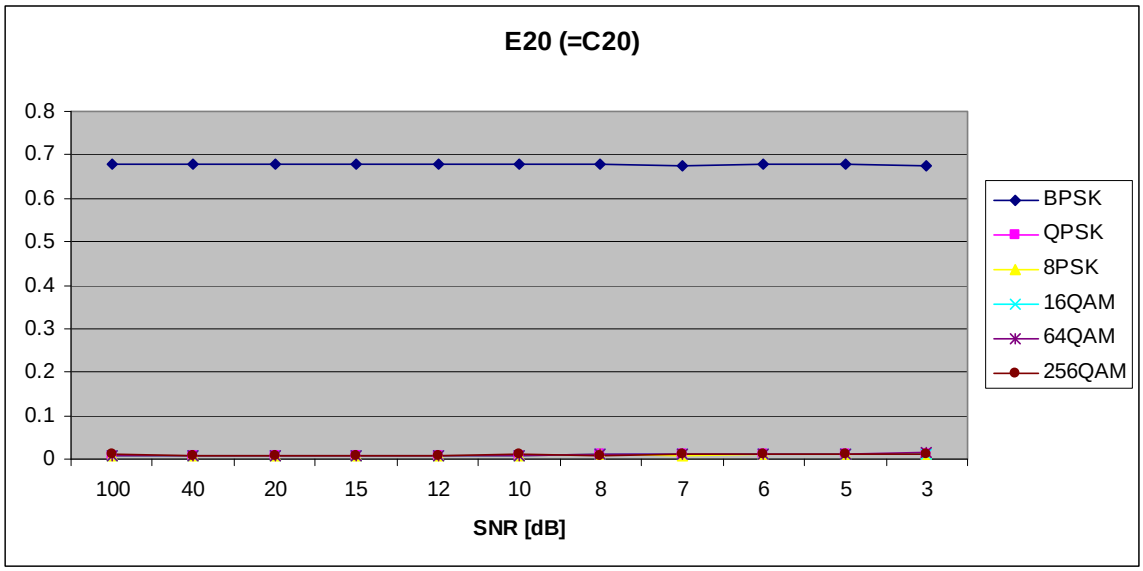

Figure 199. $E_{x,2,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

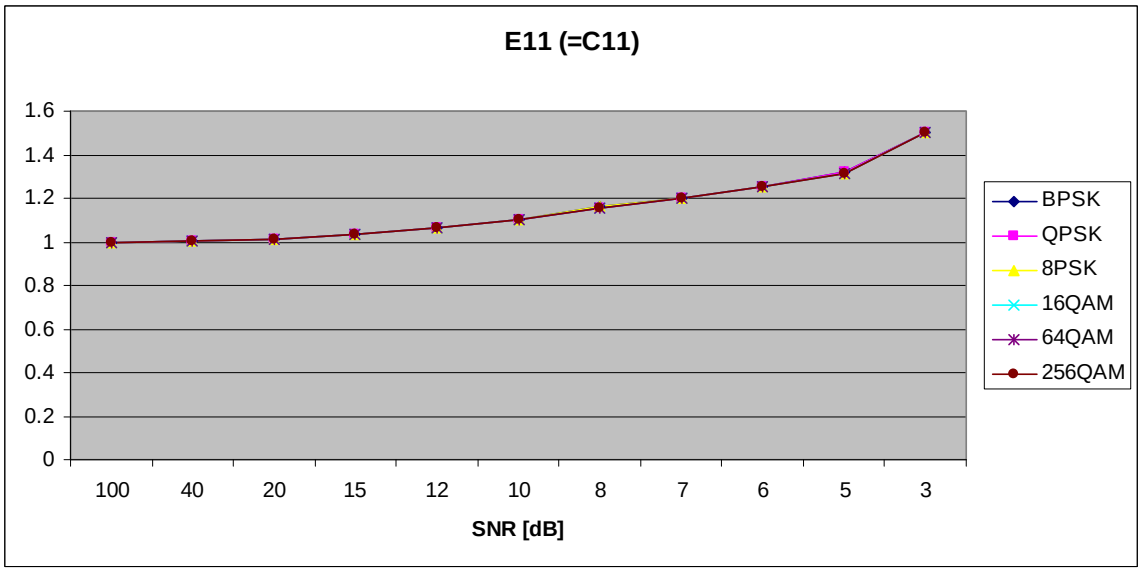

Figure 200. $E_{x,1,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

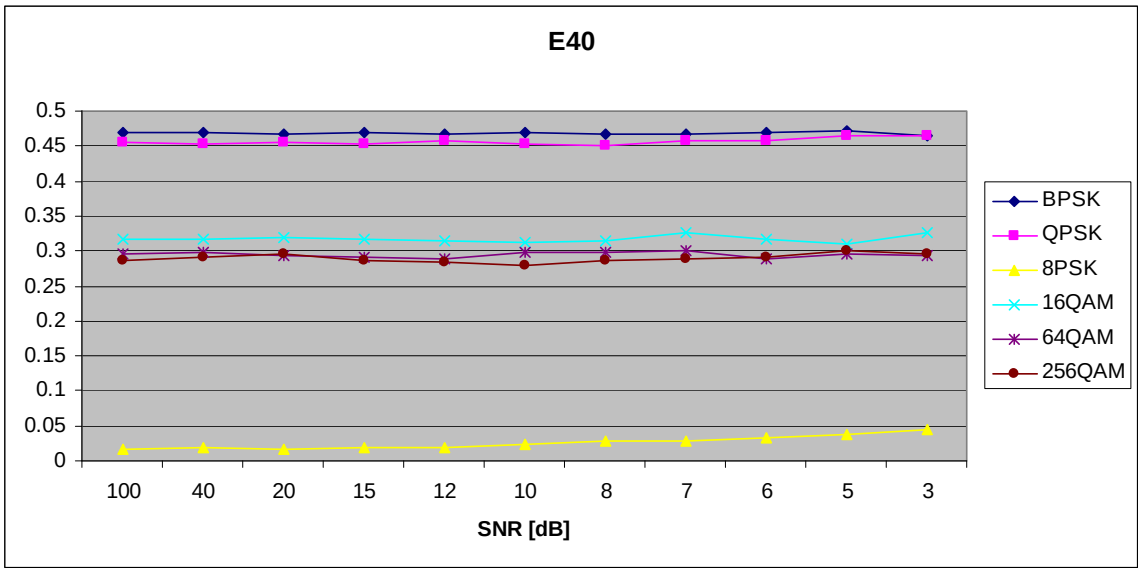

Figure 201. $E_{x,4,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

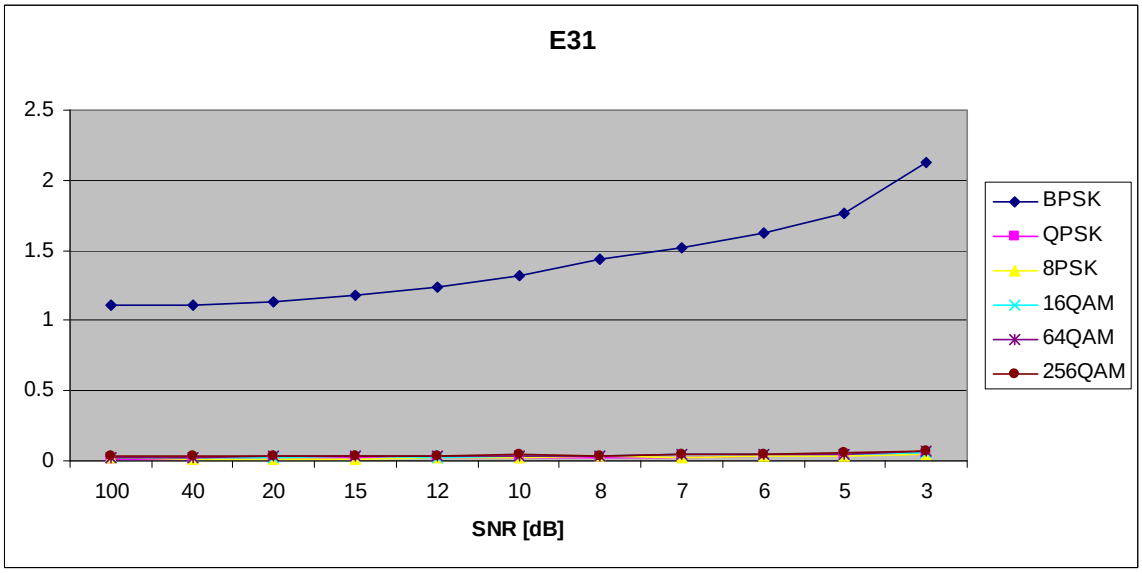

Figure 202. $E_{x,3,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

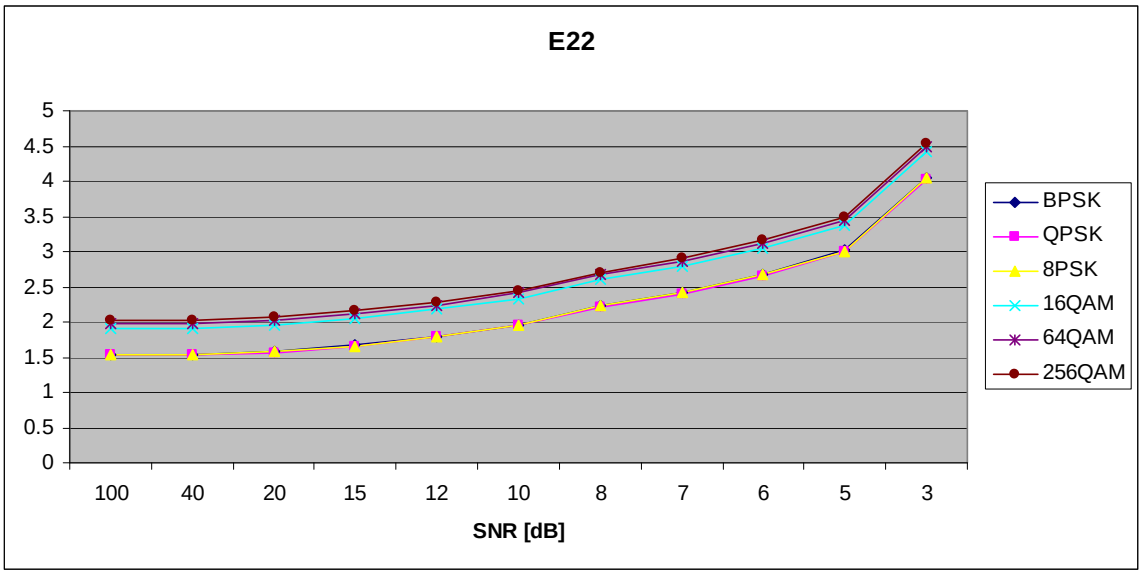

Figure 203. $E_{x,2,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

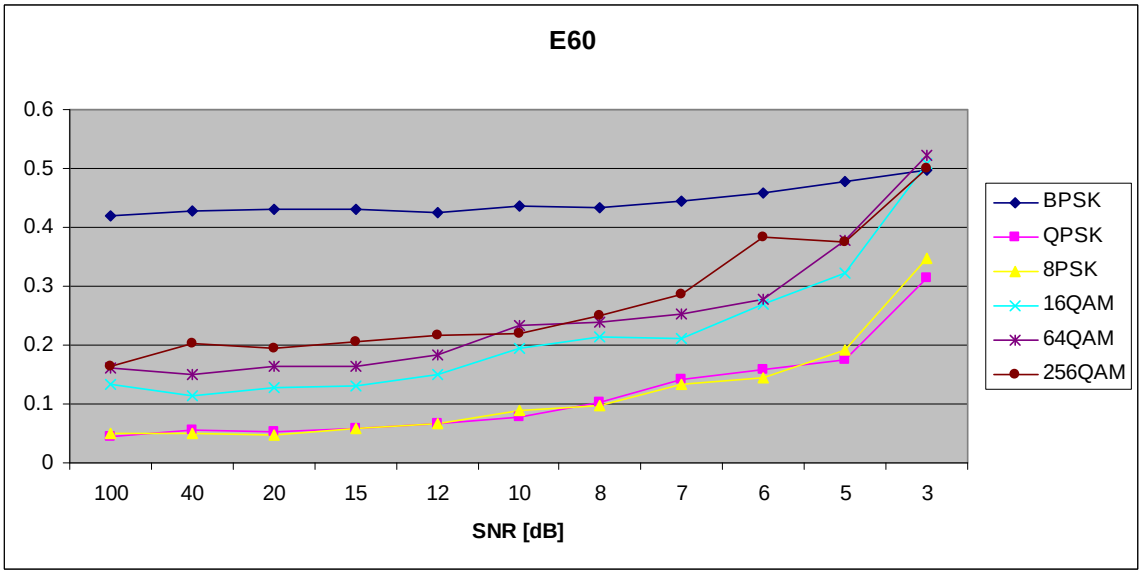

Figure 204. $E_{x,6,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

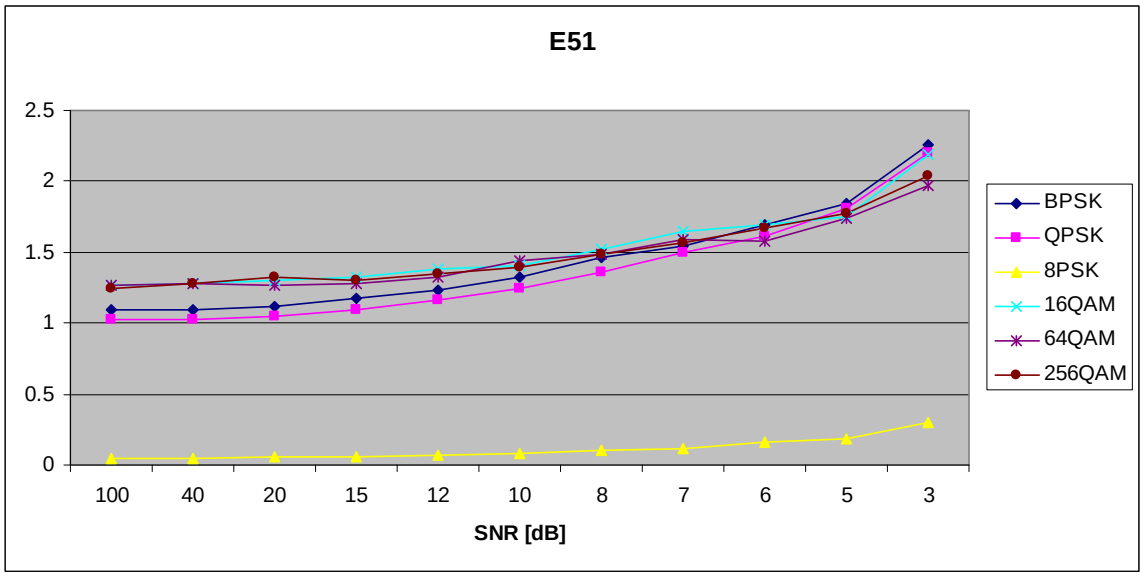

Figure 205. $E_{x,5,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

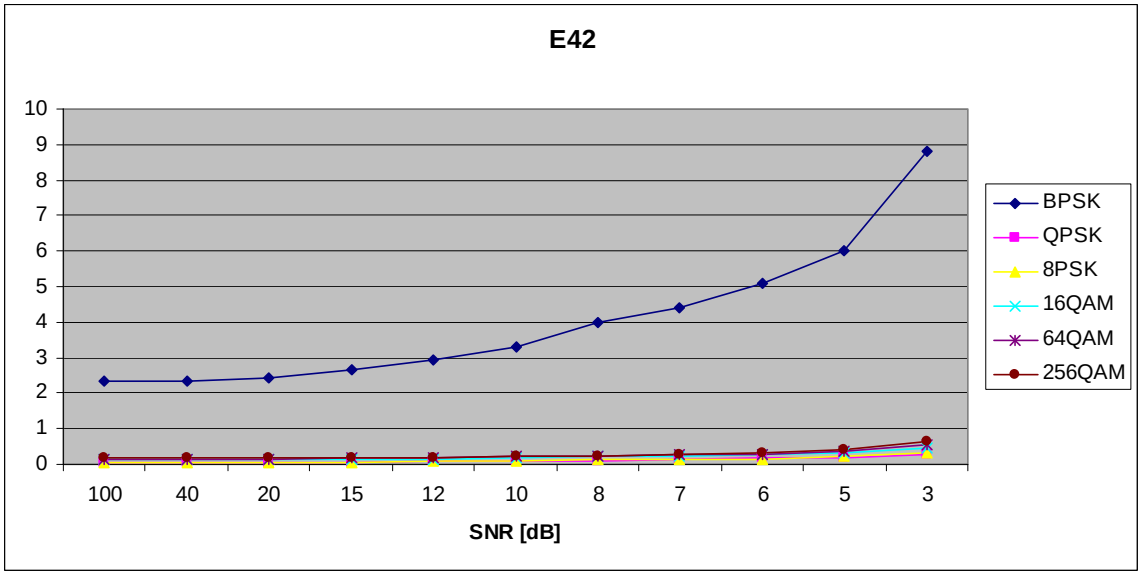

Figure 206. $E_{x,4,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

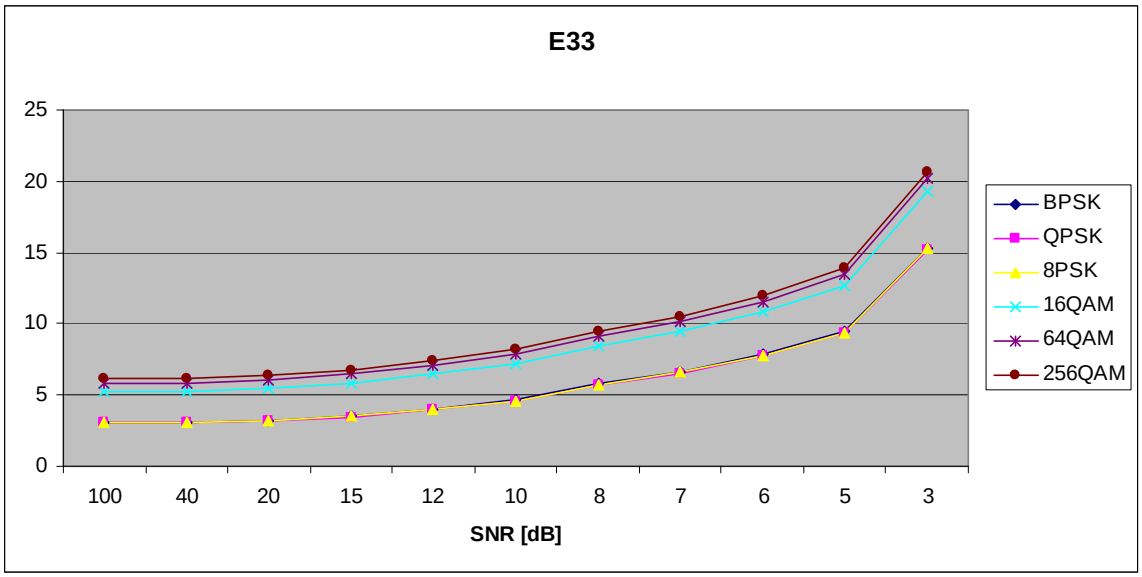

Figure 207. $E_{x,3,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

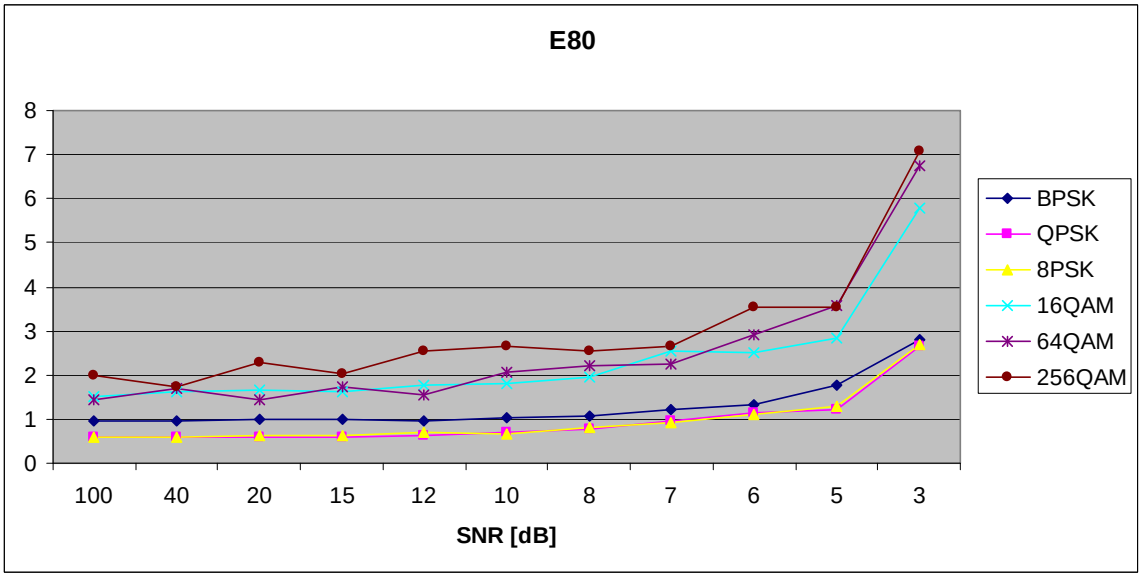

Figure 208. $E_{x,8,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

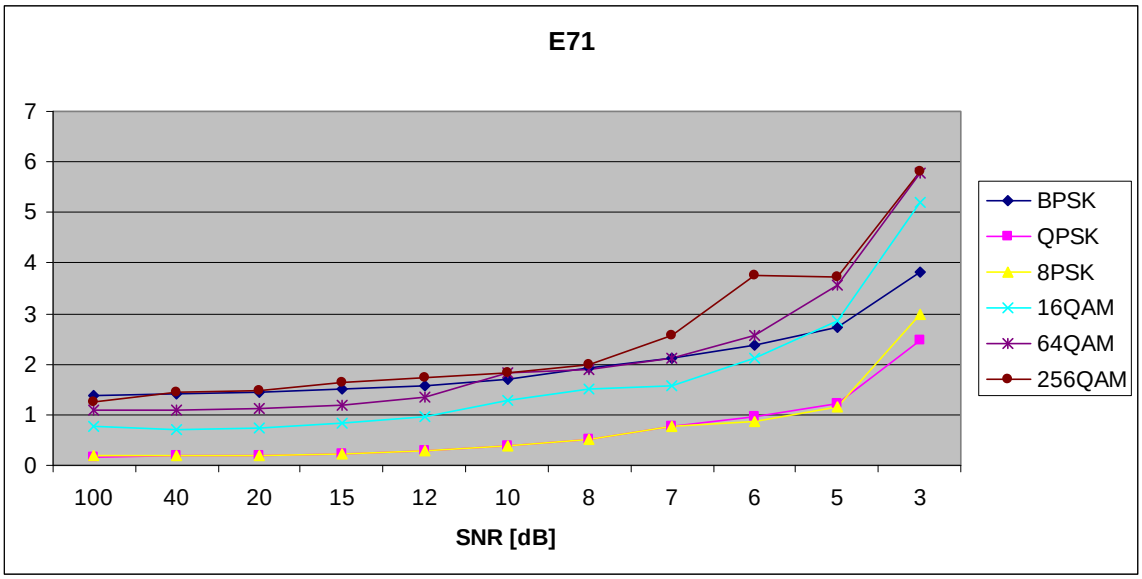

Figure 209. $E_{x,7,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

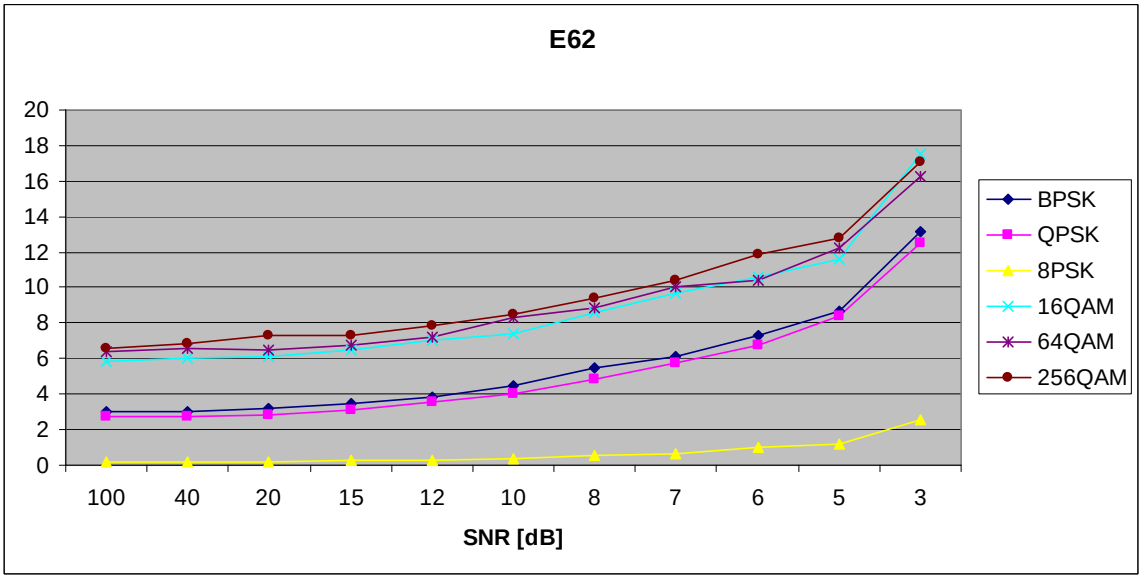

Figure 210. $E_{x,6,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

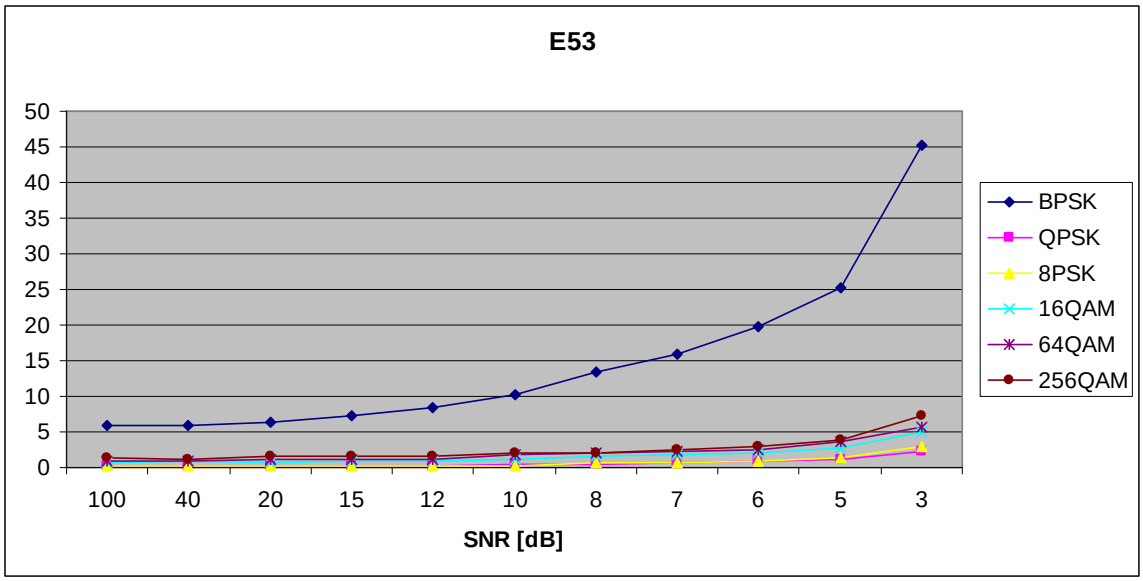

Figure 211. $E_{x,5,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

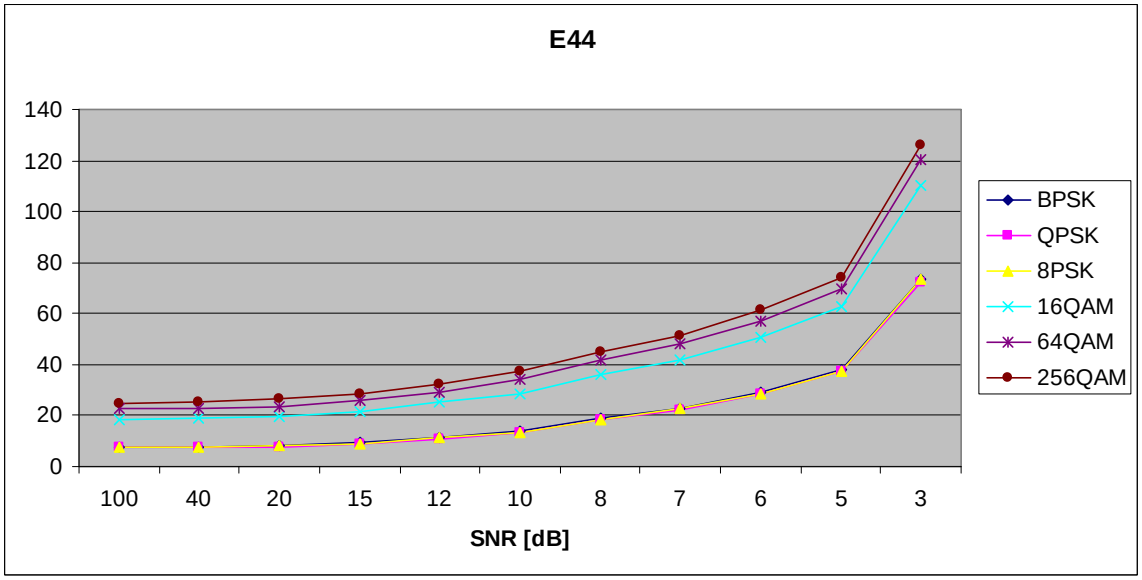

Figure 212. $E_{x,4,4}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

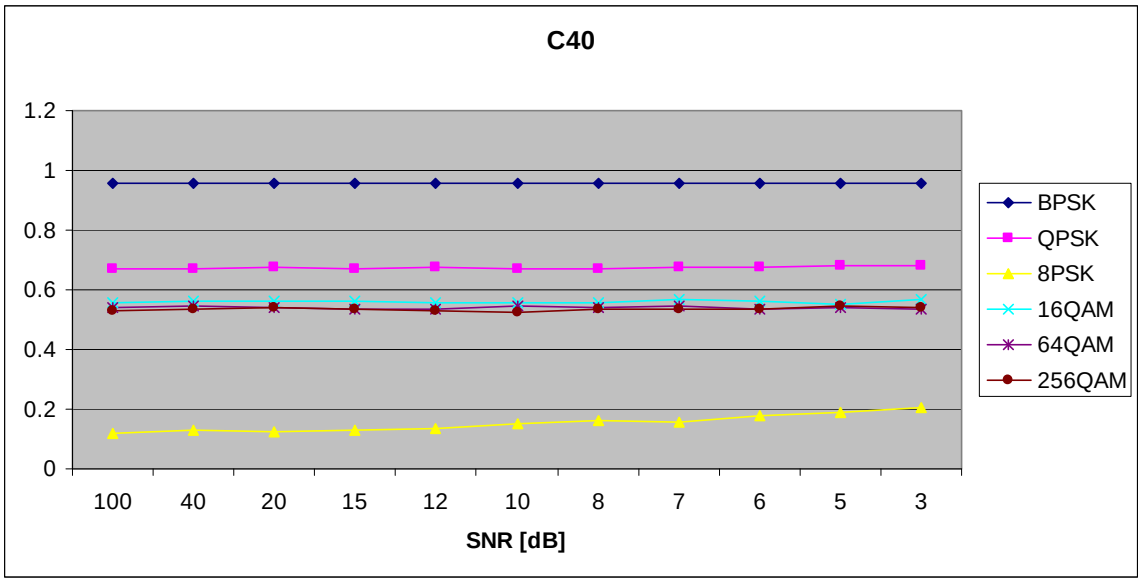

Figure 213. $C_{x,4,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

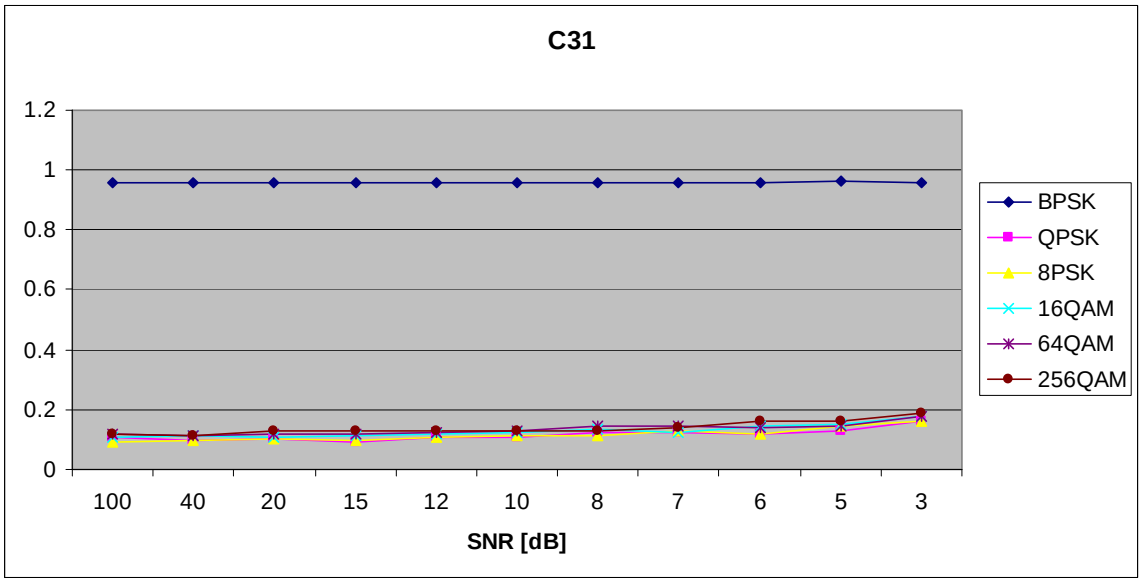

Figure 214. $C_{x,3,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

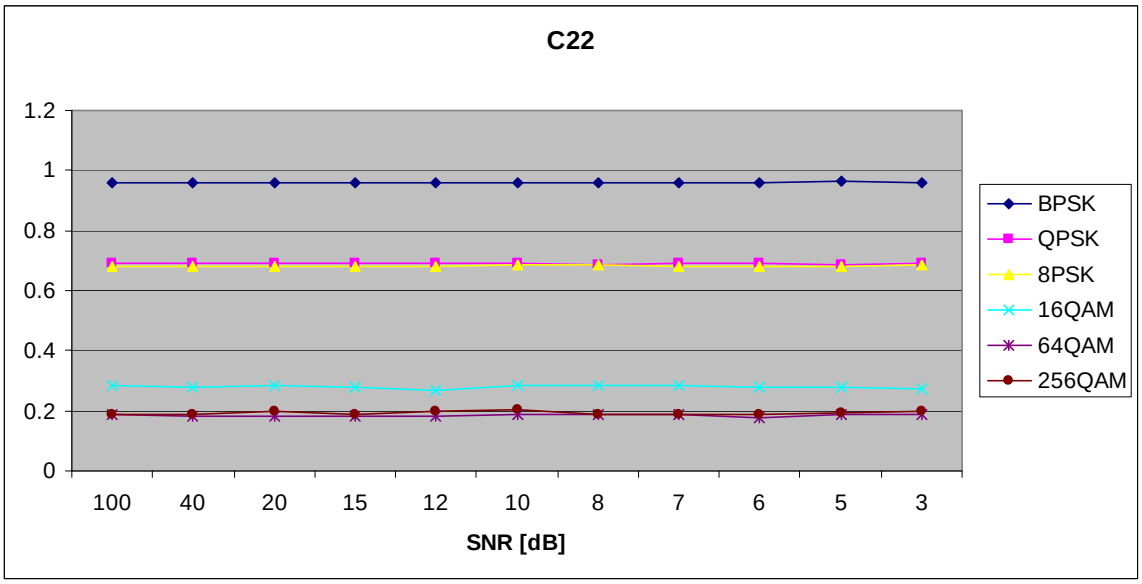

Figure 215. $C_{x,2,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

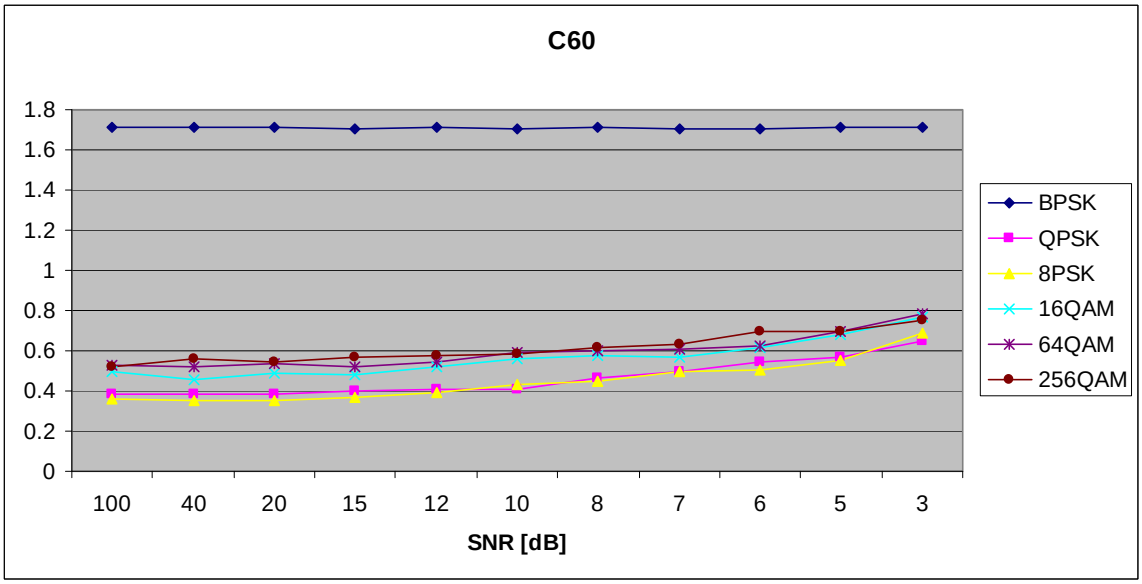

Figure 216. $C_{x,6,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

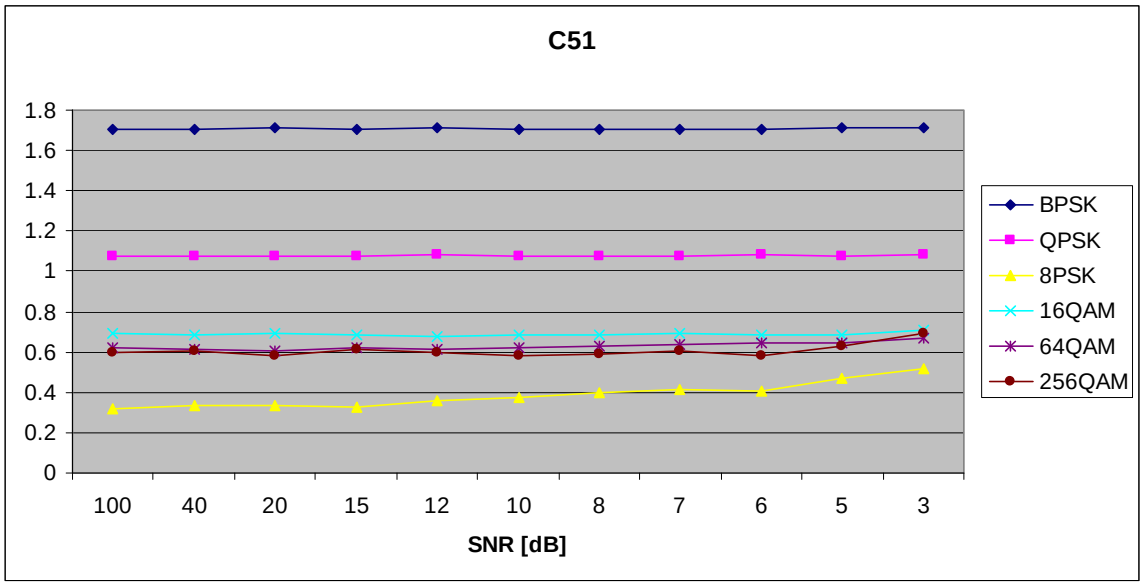

Figure 217. $C_{x,5,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

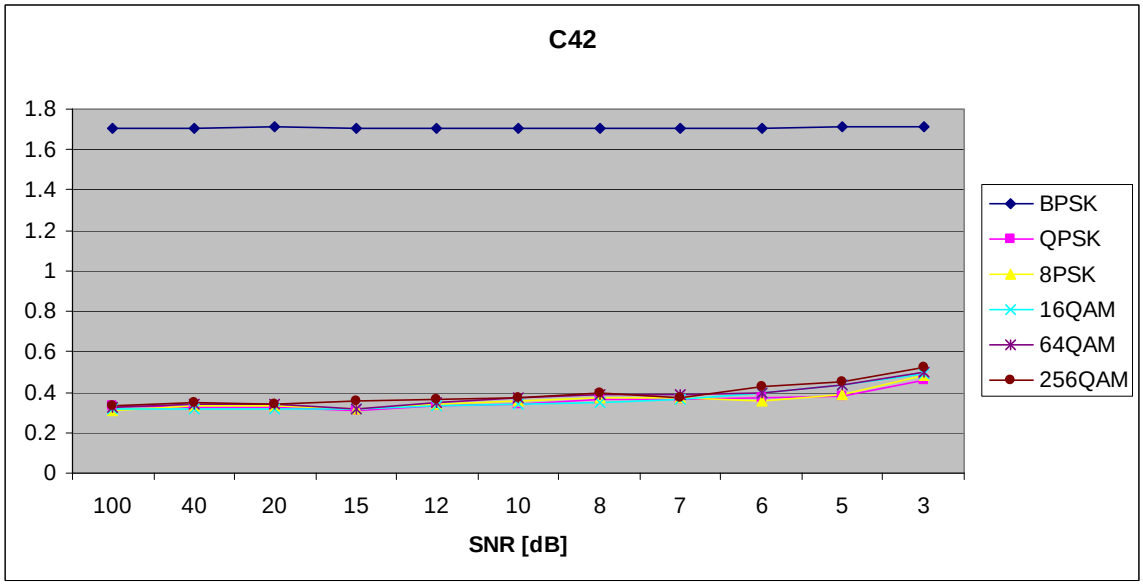

Figure 218. $C_{x,4,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

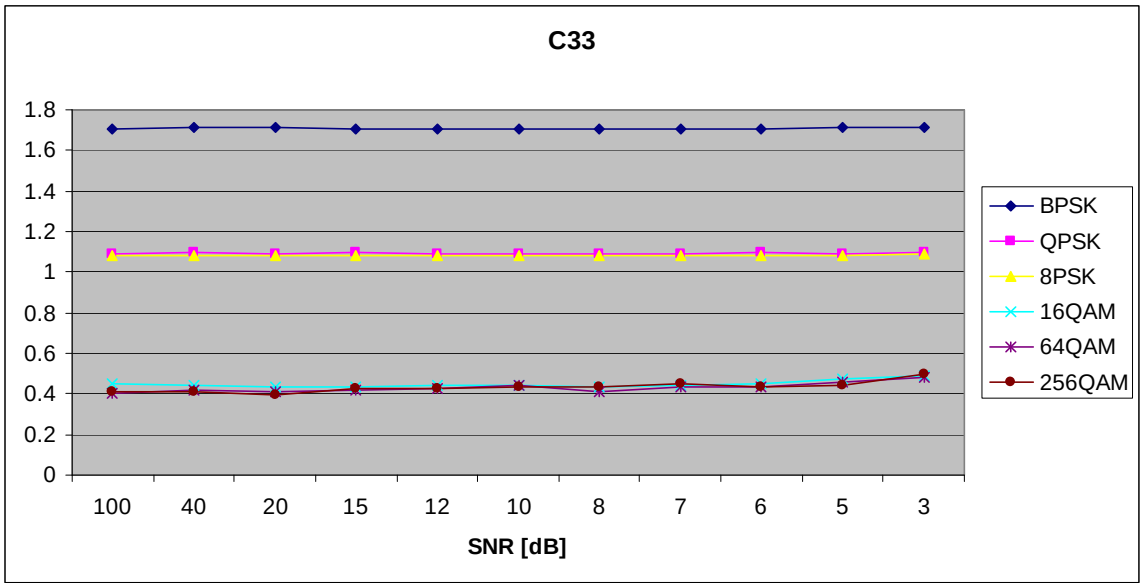

Figure 219. $C_{x,3,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

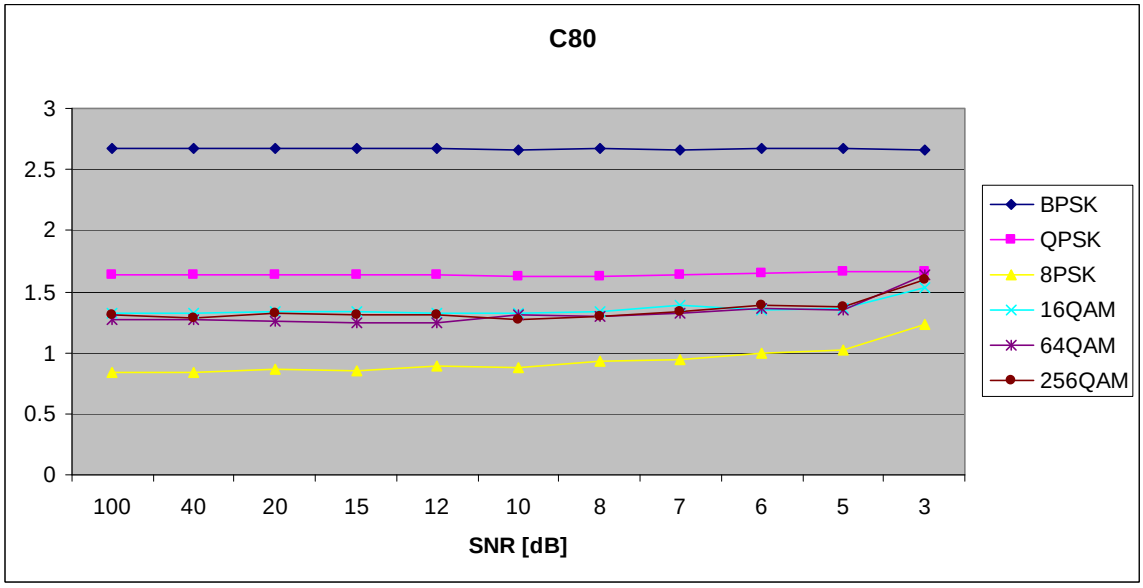

Figure 220. $C_{x,8,0}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

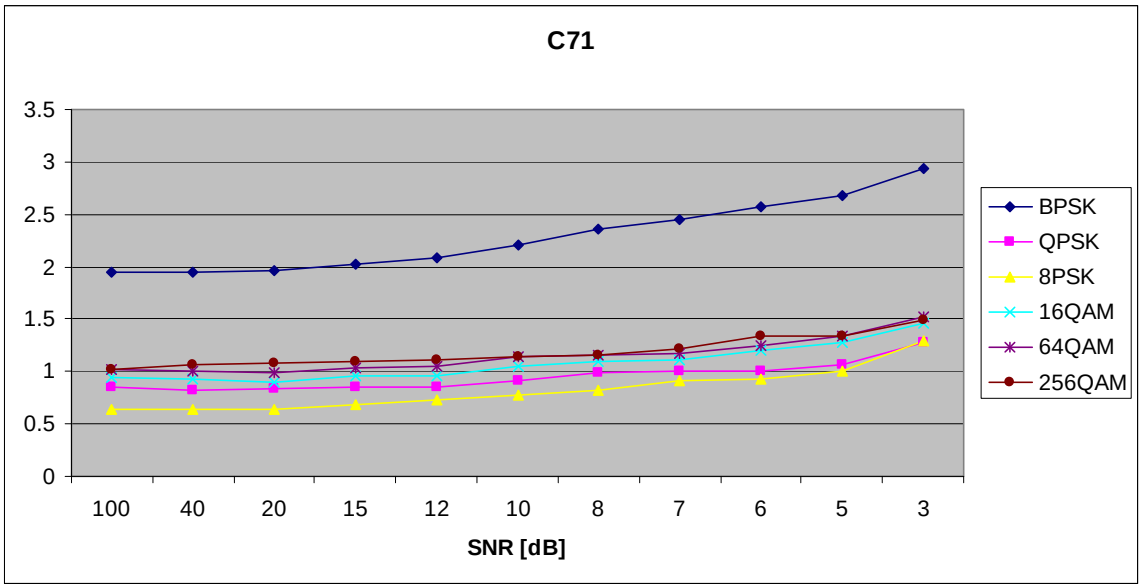

Figure 221. $C_{x,7,1}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

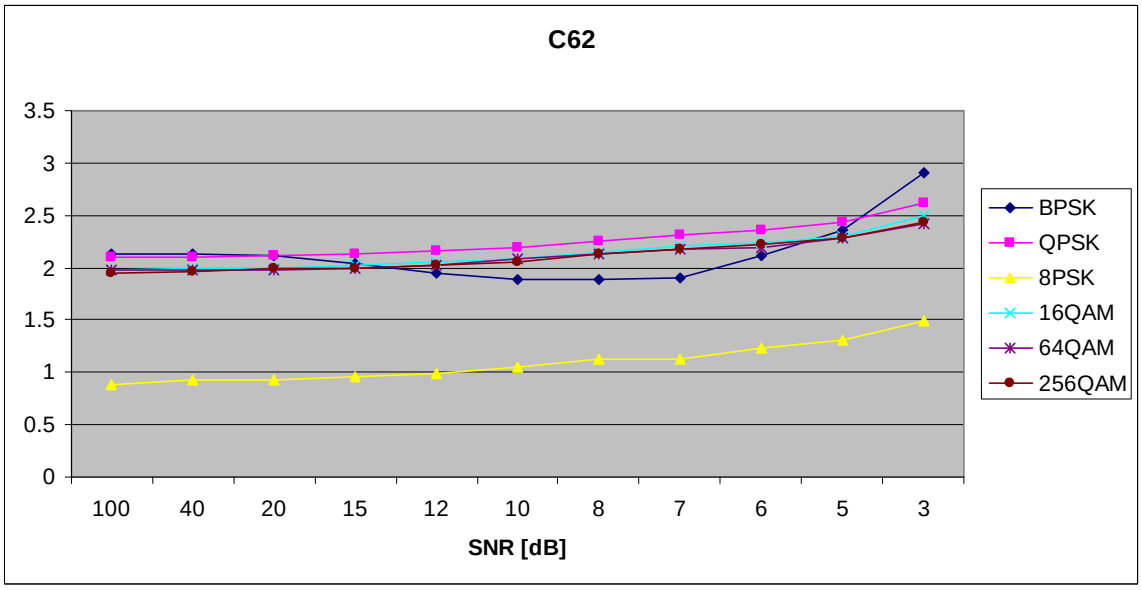

Figure 222. $C_{x,6,2}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

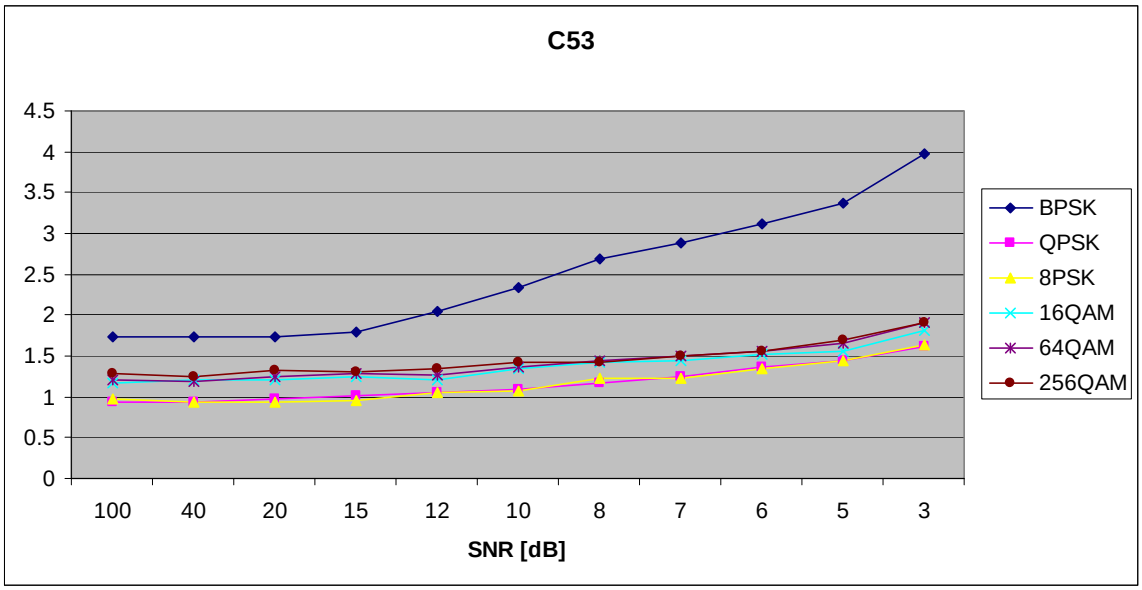

Figure 223. $C_{x,5,3}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

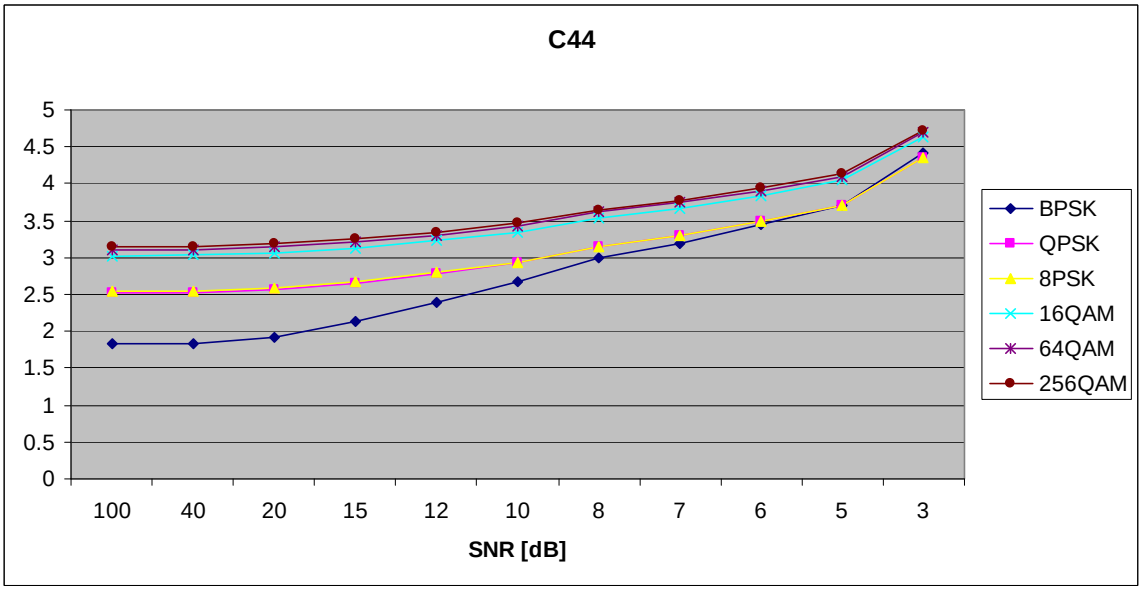

Figure 224. $C_{x,4,4}$ in AWGN and Fast, Frequency-Selective Rayleigh Fading.

I. AWGN PLUS FAST, FREQUENCY-SELECTIVE RICEAN FADING

Parameters for the ricianchan.m function in MATLAB are:

- Sampling interval: 1×10^{-6}

- Maximum Doppler shift: 5000 Hz
- Path Delays: [0, 2×10^{-6}]
- Average Path Gains: [0, -10]

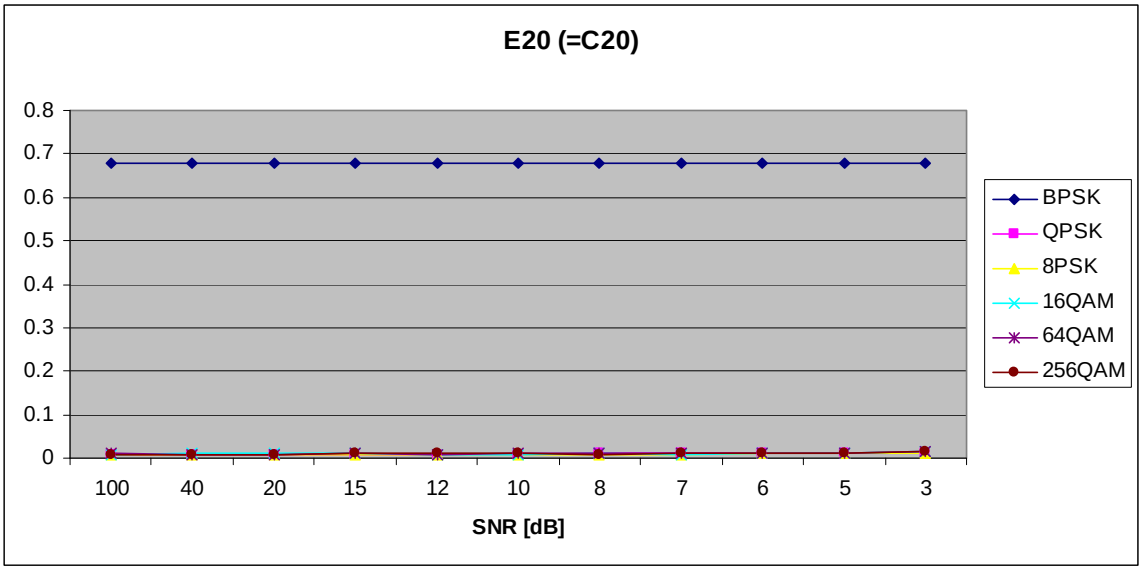

Figure 225. $E_{x,2,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

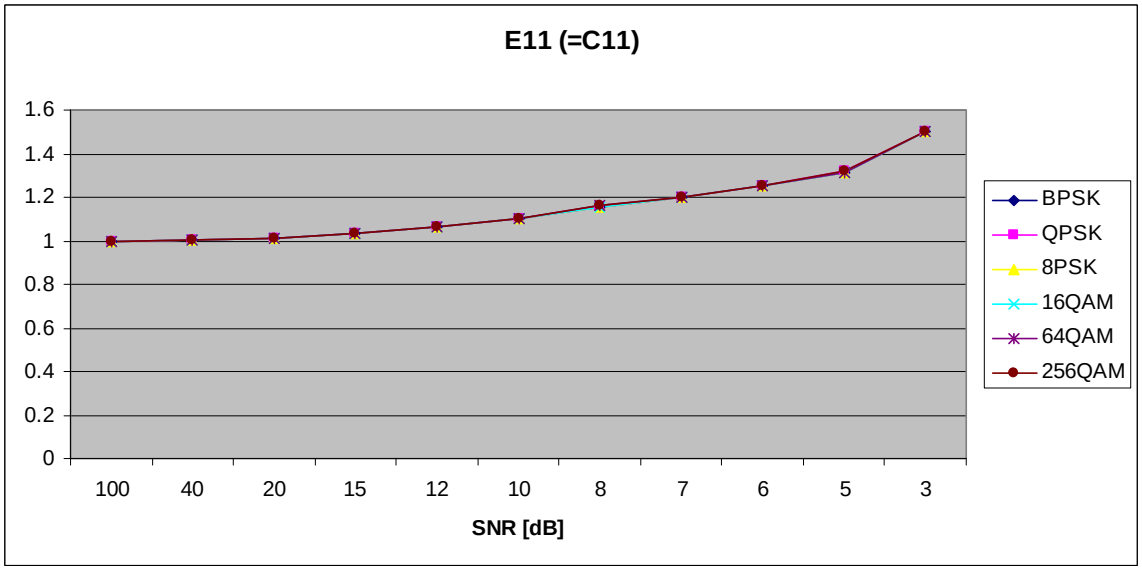

Figure 226. $E_{x,1,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

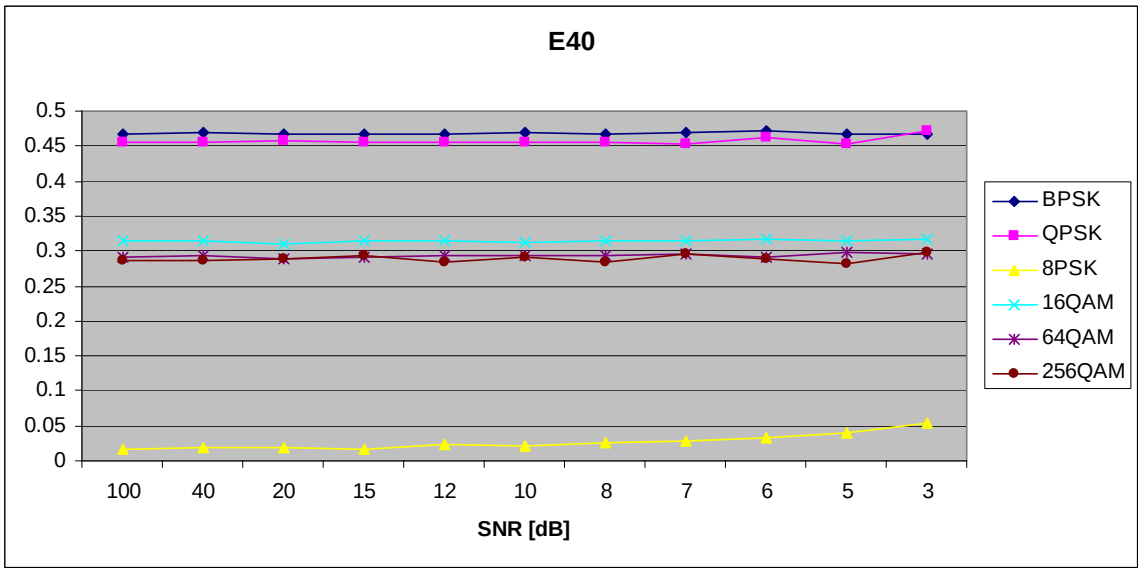

Figure 227. $E_{x,4,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

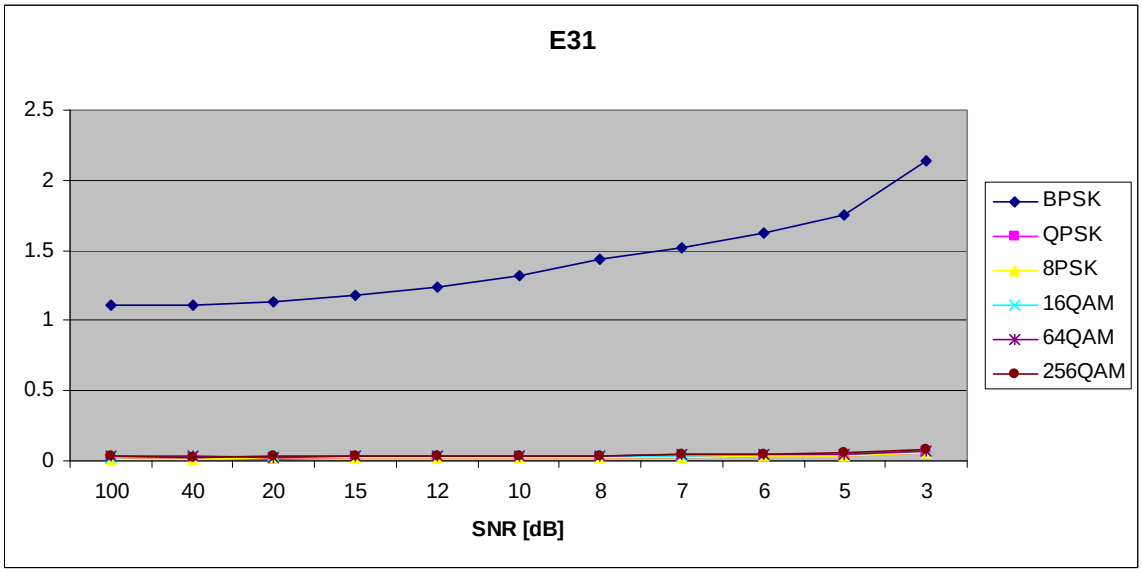

Figure 228. $E_{x,3,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

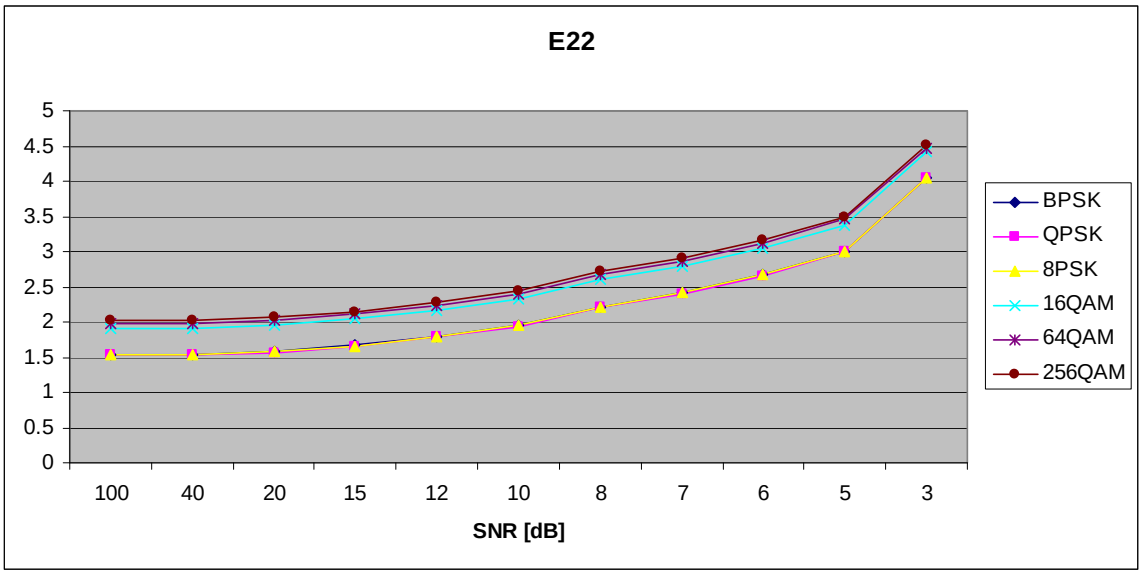

Figure 229. $E_{x,2,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

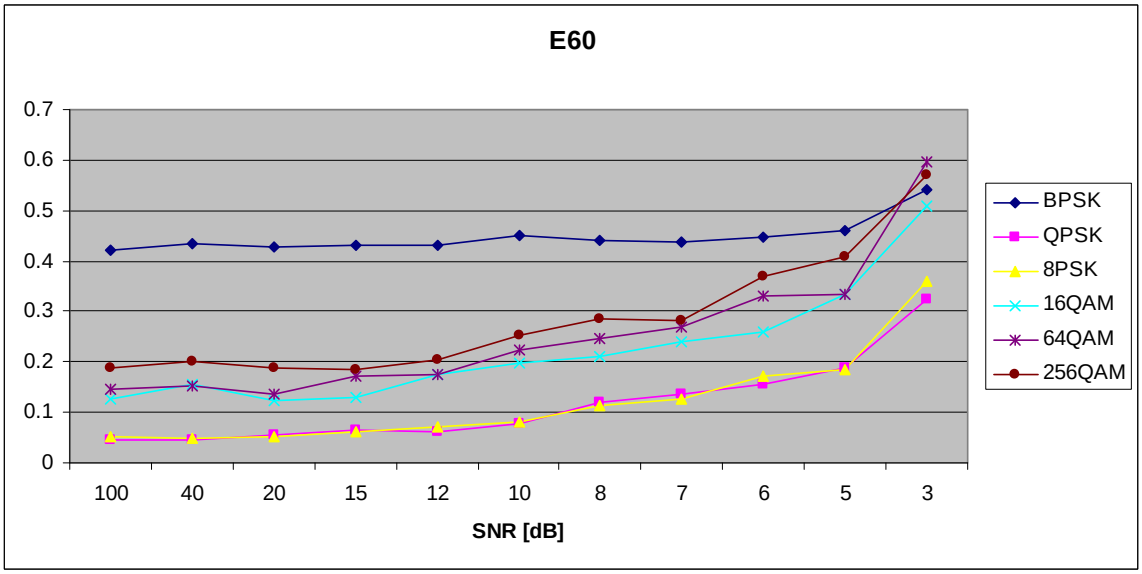

Figure 230. $E_{x,6,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

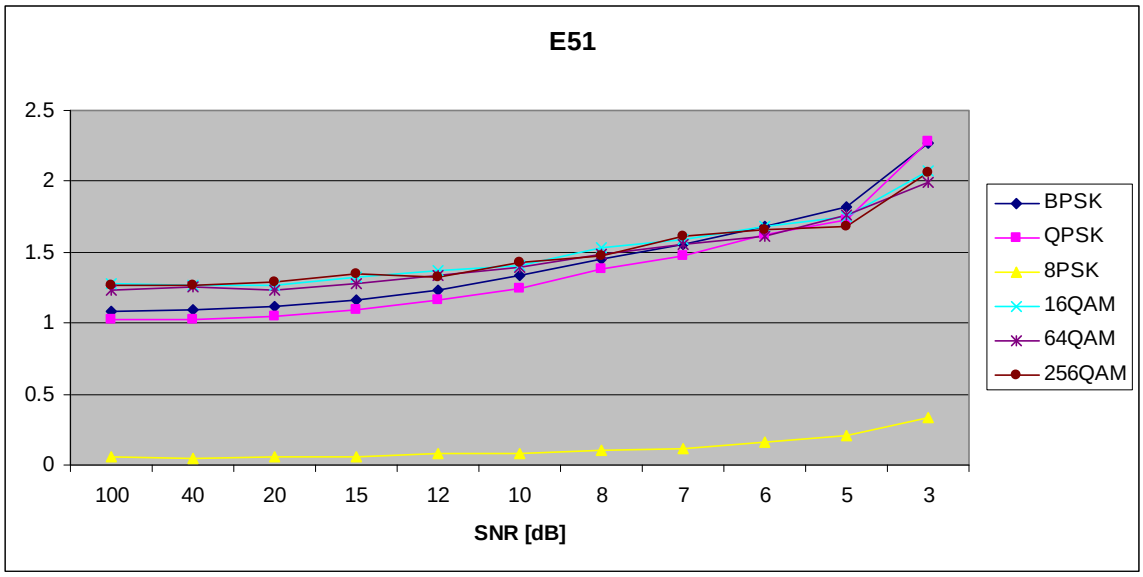

Figure 231. $E_{x,5,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

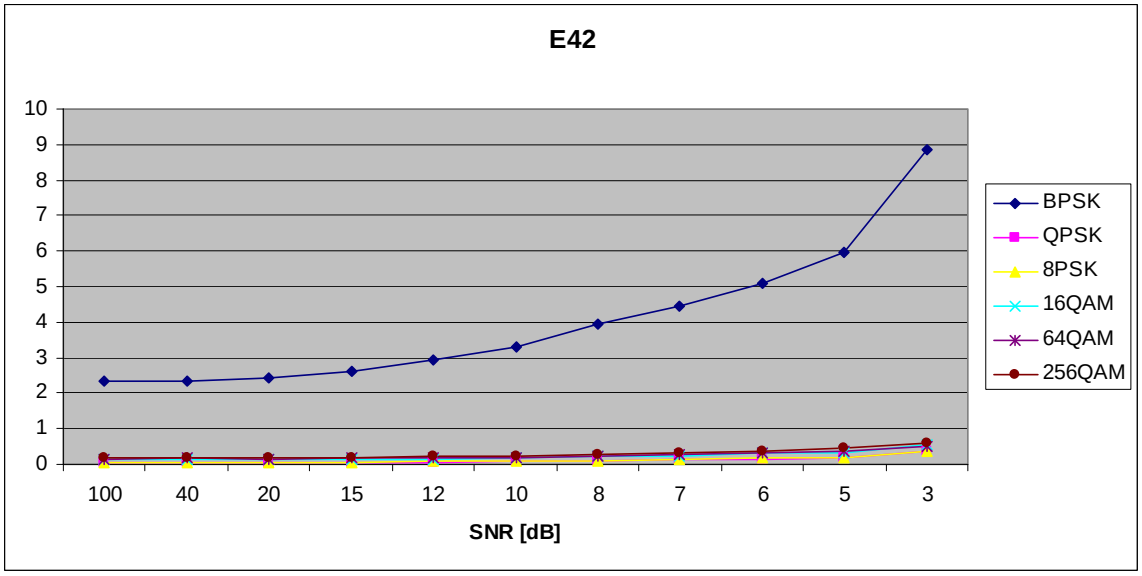

Figure 232. $E_{x,4,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

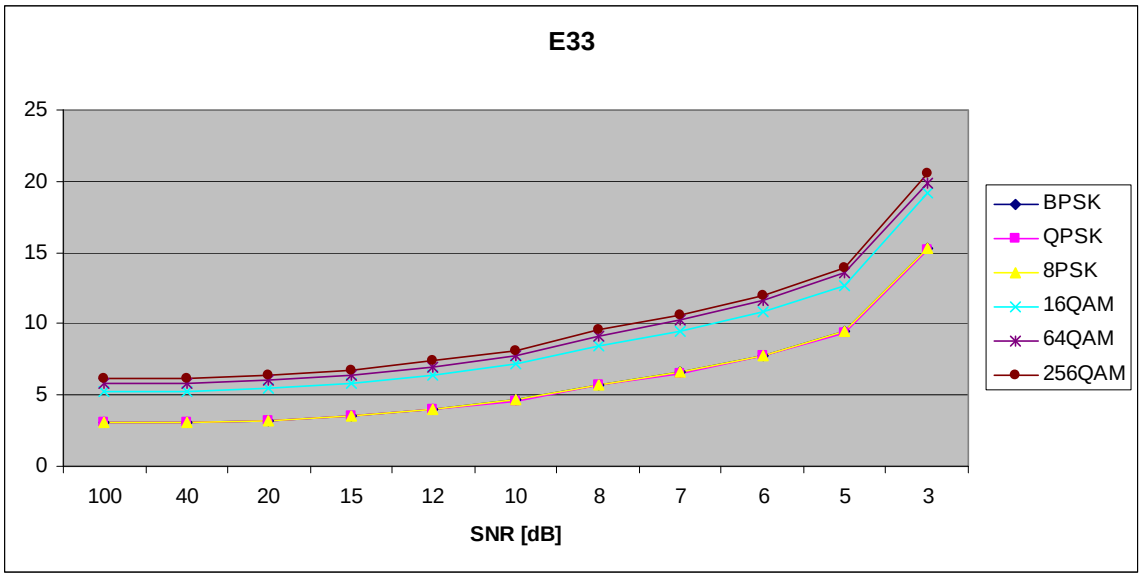

Figure 233. $E_{x,3,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

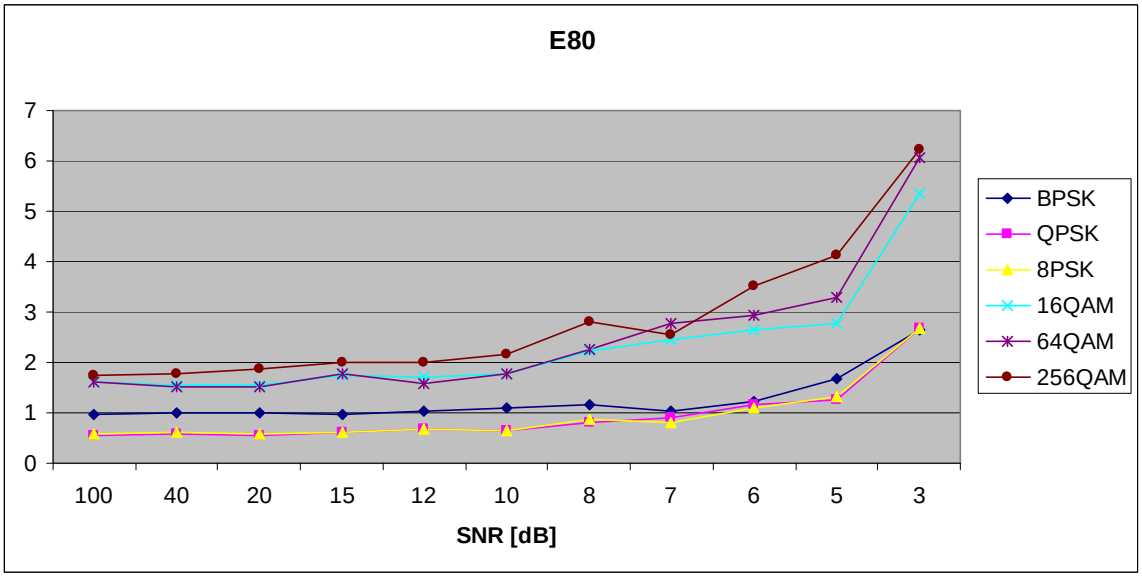

Figure 234. $E_{x,8,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

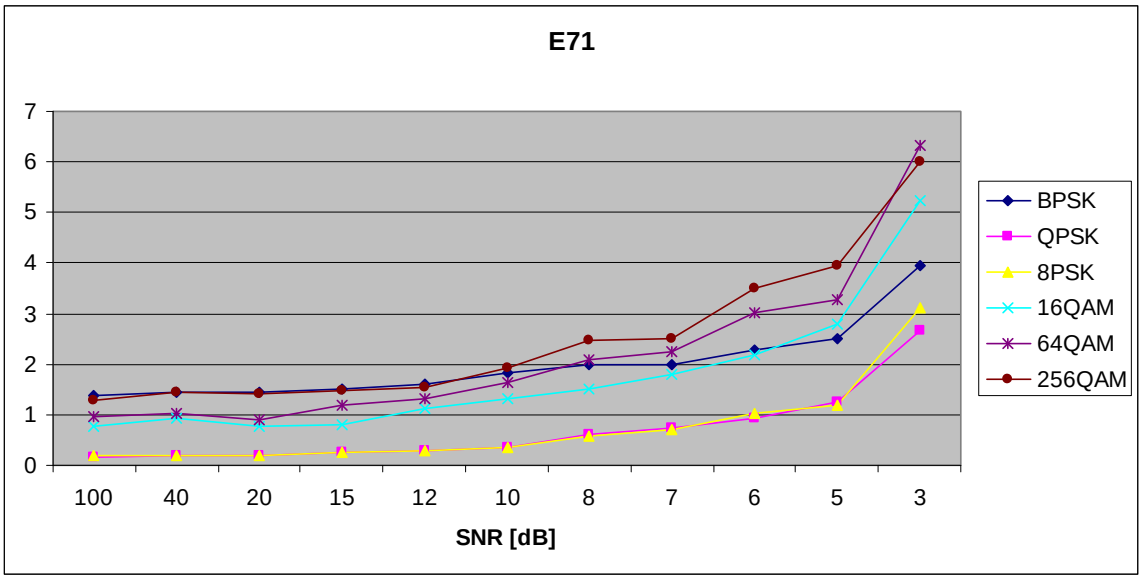

Figure 235. $E_{x,7,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

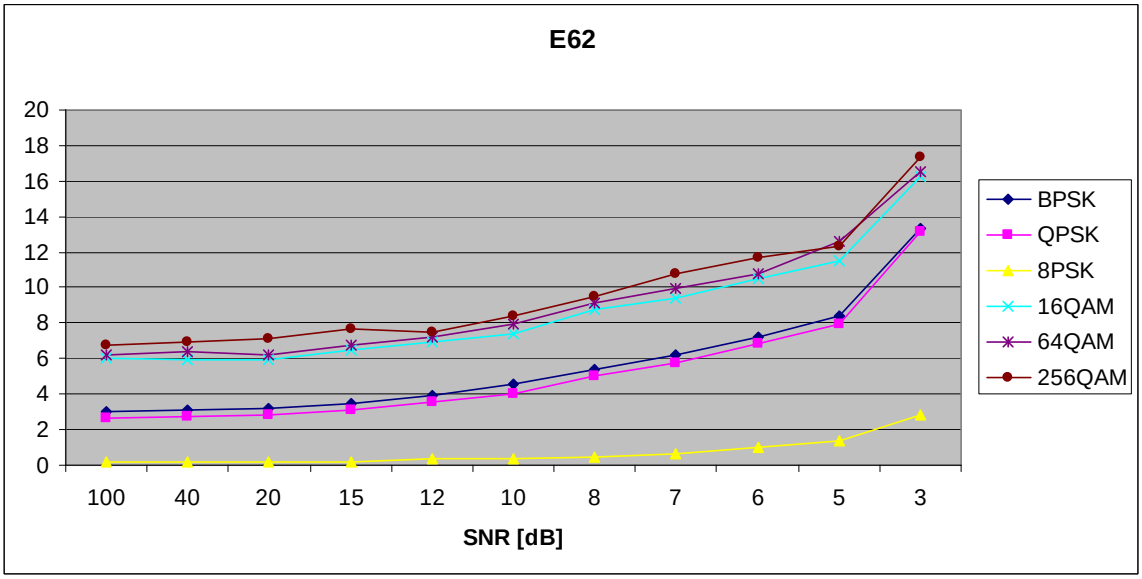

Figure 236. $E_{x,6,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

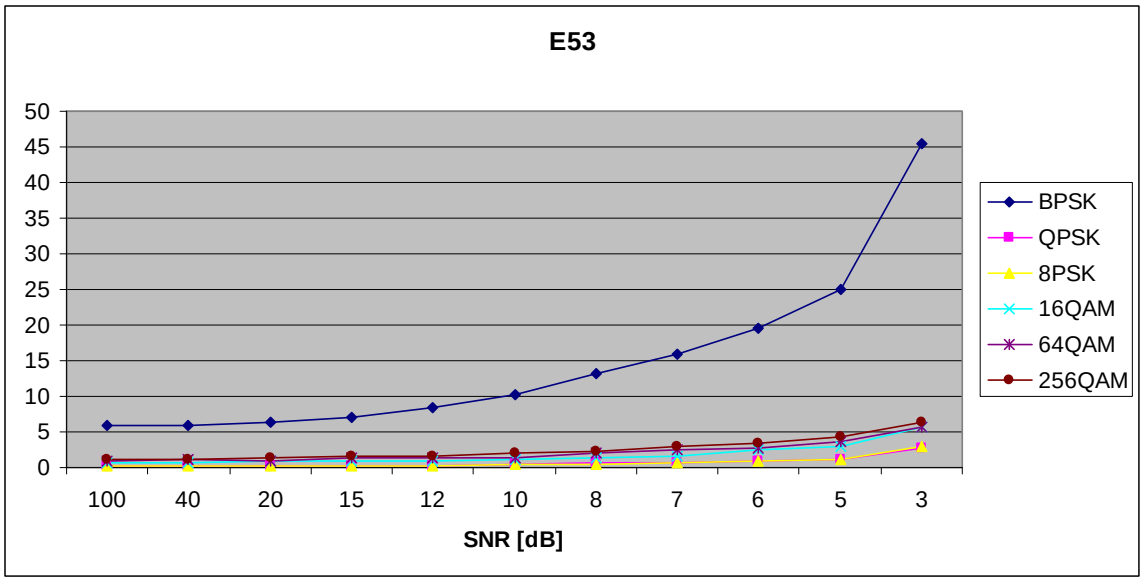

Figure 237. $E_{x,5,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

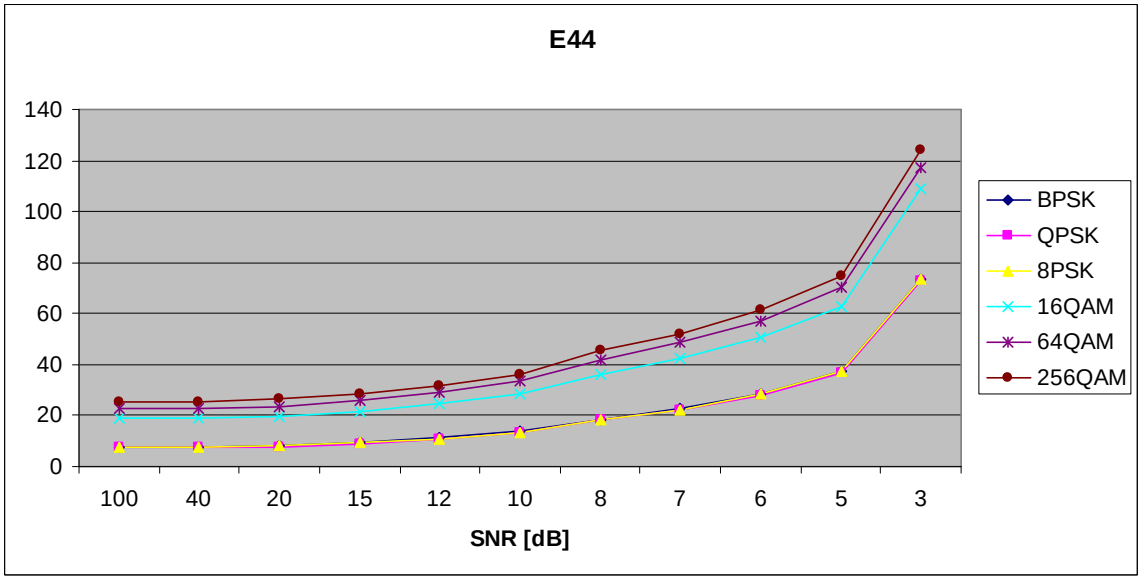

Figure 238. $E_{x,4,4}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

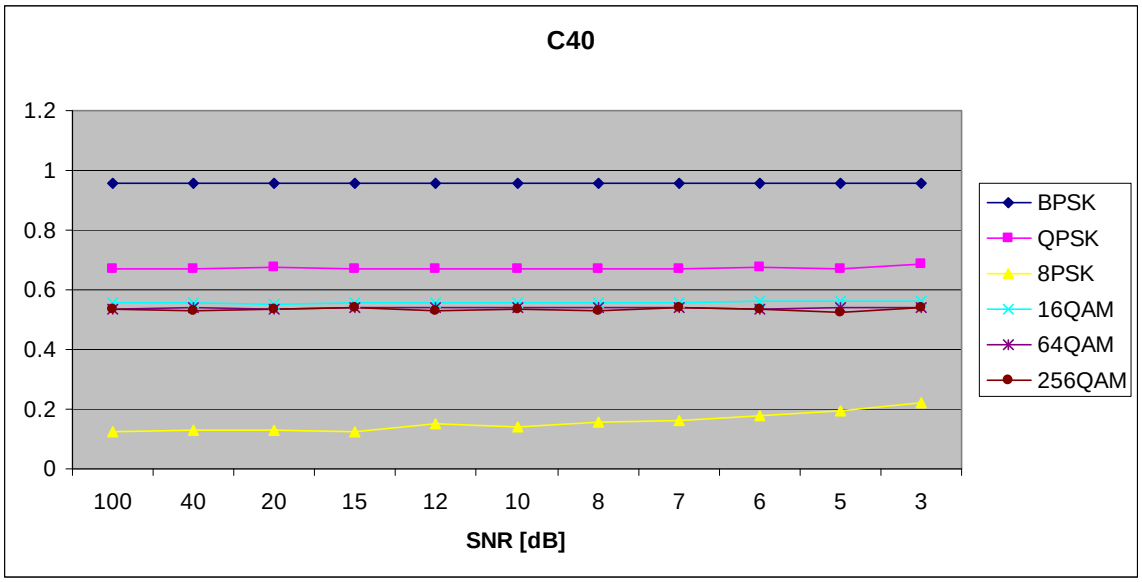

Figure 239. $C_{x,4,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

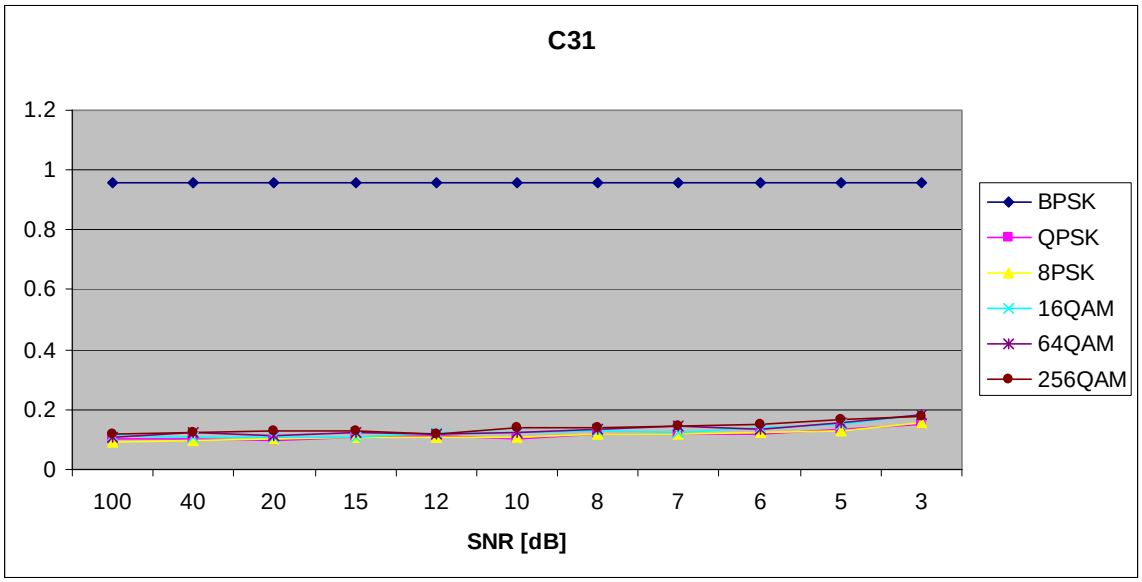

Figure 240. $C_{x,3,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

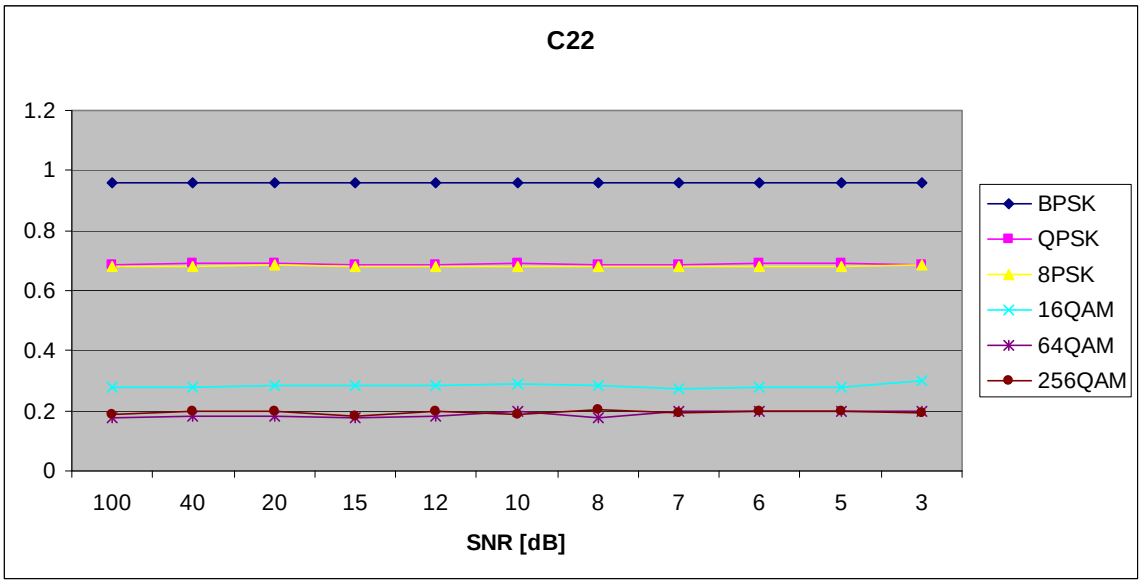

Figure 241. $C_{x,2,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

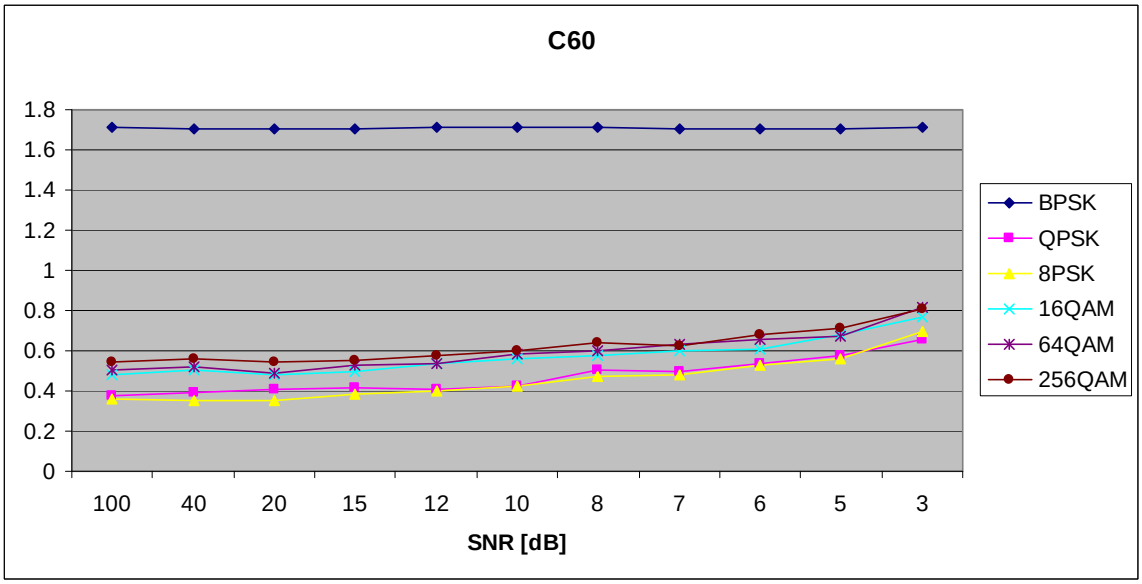

Figure 242. $C_{x,6,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

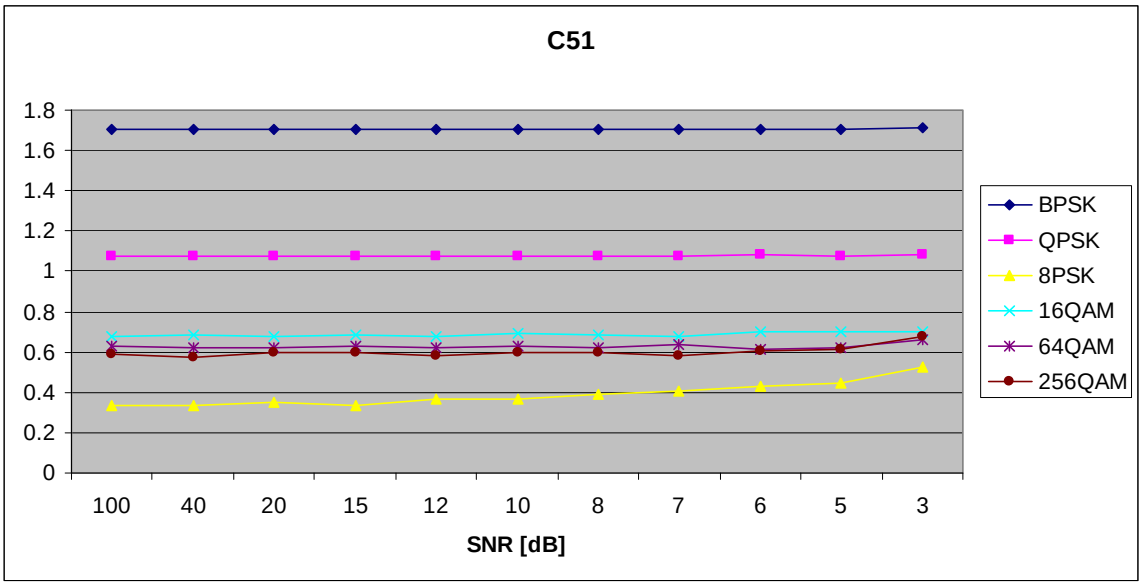

Figure 243. $C_{x,5,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

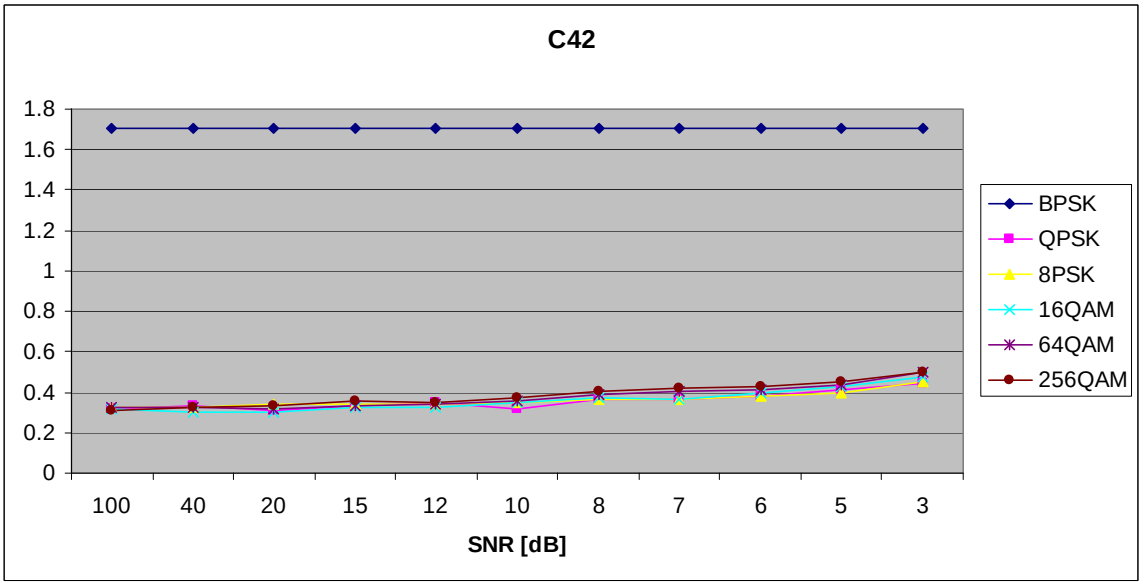

Figure 244. $C_{x,4,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

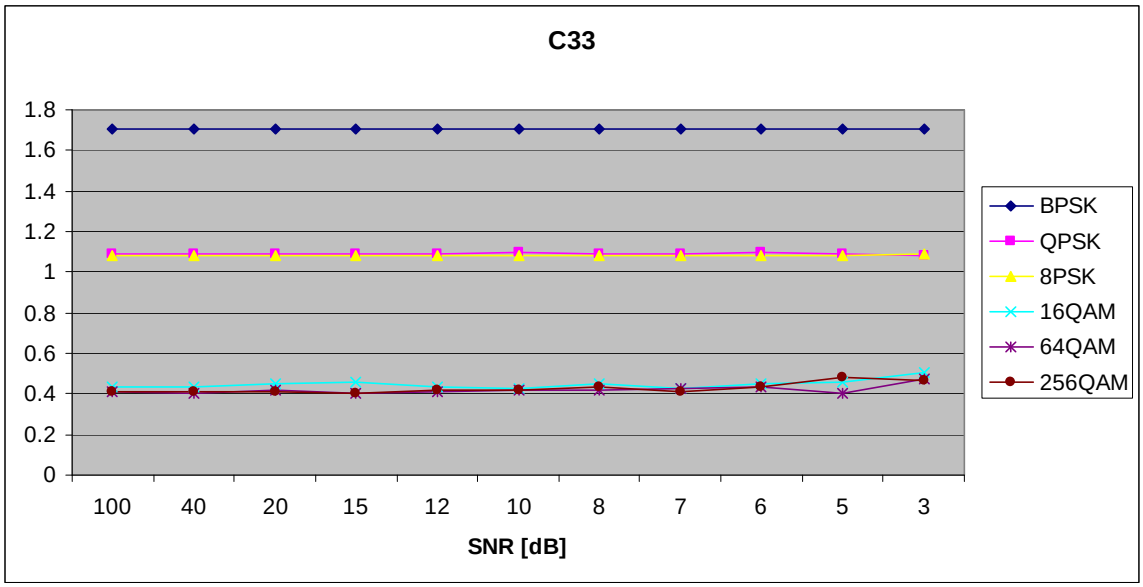

Figure 245. $C_{x,3,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

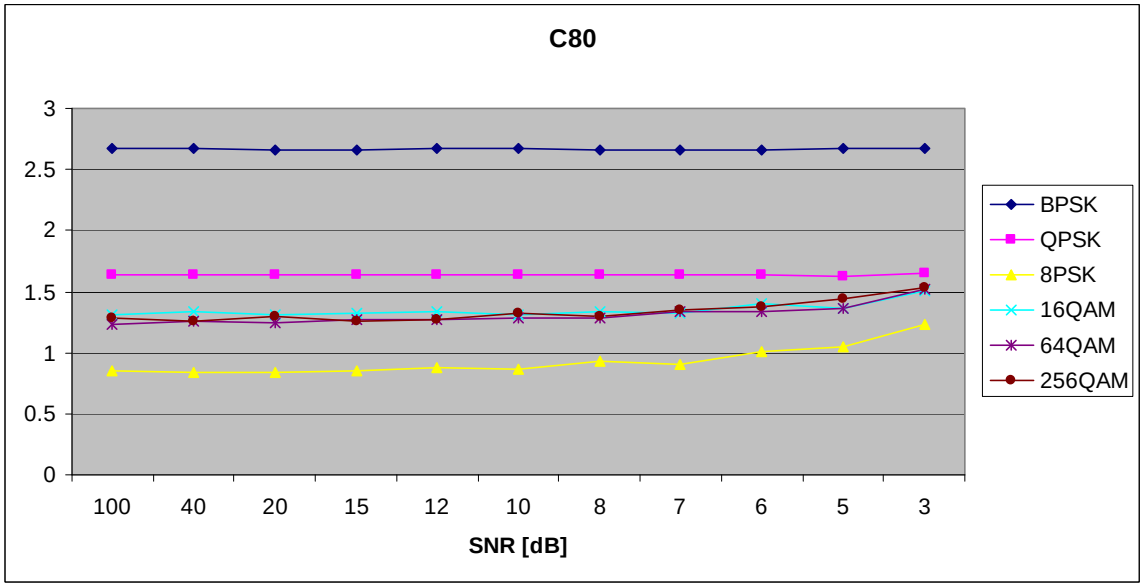

Figure 246. $C_{x,8,0}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

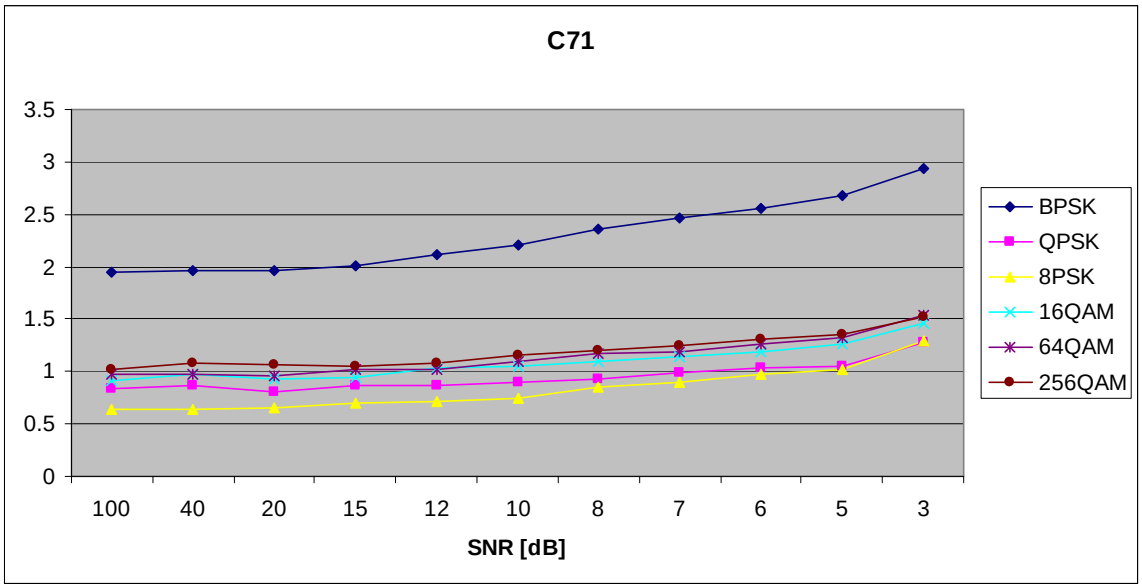

Figure 247. $C_{x,7,1}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

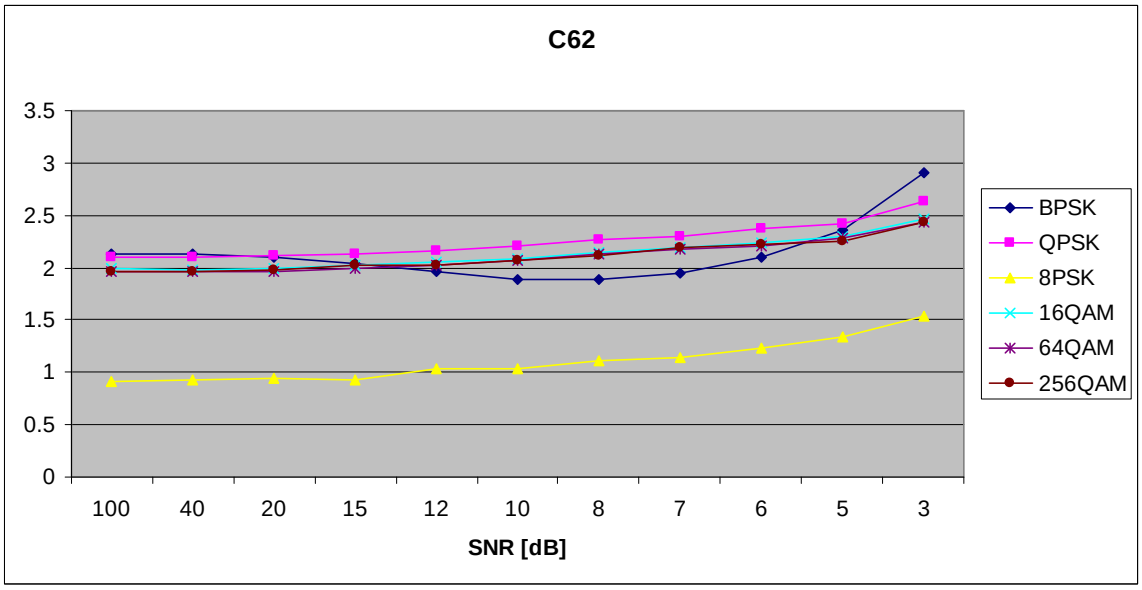

Figure 248. $C_{x,6,2}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

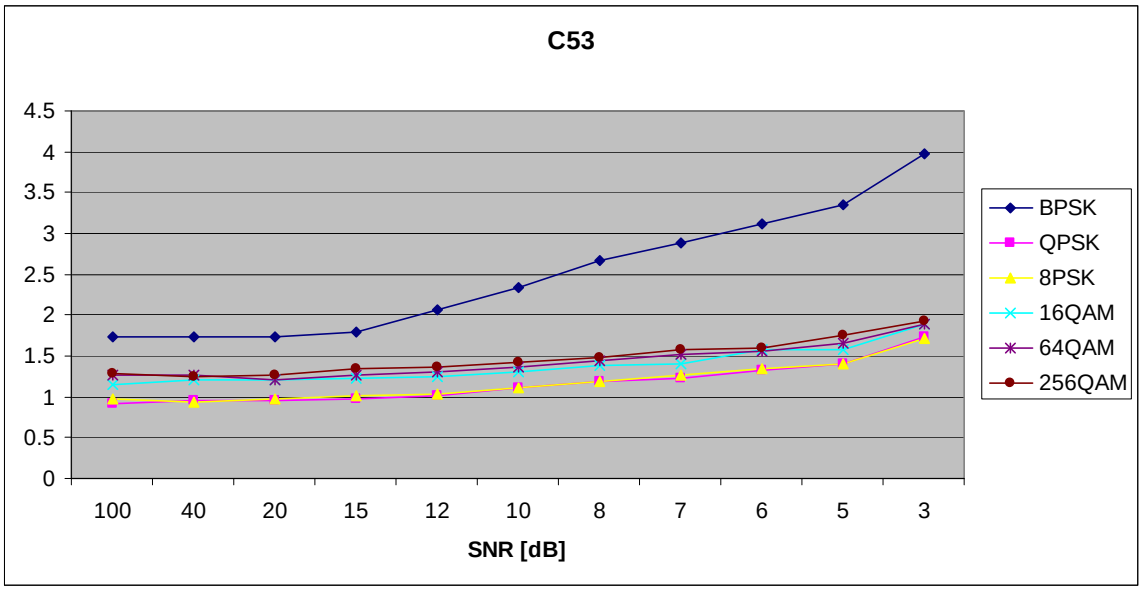

Figure 249. $C_{x,5,3}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

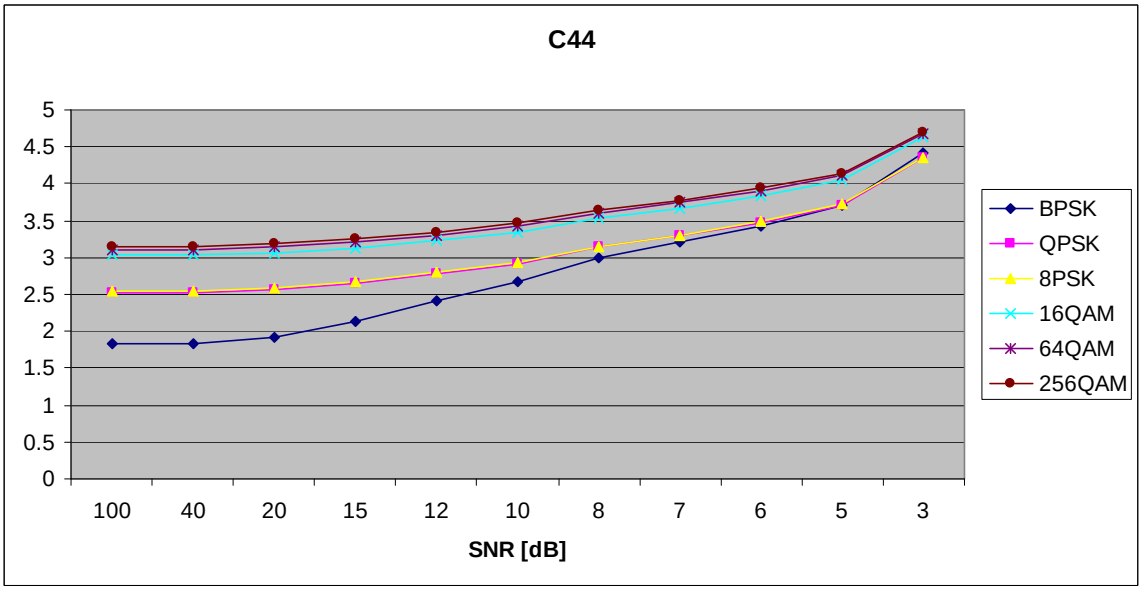

Figure 250. $C_{x,4,4}$ in AWGN and Fast, Frequency-Selective Ricean Fading.

APPENDIX C: EFFECT OF CONSTELLATION ROTATION ON MOMENTS AND CUMULANTS FOR PSK AND QAM SIGNALS

A. MOMENTS

1. PSK Signals

The complex envelope expression for a PSK signal is given as

$$x(k) = A\exp\left(j\left(\frac{2\pi}{N} d(k) \right) \right), \tag{C.1}$$

where A is the amplitude, N is the number of symbols, $d(k)$ is a uniformly distributed random integer between 0 and N-1. Let $x'(k)$ be the same signal after an arbitrary phase shift is introduced:

$$x'(k) = A\exp\left(j\left(\frac{2\pi}{N} d(k) + \phi \right) \right). \tag{C.2}$$

The general expression for the $(p+q)$th-order moment of $x'(k)$ is defined as

$$E_{x',p,q} = E\left[\left(A\exp\left(j\left(\frac{2\pi}{N} d(k) + \phi \right) \right) \right)^p \left(A\exp\left(-j\left(\frac{2\pi}{N} d(k) + \phi \right) \right) \right)^q \right] \tag{C.3}$$

which may be rewritten as

$$\begin{aligned} E_{x',p,q} &= \exp(j\phi(p-q)) A^{p+q} E\left[\exp\left(j\frac{2\pi(p-q)}{N} d(k) \right) \right] \\ &= \exp(j\phi(p-q)) E_{x,p,q}. \end{aligned} \tag{C.4}$$

2. QAM Signals

The complex envelope expression for a QAM signal is

$$x(k) = a(k) + jb(k), \tag{C.5}$$

or, in polar form,

$$x(k) = \sqrt{(a(k))^2 + (b(k))^2}\exp\left(j\tan^{-1}\frac{b(k)}{a(k)} \right), \tag{C.6}$$

where $a(k)$ and $b(k)$ are random variables representing the I and Q channels, respectively. Let $x'(k)$ be the same signal after an arbitrary phase shift is introduced:

$$x'(k) = \sqrt{\left(a(k)\right)^2 + \left(b(k)\right)^2}\exp\left(j\left(\tan^{-1}\frac{b(k)}{a(k)} + \phi\right)\right). \tag{C.7}$$

The $(p+q)$th-ordered moment obtained for this signal is given by

$$\begin{aligned} E_{x',p,q} &= E\left[\left(\sqrt{\left(a(k)\right)^2 + \left(b(k)\right)^2}\exp\left(j\left(\tan^{-1}\frac{b(k)}{a(k)} + \phi\right)\right)\right)^p \left(\sqrt{\left(a(k)\right)^2 + \left(b(k)\right)^2}\exp\left(-j\left(\tan^{-1}\frac{b(k)}{a(k)} + \phi\right)\right)\right)^q\right] \\ &= \exp\left(j\phi\left(p-q\right)\right)E\left[\left(\sqrt{\left(a(k)\right)^2 + \left(b(k)\right)^2}\exp\left(j\left(\tan^{-1}\frac{b(k)}{a(k)}\right)\right)\right)^p \left(\sqrt{\left(a(k)\right)^2 + \left(b(k)\right)^2}\exp\left(-j\left(\tan^{-1}\frac{b(k)}{a(k)}\right)\right)\right)^q\right] \\ &= \exp\left(j\phi\left(p-q\right)\right)E_{x,p,q}. \end{aligned} \tag{C.8}$$

B. CUMULANTS

From the proof above and the expressions in Appendix A, it will be shown that the magnitude of most of the cumulants is unaffected by an arbitrary phase shift. In the paragraphs to follow, $E'_{x,p,q}$ refers to the moment $E_{x,p,q}$ whose underlying signal has been shifted by an arbitrary phase ϕ. Note that there is no need to consider $C_{x,2,0}$ and $C_{x,1,1}$ as they are mathematically identical to $E_{x,2,0}$ and $E_{x,1,1}$.

1. $C_{x,4,0}$

$$\begin{aligned} C'_{x,4,0} &= E'_{x,4,0} - 3\left(E'_{x,2,0}\right)^2 \\ &= \exp\left(j4\phi\right)E_{x,4,0} - 3\left(\exp\left(j2\phi\right)E_{x,2,0}\right)^2 \\ &= \exp\left(j4\phi\right)\left(E_{x,4,0} - 3\left(E_{x,2,0}\right)^2\right) \\ &= \exp\left(j4\phi\right)C_{x,4,0}. \end{aligned}$$

2. $C_{x,3,1}$

$$\begin{aligned}
C'_{x,3,1} &= E'_{x,3,1} - 3E'_{x,2,0}E'_{x,1,1} \\
&= \exp(j2\phi)E_{x,3,1} - 3\exp(j2\phi)E_{x,2,0}E_{x,1,1} \\
&= \exp(j2\phi)\left(E_{x,3,1} - 3E_{x,2,0}E_{x,1,1}\right) \\
&= \exp(j2\phi)C_{x,3,1}.
\end{aligned}$$

3. $C_{x,2,2}$

$C_{x,2,2}$ is not magnitude-invariant to phase shifts.

4. $C_{x,6,0}$

$$\begin{aligned}
C'_{x,6,0} &= E'_{x,6,0} - 15E'_{x,2,0}E'_{x,4,0} + 30\left(E'_{x,2,0}\right)^3 \\
&= \exp(j6\phi)E_{x,6,0} - 15\exp(j2\phi)E_{x,2,0}\exp(j4\phi)E_{x,4,0} + 30\left(\exp(j2\phi)E_{x,2,0}\right)^3 \\
&= \exp(j6\phi)\left(E_{x,6,0} - 15E_{x,2,0}E_{x,4,0} + 30\left(E_{x,2,0}\right)^3\right) \\
&= \exp(j6\phi)C_{x,6,0}.
\end{aligned}$$

5. $C_{x,5,1}$

$$\begin{aligned}
C'_{x,5,1} &= E'_{x,5,1} - 10E'_{x,2,0}E'_{x,3,1} - 5E'_{x,1,1}E'_{x,4,0} + 30\left(E'_{x,2,0}\right)^2 E'_{x,1,1} \\
&= \exp(j4\phi)E_{x,5,1} - 10\exp(j2\phi)E_{x,2,0}\exp(j2\phi)E_{x,3,1} - 5E_{x,1,1}\exp(j4\phi)E_{x,4,0} \\
&\quad + 30\left(\exp(j2\phi)E_{x,2,0}\right)^2 E_{x,1,1} \\
&= \exp(j4\phi)\left(E_{x,5,1} - 10E_{x,2,0}E_{x,3,1} - 5E_{x,1,1}E_{x,4,0} + 30\left(E_{x,2,0}\right)^2 E_{x,1,1}\right) \\
&= \exp(j4\phi)C_{x,5,1}.
\end{aligned}$$

6. $C_{x,4,2}$

$C_{x,4,2}$ is not magnitude-invariant to phase shifts.

7. $C_{x,3,3}$

$C_{x,3,3}$ is not magnitude-invariant to phase shifts.

8. $C_{x,8,0}$

$$\begin{aligned}
C'_{x,8,0} &= E'_{x,8,0} - 35\left(E'_{x,4,0}\right)^2 - 630\left(E'_{x,2,0}\right)^4 + 420\left(E'_{x,2,0}\right)^2 E'_{x,4,0} \\
&= \exp(j8\phi)E_{x,8,0} - 35\left(\exp(j4\phi)E_{x,4,0}\right)^2 - 630\left(\exp(j2\phi)E_{x,2,0}\right)^4 \\
&\quad +420\left(\exp(j2\phi)E_{x,2,0}\right)^2 \exp(j4\phi)E_{x,4,0} \\
&= \exp(j8\phi)\left(E_{x,8,0} - 35\left(E_{x,4,0}\right)^2 - 630\left(E_{x,2,0}\right)^4 + 420\left(E_{x,2,0}\right)^2 E_{x,4,0}\right) \\
&= \exp(j8\phi)C_{x,8,0}.
\end{aligned}$$

9. $C_{x,7,1}$, $C_{x,6,2}$, $C_{x,5,3}$, and $C_{x,4,4}$

$C_{x,7,1}$, $C_{x,6,2}$, $C_{x,5,3}$, and $C_{x,4,4}$ are not magnitude-invariant to phase shifts.

APPENDIX D: MATLAB CODE

A. LINEAR CLASSIFIERS

1. Description:

- PCA.m performs Principal Component Analysis on training data.
- LDA.m performs Linear Discriminant Analysis on training data.
- LINEARTESTER.m generates digitally modulated signals by randomly picking the modulation type, then corrupting them with AWGN and (optionally) applying fading. Finally, it uses the projection matrix and class centroids passed as inputs to classify the signals. Works with either class data calculated by either PCA.m or LDA.m

2. PCA.m

```
function [TrainMean centroids projmatrix] = pca(TrainData, Neigvecs, Nclasses, Nsamp)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% PCA.m performs principal component analysis on the column vectors in
% TrainData.

% Inputs: TrainData (Data to be used for training the classifier, stored as
% column vectors)
% Neigvecs: number of eigenvectors to be used in the projection matrix
% Nclasses: number of classes present in the training data
% Nsamp: number of samples per class

% Calculate mean training vector and subtract from each vector to form a
% centered training matrix
```

```
TrainMean = mean(TrainData,2);
CentTrainData = (TrainData - TrainMean*ones(1,Nclasses*Nsamp));

% Calculate covariance matrix and decompose into eigenvectors. Keep the
% desired number of eigenvectors (Neigvecs) as the projection matrix
covarmatrix = CentTrainData*conj(CentTrainData)';
[eigvecs,eigvals] = eig(covarmatrix);
SortedEigvals = sort(eigvals,1,'descend');
SortedEigvals = SortedEigvals(1,:);
[SortedEigvals,index] = sort(SortedEigvals,2,'descend');
projmatrix = eigvecs(:,index(1:Neigvecs));

% Calculate class centroids by projecting training data for each class and
% calculating the mean
centroids = zeros(size(projmatrix,2),Nclasses);
for n = 1:Nclasses
   projclass = projmatrix'*CentTrainData(:,[(n-1)*Nsamp+1:n*Nsamp]);
   centroids(:,n) = mean(projclass,2);
end
```

3. LDA.m

```
function [TrainMean centroids W] = lda(TrainData, Neigvecs, Nclasses, Nsamp)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% LDA.m performs linear discrimant analysis on the column vectors in
% TrainData.

% Inputs: TrainData (Data to be used for training the classifier, stored as
% column vectors)
% Neigvecs: number of eigenvectors to be used in the projection matrix
% Nclasses: number of classes present in the training data
```

```
% Nsamp: number of samples per class

% First determine the mean training vector and subtract it from the
% training data to form the centered training matrix. Initialize the
% within- and between-class scatter matrices Sw and Sb and the matrix of
% class centroids.
r = size(TrainData,1);
TrainMean = mean(TrainData,2);
CentTrainData = (TrainData - TrainMean*ones(1,Nclasses*Nsamp));
Sw = zeros(r,r);
Sb = zeros(r,r);
centroids = zeros(Neigvecs,Nclasses);

% Calculate Sw and Sb
for n = 1:Nclasses
   Si = zeros(r,r);
   mu = mean(CentTrainData(:,(n-1)*Nsamp+1:n*Nsamp),2);
   for m = 1:Nsamp
      x = CentTrainData(:,(n-1)*Nsamp+m);
      Si = Si + (x-mu)*conj((x-mu)');
   end
   Sb = Sb + Nsamp*(TrainMean - mu)*conj((TrainMean - mu)');
   Sw = Sw + Si;
end

% Decompose inv(Sw)*Sb into eigenvectors and keep the top Neigvecs of them
% as the projectoin matrix
[eigvecs, lambda] = eig(inv(Sw)*Sb);
lambda = sort(lambda,1,'descend');
lambda = lambda(1,:);
[lambda,index] = sort(lambda,2,'descend');
W = eigvecs(:,index(1:Neigvecs));
```

```
% Find class centroids by projecting each class and taking the mean
for n = 1:Nclasses
   centroids(:,n) = mean((W'*CentTrainData(:,(n-1)*Nsamp+1:n*Nsamp)),2);
end
```

4. LINEARTESTER.m

```
function[tracker,percent]=lineartester(numtrials,msglength,ModSchemes,symorder,SymSNR,MeanProf,centroids,projmatrix,channeltype)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% LINEARTESTER.m generates digitally modulated signals by randomly picking
% the modulation type, then corrupts it with AWGN and (optionally) fading.
% Finally, uses the projection matrix and class centroids passed as inputs
% to classify the signals. Works with either lda.m or pca.m.

% Inputs:
% numtrials: number of signals to generate
% msglength: length of each signal
% ModSchemes: vector containing number of distinct symbols for each scheme.
% Usually [2 4 8 2 4 8 16 64 256] for BFSK, QFSK, 8FSK, BPSK, QPSK, 8PSK,
% 16QAM, 64QAM, 256QAM.
% symorder: 'gray' or 'binary'
% SymSNR: SNR used to corrupt signals with AWGN
% MeanProf: the mean profile of the training data (determined by lda.m or
% pca.m), to be subtracted from each signal.
% centroids: Class centroids determined by lda.m or pca.m
% projmatrix: Projection matrix determined by lda.m or pca.m
% channeltype: 1 = AWGN only, 2 = Rayleigh, 3 = Ricean, 4 = mixed.
```

```
NumSchemes = length(ModSchemes);

% tracker will keep track of the results as a confusion matrix
tracker = zeros(NumSchemes);

% define modulator objects for PSK and QAM signals
bpskmodulator = modem.pskmod('M',2,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
qpskmodulator = modem.pskmod('M',4,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
epskmodulator = modem.pskmod('M',8,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
modulator16 = modem.qammod('M',16,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
modulator64 = modem.qammod('M',64,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
modulator256 = modem.qammod('M',256,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');

% define parameters for FSK signals
freqsep = 16;
Nsamp = 16;
Fs = 128;

% The following thresholds for distinguishing between BFSK, QFSK, and 8FSK
% were determined empirically.
thold23 = 2;
thold45 = 1.5;
thold89 = 1.15;

% Define fading channel parameters
symrate = 1e6;
raychan = rayleighchan(1/symrate,5000,[0, 2e-6], [0,-10]);
richan = ricianchan(1/symrate,5000,3,[0, 2e-6], [0,-10]);
```

```
h=waitbar(0,'Conducting Trials...');
for n = 1:numtrials
   % Pick modulation scheme at random and generate msg
   modtype = randsrc(1,1,1:NumSchemes);
   M = ModSchemes(modtype);
   msg = randsrc(msglength,1,[0:M-1]);
   switch modtype
      case {1,2,3}
         modmsg = fskmod(msg,M,freqsep,Nsamp,Fs,'cont',symorder);
      case 4
         modmsg = modulate(bpskmodulator,msg);
      case 5
         modmsg = modulate(qpskmodulator,msg);
      case 6
         modmsg = modulate(epskmodulator,msg);
      case 7
         modmsg = modulate(modulator16,msg)./sqrt(10);
      case 8
         modmsg = modulate(modulator64,msg)./sqrt(42);
      case 9
         modmsg = modulate(modulator256,msg)./sqrt(170);
   end

   % Use channel type to determine the propagation model to use
   if channeltype == 4
      propmodel = randsrc(1,1,1:3);
   else
      propmodel = channeltype;
   end

   % Apply the appropriate model. AWGNFreePower is a measurement of the
   % signal power before adding AWGN.
```

```
switch propmodel
   case 1
      AWGNFreePower = sqrt(mean(modmsg.*conj(modmsg)));
      noisymsg = awgn(modmsg,SymSNR,'measured');
   case 2
      multipath = filter(raychan,modmsg(1:msglength));
      AWGNFreePower = sqrt(mean(multipath.*conj(multipath)));
      noisymsg = awgn(multipath,SymSNR,'measured');
   case 3
      multipath = filter(richan,modmsg(1:msglength));
      AWGNFreePower = sqrt(mean(multipath.*conj(multipath)));
      noisymsg = awgn(multipath,SymSNR,'measured');
end

% Normalize by AWGNFreePower
noisymsg = noisymsg./AWGNFreePower;

%First check to see if the signal was FSK
sfft = fft(noisymsg(1:4096));
sortfft = sort(abs(sfft),1,'descend');
ratio23 = sortfft(2)/sortfft(3);
ratio45 = sortfft(4)/sortfft(5);
ratio89 = sortfft(8)/sortfft(9);

if ratio23>thold23
   modguess = 1;
else if ratio45>thold45
      modguess = 2;
   else if ratio89>thold89
         modguess = 3;
      else
         % If not FSK, subtract the mean training vector, project
```

```
                % the result, and calculate the distance to all centroids.
                % Closest distance wins
                sigprof = (profgen(noisymsg) - MeanProf);
                projdata = projmatrix'*sigprof;
                euclidnorm = sqrt(sum((centroids - projdata*ones(1,NumSchemes-
3)).^2));
                [dist,modguess] = min(euclidnorm);
                modguess = modguess+3;
            end
        end
    end
    tracker(modtype,modguess) = tracker(modtype,modguess) + 1;
    waitbar(n/numtrials,h);
end
close(h);
% Calculate overall performance
percent = sum(diag(tracker))/numtrials;
```

B. NONLINEAR CLASSIFIERS

1. Description:

- KPCA.m performs kernel-based (nonlinear) principal component analysis on training data.
- GDA.m performs general discriminant analysis on training data.
- NONLINTESTER.m generates digitally modulated signals by randomly picking the modulation type, then corrupting them with AWGN and (optionally) applying fading. Finally, it uses the paramters calculated by either KPCA.m or GDA.m to classify the signal.

2. KPCA.m

```
function [TrainMean CentTrainData centroids K M1 u] = kpca(TrainData, Nclasses, Nsamp, maptype,a,b,d)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% KPCA.m performs kernel-based (nonlinear) principal component analysis on
% the column vectors in TrainData.

% Inputs: TrainData (Data to be used for training the classifier, stored as
% column vectors)
% Neigvecs: number of eigenvectors to be used in the projection matrix
% Nclasses: number of classes present in the training data
% Nsamp: number of samples per class
% maptype: Type of kernel. 'gauss', 'sig', or 'poly.'
% a,b,d: kernel paramters (set to 0 if not used by your kernel)

% Calculate mean training vector and subtract from each vector to form a
% centered training matrix
TrainMean = mean(TrainData,2);
CentTrainData = TrainData - TrainMean*ones(1,Nclasses*Nsamp);

% Call KERNEL.m to calculate kernel matrix
K = kernel(CentTrainData,CentTrainData,maptype,a,b,d);
% M1 = (1/no. of training vectors) * I, used to center the kernelized
% training data matrix in the higher-dimensional space
M1 = (1/(Nclasses*Nsamp))*ones(size(K));
K = K-M1*K-K*M1+M1*K*M1;

% Eigen-decomposition of Kernel matrix. Sort vectors and values in
% descending order and keep any that are at least 1/100 of the magnitude of
% the largest eigenvalue
[u,gamma] = eig(K);
```

```
[sortgamma,index] = sort(diag(gamma),1,'descend');
u = u(:,index);
sortgamma = sortgamma(sortgamma>max(sortgamma)/100);
gamma = diag(sortgamma);
u = u(:,1:length(sortgamma));

% K must be reformed from the most significant eigenvectors and eigenvalues
% in order for the dimensions to match.
K = u*gamma*u';
% Normalize the eigenvectors. This will form the projection matrix
u = u./(ones(size(u,1),1)*sum(sqrt(u'*K*u)));

% Find class centroids by projecting the kernel matrix class by class and
% calculating the mean.
ProjData = (K*u)';
centroids = zeros(size(u,2),Nclasses);
for n = 1:Nclasses
   projclass = ProjData(:,(n-1)*Nsamp+1:n*Nsamp);
   centroids(:,n) = mean(projclass,2);
end
```

3. GDA.m

```
function [TrainMean CentTrainData centroids K M1 alpha] = gda(TrainData, Nclasses, Nsamp, maptype, a,b,d)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% GDA.m performs general discriminant analysis on the column vectors in
% TrainData.

% Inputs: TrainData (Data to be used for training the classifier, stored as
% column vectors)
```

```
% Neigvecs: number of eigenvectors to be used in the projection matrix
% Nclasses: number of classes present in the training data
% Nsamp: number of samples per class
% maptype: Type of kernel. 'gauss', 'sig', or 'poly.'
% a,b,d: kernel paramters (set to 0 if not used by your kernel)

% Calculate mean training vector and subtract from each vector to form a
% centered training matrix
TrainMean = mean(TrainData,2);
CentTrainData = TrainData - TrainMean*ones(1,Nclasses*Nsamp);

% Call KERNEL.m to calculate kernel matrix
K = kernel(CentTrainData,CentTrainData,maptype,a,b,d);
% M1 = (1/no. of training vectors) * I, used to center the kernelized
% training data matrix in the higher-dimensional space
M1 = (1/(Nclasses*Nsamp))*ones(size(K));
K = K-M1*K-K*M1+M1*K*M1;

% Eigen-decomposition of Kernel matrix. Sort vectors and values in
% descending order and keep any that are at least 1/100 of the magnitude of
% the largest eigenvalue
[u,gamma] = eig(K);
[sortgamma,index] = sort(diag(gamma),1,'descend');
u = u(:,index);
sortgamma = sortgamma(sortgamma>max(sortgamma)/100);
gamma = diag(sortgamma);
u = u(:,1:length(sortgamma));
K = u*gamma*u';

% Form the block diagonal matrix W. There will be one block for each class
% (modulation type). Each block will be equal to (1/no. of samples per
% class) * ones(no. of samples per class)
```

```
Wl = (1/Nsamp)*ones(Nsamp);
W = Wl;

for m = 1:Nclasses-1
   W = blkdiag(W,Wl);
end

% Eigendecomposition of u'Wu. Vectors are used to calculate the projection
% matrix alpha.
[beta,lambda] = eig(u'*W*u);
alpha = u*inv(gamma)*beta;
% Normalize all vectors in alpha. This is the projection matrix.
alpha = alpha./(ones(size(alpha,1),1)*sum(sqrt(alpha'*K*alpha)));

% Project kernelized training data and calculate class centroids
ProjData = (K*alpha)';
centroids = zeros(size(alpha,2),Nclasses);

for n = 1:Nclasses
   projclass = ProjData(:,(n-1)*Nsamp+1:n*Nsamp);
   centroids(:,n) = mean(projclass,2);
end
```

4. NONLINTESTER.m

```
function [tracker,percent] = nonlintester(numtrials,msglength,ModSchemes,symorder,SymSNR,MeanProf,CentProfs,K,M1,centroids,alpha,maptype,a,b,d,channeltype)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% NONLINTESTER.m generates digitally modulated signals by randomly picking
% the modulation type, then corrupts it with AWGN and (optionally) fading.
```

```
% Finally, uses the paramters calculated by either KPCA.m or GDA.m to
% classify the signal.

% Inputs:
% numtrials: number of signals to generate
% msglength: length of each signal
% ModSchemes: vector containing number of distinct symbols for each scheme.
% Usually [2 4 8 2 4 8 16 64 256] for BFSK, QFSK, 8FSK, BPSK, QPSK, 8PSK,
% 16QAM, 64QAM, 256QAM.
% symorder: 'gray' or 'binary'
% SymSNR: SNR used to corrupt signals with AWGN
% MeanProf: the mean profile of the training data (determined by lda.m or
% pca.m), to be subtracted from each signal.
% CentProfs: Matrix of centered training profiles (from KPCA.m or GDA.m)
% K: Kernel matrix (from GDA.m or KPCA.m)
% M1: matrix used to center the testing signals in the higher-dimensional
% space (see GDA.m or KPCA.m)
% centroids: Class centroids determined by GDA.m or KPCA.m
% alpha: Projection matrix determined by GDA.m or KPCA.m
% maptype: Type of kernel. 'gauss', 'sig', or 'poly.'
% a,b,d: kernel paramters (set to 0 if not used by your kernel)
% channeltype: 1 = AWGN only, 2 = Rayleigh, 3 = Ricean, 4 = mixed.

NumSchemes = length(ModSchemes);
% tracker will keep track of the results as a confusion matrix
tracker = zeros(length(ModSchemes));

% define modulator objects for PSK and QAM signals
bpskmodulator =
modem.pskmod('M',2,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
qpskmodulator =
modem.pskmod('M',4,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
```

```
    epskmodulator = modem.pskmod('M',8,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
    modulator16 = modem.qammod('M',16,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
    modulator64 = modem.qammod('M',64,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
    modulator256 = modem.qammod('M',256,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');

    % define parameters for FSK signals
    freqsep = 16;
    Nsamp = 16;
    Fs = 128;

    % The following thresholds for distinguishing between BFSK, QFSK, and 8FSK
    % were determined empirically.
    thold23 = 2;
    thold45 = 1.5;
    thold89 = 1.15;

    % Define fading channel parameters
    symrate = 1e6;
    raychan = rayleighchan(1/symrate,3.5,[0, .1e-6], [0,-10]);
    richan = ricianchan(1/symrate,3.5,3,[0, .1e-6], [0,-10]);

    h=waitbar(0,'Conducting Trials...');
    for n = 1:numtrials
       % Pick modulation scheme at random and generate msg
       modtype = randsrc(1,1,1:NumSchemes);
       M = ModSchemes(modtype);
       msg = randsrc(msglength,1,[0:M-1]);
       switch modtype
          case {1,2,3}
```

```
        modmsg = fskmod(msg,M,freqsep,Nsamp,Fs,'cont',symorder);
    case 4
        modmsg = modulate(bpskmodulator,msg);
    case 5
        modmsg = modulate(qpskmodulator,msg);
    case 6
        modmsg = modulate(epskmodulator,msg);
    case 7
        modmsg = modulate(modulator16,msg)./sqrt(10);
    case 8
        modmsg = modulate(modulator64,msg)./sqrt(42);
    case 9
        modmsg = modulate(modulator256,msg)./sqrt(170);
end

% Use channel type to determine the propagation model to use
if channeltype == 4
    propmodel = randsrc(1,1,1:3);
else
    propmodel = channeltype;
end

% Apply the appropriate model. AWGNFreePower is a measurement of the
% signal power before adding AWGN.
switch propmodel
    case 1
        AWGNFreePower = sqrt(mean(modmsg.*conj(modmsg)));
        noisymsg = awgn(modmsg,SymSNR,'measured');
    case 2
        multipath = filter(raychan,modmsg(1:msglength));
        AWGNFreePower = sqrt(mean(multipath.*conj(multipath)));
        noisymsg = awgn(multipath,SymSNR,'measured');
```

```
    case 3
        multipath = filter(richan,modmsg(1:msglength));
        AWGNFreePower = sqrt(mean(multipath.*conj(multipath)));
        noisymsg = awgn(multipath,SymSNR,'measured');
end

% Normalize by AWGNFreePower
noisymsg = noisymsg./AWGNFreePower;

%First check to see if the signal was FSK
sfft = fft(noisymsg(1:4096));
sortfft = sort(abs(sfft),1,'descend');
ratio23 = sortfft(2)/sortfft(3);
ratio45 = sortfft(4)/sortfft(5);
ratio89 = sortfft(8)/sortfft(9);

if ratio23>thold23
    modguess = 1;
else if ratio45>thold45
        modguess = 2;
    else if ratio89>thold89
            modguess = 3;
        else
            % If not FSK, subtract the mean training vector. Calculate
            % the higher-dimensional version of the vector (ksignal)
            % and center in the higher-dimensional space using M1p and
            % M1.
            sigprof = (profgen(noisymsg) - MeanProf);
            ksignal = kernel(sigprof,CentProfs,maptype,a,b,d);
            M1p = 1/(size(ksignal,2))*ones(size(ksignal));
            ksignal = ksignal - M1p*K - ksignal*M1 + M1p*K*M1;
            % Project signal and calculate the distance to each
```

```
                % centroid. Smallest distance is chosen as the modulation
                % type.
                projsig = (ksignal*alpha)';
                euclidnorm = sqrt(sum((centroids - projsig*ones(1,NumSchemes-
3)).^2));
                [dist,modguess] = min(euclidnorm);
                modguess = modguess+3;
            end
        end
    end
    tracker(modtype,modguess) = tracker(modtype,modguess) + 1;
    waitbar(n/numtrials,h);
end
close(h);
% Calculate overall performance
percent = sum(diag(tracker))/numtrials;
```

C. SUPPORTING FUNCTIONS

1. Description

- PROFGEN.M calculates higher order statistics of the input signal and stores them in a vector.
- TRAINPROFS.m generates PSK and QAM signals for training classifiers.
- KERNEL.m calculates a kernel matrix given two collection of columnwise input vectors.
- MDIST.m calculates the distance between pairs of vectors stored columnwise in two input matrices.

2. PROFGEN.m

```
function profile = profgen(s)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% PROFGEN.M calculates higher order statistics of input signal s and stores
% them in the vector profile. Note that many are commented out since they
% are not used.

% First ensure that mean of s is 0
[r,c] = size(s);
s = s - ones(r,1)*mean(s);
sbar = conj(s);
% spower = mean(s.*sbar);
% Calculate higher order moments
e20 = (mean(s.^2));
e11 = (mean(s.*sbar));
e40 = (mean(s.^4));
e31 = (mean((s.^3).*sbar));
%e22 = (mean((s.^2).*(sbar.^2)));
%e60 = (mean(s.^6));
e51 = (mean((s.^5).*sbar));
%e42 = (mean((s.^4).*(sbar.^2)));
%e33 = (mean((s.^3).*(sbar.^3)));
e80 = (mean(s.^8));
%e71 = (mean((s.^7).*sbar));
%e62 = (mean((s.^6).*(sbar.^2)));
%e53 = (mean((s.^5).*(sbar.^3)));
%e44 = (mean((s.^4).*(sbar.^4)));

% Calculate cumulants
%c20 = e20;
```

%c11 = e11;

c40 = e40 - 3*e20.^2;

%c31 = e31 - 3*e20.*e11;

%c22 = e22 - e20.^2 - 2*e11.^2;

%c60 = e60 - 15*e20.*e40 + 30*e20.^3;

c51 = e51 - 10*e20.*e31 - 5*e11.*e40 + 30*(e20.^2).*e11;

%c42 = e42 - e20.*e40 - 8*e11.*e31 - 6*e20.*e22 + 6*e20.^3 + 24*(e11.^2).*e20;

%c33 = e33 - 6*e20.*e31 - 9*e11.*e22 + 18*(e20.^2).*e11 + 12*e11.^3;

c80 = e80 - 35*e40.^2 - 630*e20.^4 + 420*(e20.^2).*e40;

%c71 = e71 - 35*e40.*e31 - 630*(e20.^3).*e11 + 210*e40.*e20.*e11 + 210*e20.*e31;

%c62 = e62 - 15*e40.*e22 - 20*e31.^2 + 30*e40.*(e20.^2) + 60*e40.*(e11.^2) + 240*e31.*e11.*e20 + 90*e22.*e20.^2 - 90*e20.^4 - 540*(e20.^2).*e11.^2;

%c53 = e53 - 5*e40.*e31 - 30*e31.*e22 + 90*e31.*e20.^2 + 120*e31.*e11.^2 + 180*e22.*e11.*e20 + 30*e40.*e20.*e11 -270*(e20.^3).*e11 - 360*(e11.^3).*e20;

%c44 = e44 - e40.^2 -18*e22.^2 - 16*e31.^2 - 54*e20.^4 - 144*e11.^4 -

%432*(e20.^2).*e11.^2 + 12*e40.*e20.^2 + 96*e31.*e11.*e20 + 144*e22.*e11.^2 + 72*e22.*e20.^2 + 96*e31.*e20.*e11;

%c20 = c20./spower;

%c11 = c11./spower;

c40 = (c40).^.5;

%c31 = (c31).^.5;

%c22 = (c22).^.5;

%c60 = (c60).^(1/3);

c51 = (c51).^(1/3);

%c42 = (c42).^(1/3);

%c33 = (c33).^(1/3);

c80 = (c80).^.25;

%c71 = (c71).^.25;

%c62 = (c62).^.25;

%c53 = (c53).^.25;

```
%c44 = (c44).^.25;

profile = abs([c40;c51;c80]);
```

3. TRAINPROFS.m

```
function profmatrix = TrainProfs(SymSNR,msglength,symorder,MsgsPerScheme)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% TRAINPROFS.m generates PSK and QAM signals for training classifiers

% Inputs:
% SymSNR: SNR at which the training signals will be generated
% msglength: length of signals
% symorder: 'gray' or 'binary'
% MsgsPerScheme: Number of training signals to generate per modulation
% scheme

numstats = length(profgen(1));%This will determine the size of profmatrix
ModSchemes = [2 4 8 16 64 256];%Number of symbols in each scheme
NumSchemes = length(ModSchemes);

% Modem object for each scheme
bpskmodulator = modem.pskmod('M',2,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
qpskmodulator = modem.pskmod('M',4,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
epskmodulator = modem.pskmod('M',8,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
modulator16 = modem.qammod('M',16,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
```

```
    modulator64 =
modem.qammod('M',64,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');
    modulator256 =
modem.qammod('M',256,'PhaseOffset',0,'SymbolOrder',symorder,'InputType','integer');

    h2 = waitbar(0, 'Generating Training Signals...');
    % Initialize profmatrix, where statistics for each signal will be stored
    profmatrix = zeros(numstats,NumSchemes*MsgsPerScheme);
    for n = 1:length(ModSchemes)
      M = ModSchemes(n);
      for k = 1:MsgsPerScheme
        % Iterate through each modulation scheme, generate signal, add
        % AWGN, and store its statistics in profmatrix
        msg = randsrc(msglength,1,[0:M-1]);
        switch n
          case 1
            modmsg = modulate(bpskmodulator,msg);
          case 2
            modmsg = modulate(qpskmodulator,msg);
          case 3
            modmsg = modulate(epskmodulator,msg);
          case 4
            modmsg = modulate(modulator16,msg)./sqrt(10);
          case 5
            modmsg = modulate(modulator64,msg)./sqrt(42);
          case 6
            modmsg = modulate(modulator256,msg)./sqrt(170);
        end
        noisymsg = awgn(modmsg(1:msglength),SymSNR,'measured');
        profmatrix(:,(n-1)*MsgsPerScheme+k) = profgen(noisymsg);
        waitbar(((n-1)*MsgsPerScheme+k)/(NumSchemes*MsgsPerScheme),h2);
      end
```

```
end
close(h2);
```

4. KERNEL.m

```
function K = kernel(x1,x2,ktype,a,b,d)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% KERNEL.m calculates the kernel matrix given two collection of columnwise
% input vectors x1,x2 and kernel parameters.

switch ktype
   case 'gauss'
      K = exp(-Mdist(x1,x2)./a);
   case 'sig'
      K = tanh(a*x1'*x2+b);
   case 'poly'
      K = (a*x1'*x2+b).^d;
end
```

5. MDIST.m

```
function d = Mdist(x1,x2)
% Author: Nathan Geisinger, Naval Postgraduate School
% Date: 24 February 2010
% MDIST.m calculates the distance between columnwise vectors in x1 and x2.
% In other words, if each vector describes a point, d(m,n) will be the
% distance between the mth vector of x1 and the nth vector of x2. This was
% done without for loops using a trick from dist2.m by Christopher Bishop
% and Ian T Nabney.
```

```
[n,p] = size(x1);
[m,k] = size(x2);
x1sq = x1.^2;
x2sq = x2.^2;
sumx1sq = ones(k,1)*sum(x1sq);
sumx2sq = ones(p,1)*sum(x2sq);
d = sumx1sq' + sumx2sq - 2*x1'*x2;
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] P. Marchand, *Détection et Reconnaissance de Modulations Numériques a l'aide des Statistiques Cycliques d'ordre Supérieur*, Thése de l'Institut National Polytechnique de Grenoble, 1998.

[2] G. Hatzichristos, *Classification of Digital Modulation Types in Multipath Environments*, Electrical Engineer Thesis, Naval Postgraduate School, Monterey, CA 2001.

[3] A. Young, *Classification of Digital Modulation Types in Multipath Environments*, MSEE Thesis, Naval Postgraduate School, 2008.

[4] E. Like, V. Chakravarthy, P. Ratazzi, Z. Wu, "Signal Classification in Fading Channels Using Cyclic Spectral Analysis," *EURASIP Journal on Wireless Communications and Networking*, Vol. 2009, pp.879812 (14 pages), 2009.

[5] K. Ho, W. Prokopiw, Y. Chan, "Modulation Identification by the Wavelet Transform," *Conference Record of the 1995 IEEE Military Communications Conference*, Vol. 2, pp. 886–890, 1995.

[6] K. Ho, W. Prokopiw, Y. Chan, "Modulation Identification of Digital Signals by the Wavelet Transform," *Radar, Sonar and Navigation, IEE Proceedings*, Vol. 147, Issue 4, pp. 169–176, 2000.

[7] P. Prakasam and M. Madheswaran, "Digital Modulation Identification Model Using Wavelet Transform and Statistical Parameters," *Journal of Computer Systems, Networks, and Communications*, Vol. 2008, pp. 175236 (8 pages), 2008.

[8] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Blind Modulation Classification: A Concept Whose Time Has Come," 2005 IEEE Sarnoff Symposium on Advances on Wired and Wireless Communication, pp. 226–231, 2005.

[9] D. Domboulas, *Infrared Imaging Face Recognition Using Nonlinear Kernel-Based Classifiers*, Electrical Engineer Thesis, Naval Postgraduate School, Monterey, CA 2004.

[10] I. Alexandropoulos, *Uncooled Infrared Imaging Face Recognition Using Kernel-Based Feature Vector Selection*, Electrical Engineer Thesis, Naval Postgraduate School, Monterey, CA 2006.

[11] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd Edition, Prentice Hall, 2001.

[12] S. Haykin and M. Moher, *Introduction to Analog and Digital Communications*, 2nd Edition, John Wiley & Sons, 2007.

[13] J. Proakis and M. Salehi, Digital Communications, 5th Edition, McGraw-Hill, 2008.

[14] C. Martret and D. Boiteau, "Modulation Classification by means of different Orders Statistical Moments," *Proceedings of the 1997 IEEE Military Communications Conference*, Vol. 3, pp. 1387–1391, 1997.

[15] H. Mustafa and M. Doroslovacki, "Digital Modulation Recognition Using Support Vector Machine Classifier," *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 2238–2242, 2004.

[16] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th Edition, Tata McGraw-Hill, 2002.

[17] Spooner, C, "On the Utility of Sixth-Order Cyclic Cumulants for RF Signal Classification," Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, Vol. 1, pp. 890–897, 2001.

[18] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd Edition, John Wiley and Sons, Inc., 2001.

[19] B. Schölkopf, "Statistical Learning and Kernel Methods," Microsoft Research Technical Report MSR-TR-2000–23, 2000.

[20] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach," *Neural Computation*, Vol 12, pp. 2385–2404, 2000.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Monique P. Fargues
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Professor Roberto Cristi
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

5. Professor Ralph C. Robertson
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Nathan P. Geisinger
   Bella Vista, Arizona