

BOUNDARIES OF SIGNAL-TO-NOISE RATIO FOR ADAPTIVE CODE MODULATIONS

A Thesis by

Krittetash Pinyoanuntapong

Bachelor of Science in Electrical Engineering, Wichita State University, 2014

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

July 2016

© Copyright 2016 by Krittetash Pinyoanuntapong

All Rights Reserved

BOUNDARIES OF SIGNAL-TO-NOISE RATIO FOR ADAPTIVE CODE MODULATIONS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science, with a major in Electrical Engineering.

Hyuck M. Kwon, Committee Chair

Pu Wang, Committee Member

Thomas DeLillo, Committee Member

DEDICATION

To my parents, my sister, my brother,
and my dear friends

ACKNOWLEDGEMENTS

I would like to thank all the people who contributed in some way to the work described in this thesis. First and foremost, I am very grateful to my advisor, Dr. Hyuck M. Kwon, for giving me the opportunity to be part of this research and for providing the impetus, means, and support that enabled me to work on the project. I deeply appreciate his concern, appreciation, and encouragement, which enabled me to graduate. Additionally, I thank Dr. Thomas K. DeLillo and Dr. Pu Wang, who graciously agreed to serve on my committee and for spending their time at my thesis presentation. Thank you and the best of luck in your future endeavors. My gratitude also goes to Wichita State University for providing part of me tuition fee.

Finally, I thank my parents, Dr. Sirima and Dr. Boonchird, for their constant love and support, which have allowed me to realize my own potential, and for always encouraging me to work hard and follow my dreams. Their support over the years has been the greatest gift anyone has ever given me. They also taught me the value of hard work and an education. Without them, I may never have gotten to where I am today.

ABSTRACT

Adaptive code modulation (ACM) has been studied for last two decades to enhance the bandwidth and power efficiency in a bandwidth-constrained and power-constrained communication system environment. Also, the existing digital video broadcasting-return channel via satellite (DVB-RCS) and the second-generation DVB satellite (DVB-S2) including the future protected spectrum waveform have employed ACM. This paper studies how to decide the boundaries of a signal-to-noise ratio (SNR) for the ACM mode selection for a given set of parameters and system model. Also, this paper proposes an efficient and practical algorithm to search the proper ACM mode that meets a target bit error rate (BER) and achieves the maximum data rate. The proposed algorithm is found to be effective under jamming and interference, even if channel and jamming state information are unavailable. The results in this paper can be applicable to future satellite and mobile communication system designs against jamming and interference environments.

TABLE OF CONTENTS

Chapter		Page
1.	INTRODUCTION	1
2.	SYSTEM MODEL	2
	2.1 Adaptive Code Modulation Model	2
	2.2 Channel Model	3
	2.3 Parameters	5
3.	SELECTING ADAPTIVE CODE MODULATION PROCESS	6
	3.1 Case A: SNR Boundaries under AWGN	6
	3.2 Case B: SNR Boundaries under Jamming with JSI	9
	3.3 Case C: SNR Boundaries under Jamming without JSI	11
4.	NUMERICAL RESULTS	16
5.	CONCLUSIONS	25
	REFERENCES	26
	APPENDICES	29
	A. Main MATLAB Code Used in Calculations	30
	B. MATLAB Code Used in Computing Average Spectrum Efficiency, Average Throughput and Average of Probability Error.	35
	C. MATLAB Code Used in Optimizing SNR Value for Target BER	38

LIST OF TABLES

Table		Page
1.	Generator Matrices for Convolutional Code of Constraint Length 7 with Code Rates 1/4, 1/3, 1/2, and 2/3	3
2.	SNR Boundaries for Ten ACM Modes under AWGN or AWGN plus Jamming when $\frac{E_b}{N_j} = 10$ dB and $\beta = 1$	9
3.	Selected ACM Modes for Case C Using AWGN SNR Boundaries under AWGN plus Jamming when $\frac{E_b}{N_j} = 10$ dB and $\beta = 1$	15
4.	SNR Boundaries for Twenty ACM Modes under Applied FEC LDPC Coding under AWGN or AWGN plus Jamming when $\frac{E_b}{N_j} = 10$ dB and $\beta = 1$	22

LIST OF FIGURES

Figure	Page
1. System model	2
2. Bit error rate $P_b(\gamma)$ versus $\gamma = E_b / N_0$ in dB under AWGN for 10 ACM modes plotted, assuming no jamming and jamming when $E_b / N_J = 10$ dB	8
3. Flow chart of proposed algorithm for ACM mode selection when JSI is unavailable	12
4. Instantaneous spectral efficiency versus $\gamma = E_b / N_0$ in dB for no jamming (Case A), jamming with JSI (Case B), and jamming without JSI (Case C) when no fading is considered	16
5. Average bandwidth efficiency comparisons of Cases A, B, and C taken over Rician fading of Rician factor $K=5$	18
6. Average BER comparisons of Cases A, B, and C taken over Rician fading of Rician factor $K=5$	19
7. Comparison between average SNR with no jamming and average SNR with jamming	20
8. Plots of bit error rate $P_b(\gamma)$ versus $\gamma = E_b / N_0$ in dB for 20 ACM modes applied with FEC LDPC coding under AWGN and assuming no jamming	21
9. Instantaneous spectral efficiency versus $\gamma = E_b / N_0$ in dB applied with FEC LDPC from Table 3 for no jamming (Case A), jamming with JSI (Case B), and jamming without JSI (Case C)	24
10. Average bandwidth efficiency comparisons of Cases A, B, and C applied FEC LDPC from Table 3; average is taken over Rician fading of Rician factor $K=5$	24

LIST OF ABBREVIATIONS

ACM	Adaptive Code Modulation
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
DTN	Delay-Tolerant Network
DVB-RCS	Digital Video Broadcasting-Return Channel via Satellite
DVB-S2	Digital Video Broadcasting Satellite Communication-Second Generation
FEC	Forward Error Correction
JSI	Jamming State Information
LDPC	Low-Density Parity Check
LOS	Line-of-Sight
MPSK	M-ary Phase Shift Keying
PBNJ	Partial-Band Noise Jamming
PBTJ	Partial-Band Tone Jamming
PSK	Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
SNR	Signal-to-Noise Jamming

CHAPTER 1

INTRODUCTION

Adaptive code modulation (ACM) has been studied for last two decades to enhance the bandwidth and power efficiency in a bandwidth- and power-constrained communication system environment [1]-[3]. For example, Ellis and Pursley [1] examined the integration of adaptive modulation and channel coding protocols with fountain coding in a packet radio system. Also, the existing digital video broadcasting-return channel via satellite (DVB-RCS) and the second-generation DVB satellite (DVB-S2) including the future protected spectrum waveform have employed an ACM [4]-[9].

This research studies how to determine the ACM boundaries of the signal-to-noise ratio (SNR) for a given set of parameters and the system model. The results here will be useful for the designing future satellite communication systems against jamming and interference environments.

Chapter 2 describes the system model, and Chapter 3 presents process in selecting the ACM mode out of ten modes, assuming that jamming state information is available and is unavailable at the receiver. Chapter 4 compares and discusses the numerical results for the three cases, and Chapter 5 concludes the thesis.

CHAPTER 2

SYSTEM MODEL

The system model is shown in Figure 1.

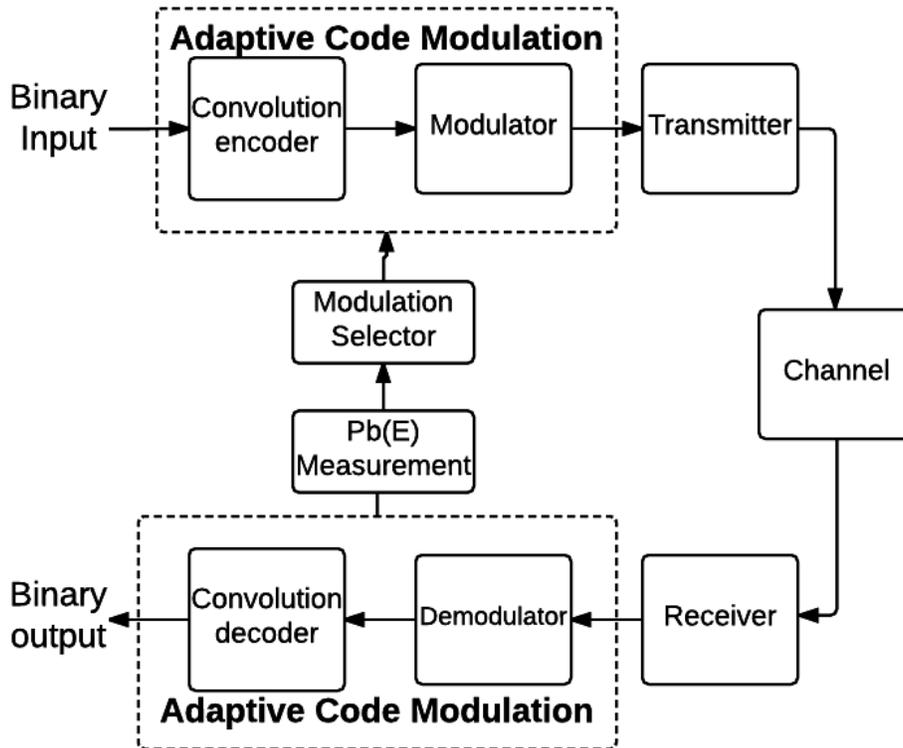


Figure 1. System model.

2.1 Adaptive Code Modulation Model

Two sets of adaptive code modulations are considered here. The first set is comprised of ten modes. These are combinations of two M-ary phase shift keying (MPSK) modulations (i.e., quadrature PSK [QPSK] and 8PSK) and a convolutional forward error correction (FEC) code with five different code rates (i.e., 1/4, 1/3, 1/2, 2/3, and 1). The connection generator matrices for a convolutional code of constraint length 7 for these five code rates are listed in Table 1 [13]. The code rates and the constraint length are obtained from the DVB-S2 standard [9]. The ten ACM modes will be discussed in Chapter 3.

TABLE 1

GENERATOR MATRICES FOR CONVOLUTIONAL CODE OF CONSTRAINT LENGTH 7 WITH CODE RATES 1/4, 1/3, 1/2, 2/3

Code Rate	Generator Matrices (Constraint Length 7)	d_{free}	Coding Gain $G_c \approx d_{free} R_c$ in dB
1	0	No Coding	0
1/4	[163 147 135 135]	22	7.4
1/3	[171 165 133]	15	7
1/2	[155 177]	8	6
2/3	$\begin{bmatrix} 5 & 23 & 27 \\ 53 & 51 & 65 \end{bmatrix}$	5	5.22

The second ACM set is comprised of 20 modes. These are combinations of two modulations—QPSK and 8PSK—and a low-density parity check (LDPC) FEC of 12 code rates (i.e., 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10, and 1). For QPSK, all 12 code rates are used, whereas for 8PSK, only eight code rates are used: 3/5, 2/3, 3/4, 4/5, 5/6, 8/9, 9/10, and 1. This is because the bandwidth efficiency of 8PSK with a low code rate (1/4, 1/3, 2/5, 1/2) is lower than that of QPSK with a 9/10 rate. These code rates are obtained from DVB-S2 standard [9, p. 13]. The twenty ACM modes will be discussed in Chapter 4.

2.2 Channel Model

A Rician fading channel in addition to additive white Gaussian noise channel (AWGN) is considered instead of Rayleigh fading because the line-of-sight (LOS) component is typically available in a satellite communication system. This paper assumes partial band tone jamming (PBTJ) or partial band noise jamming (PBNJ). Interference can be added with no difficulty because this paper considers an AWGN type of interference. Most literature also considers the AWGN type of interference.

The proposed algorithm is effective for any other type of interference and jamming because the proposed algorithm uses the bit error rate (BER) criteria to select the ACM mode and does not have the jamming and interference state information. Therefore, this paper assumes a PBNJ without loss of generality. The probability of being jammed is between 0 and 1. The probability density function of the instantaneous received symbol SNR in a Rician fading channel is written as

$$p_\gamma(\gamma) = \frac{K+1}{\bar{\gamma}} \exp\left[-K - \frac{(K+1)\gamma}{\bar{\gamma}}\right] I_0\left[2\sqrt{\frac{K(K+1)\gamma}{\bar{\gamma}}}\right] \quad (1)$$

where $\bar{\gamma} = E[\gamma]$ is the expectation of instantaneous SNR, which is equal to $\gamma = |h|^2 E_s/N_0$ or $\gamma = |h|^2 E_s/(N_0 + \beta N_j)$ under Rician fading, depending on the jamming state, K is the Rician fading factor, and h is the Rician fading coefficient whose power is written from the work of Proakis [15, eqn. 2.3-58] as

$$E[|h|^2] = \sigma \sqrt{\frac{\pi}{2}} e^{-k/2} \left[(1+K)I_0\left(\frac{K}{2}\right) + KI_1\left(\frac{K}{2}\right) \right] \quad (2)$$

and $I_0(x)$ and $I_1(x)$ are the zero-th and first order of the modified Bessel function, respectively, assuming there is no loss of generality ($\sigma^2=1$). Also, $E_s/N_0 = (\log_2 M)E_b/N_0$ and $E_s/N_j = (\log_2 M)E_b/N_j$ are the symbol energy-to-noise power spectral density ratio and symbol energy-to-jamming plus noise power spectral density ratio, respectively, where E_b is the bit energy, and M is 4 or 8.

The probability that the selected ACM mode information is fed back to the transmitter incorrectly is negligible. This is because the amount of feedback information is very small,

compared to the data amount, and sufficient power can be allocated for the ACM mode information transmission. The transmitter applies the received ACM mode information for the next frame interval. Each frame (or an update time interval for ACM) consists of sufficient number of data bits to estimate BER (or packet error probability) reliably. For example, 100,000 bits per frame is sufficient for a target BER of 10^{-3} . Assume that the ACM mode selected by the receiver is delivered perfectly for the transmitter to use during the next frame interval. Each frame consists of a large number of symbols (i.e. 300). The average of the received symbol SNR $\bar{\gamma}$ can be written as

$$\bar{\gamma} = \begin{cases} E[|h|^2] \frac{E_s}{N_0} & \text{no jamming} \\ E[|h|^2] \left(\frac{\frac{E_s}{\beta N_j} \frac{E_s}{N_0}}{\frac{E_s}{\beta N_j} + \frac{E_s}{N_0}} \right) & \text{jamming w/ prob. } \beta \end{cases} \quad (3)$$

2.3 Parameters

Table 1 lists ten modes of the adaptive code modulations, which are combinations of QPSK and 8PSK and an FEC code with five different rates, i.e., convolutional codes of 1/4, 1/3, 1/2, 2/3, and 1 (no coding). These code rates are obtained from the DVB-S2 standard.

The instantaneous bit SNR range is from 0 to 40 dB. The calculation for the jamming environment in this thesis assumes that the transmitted signal is jammed with $E_b / N_j = 10$ dB and jamming probability $\beta = 1$. ACM BER under the Rician fading of factor K is given as 5.

The MATLAB© software (version 2016) was used for this calculation. The related MATLAB codes are listed in the appendices.

CHAPTER 3

SELECTING ADAPTIVE CODE MODULATION METHOD

This chapter discusses the method in selecting an ACM mode out of ten possible combinations of convolutional FEC and QPSK/8PSK modulations (or twenty combinations of LDPC FEC and QPSK/8PSK modulation) when the instantaneous SNR is time-varying due to the presence of fading and jamming/interference. Case A finds the lower and upper SNR boundaries, i.e., SNR region for each ACM mode to be used, assuming an AWGN channel of E_s/N_0 . Case B assumes that jamming state information (JSI) (e.g., E_s/N_j and jamming probability β) is available at the receiver and finds the SNR region for each ACM mode using the known JSI. Case C assumes that JSI is unknown at the receiver and presents an efficient algorithm for how to select a proper ACM mode based on BER or symbol error rate test using the SNR regions obtained for Case A.

3.1 Case A: SNR Boundaries under AWGN

SNR boundaries will be determined to guarantee that the instantaneous BER under fading is less than or equal to a maximum allowable BER. For example, the maximum BER is set to $P_{b,\max} = 10^{-3}$ for combinations of convolutional FEC and QPSK/8PSK, and $P_{b,\max} = 10^{-5}$ for combinations of LDPC FEC and QPSK/8PSK modulations in this paper, and hence the instantaneous BER $P_b(\gamma) \leq P_{b,\max} = 10^{-3}$ or 10^{-5} . If MPSK and a convolutional error correction code are used, then an explicit union upper bound of the BER after the convolution decoding can be written from the work of Goldsmith [3, eqn. 8.71] and Moon [14, pp. 503-504] as

$$P_b(\gamma) \leq \sum_{d=d_{\text{free}}}^{\infty} a(d) P_d(\gamma) \quad (4)$$

for a given instantaneous symbol SNR $\gamma = (\log_2 M)\gamma_b$, where γ_b is the instantaneous bit SNR, d is the Hamming distance between the correct convolutional code word trellis path and an incorrect trellis path, $a(d)$ is the number of incorrect paths whose distance from the correct path is $d \geq d_{free}$, which is uncorrectable, and $P_d(\gamma)$ is the pairwise convolutional error correction code word error probability due to decoding the received word incorrectly from the correct code word trellis path to an incorrect code word trellis path of distance d . Here, $P_d(\gamma)$ can be written as

$$\begin{aligned} P_d(\gamma) &\approx Q\left(\sqrt{2dR_c\gamma} \sin \pi/M\right) \\ &= Q\left(\sqrt{2dR_c(\log_2 M)\gamma_b} \sin \pi/M\right) \end{aligned} \quad (5)$$

where $Q(\alpha) = \int_{\alpha}^{\infty} 1/\sqrt{2\pi} \exp(-t^2/2) dt$ is the tail probability of a normalized Gaussian random variable larger than or equal to α , R_c is the code rate, and it is assumed that whenever a code symbol error occurs, it is demodulated into a wrong symbol with the minimum distance in the MPSK signal constellation.

Figure 2 shows the BER rate $P_b(\gamma)$ versus $\gamma = E_b/N_0$ for the ten ACM modes under AWGN, using equations (4) and (5). The crossing points at $P_b(\gamma) = 10^{-3}$ marked by red triangles are the lower boundaries of γ_i for ACM mode i under the case of AWGN when there is no jamming.

Table 2 lists the SNR boundaries for the no-jamming case using the results in Figure 2. Note that the bandwidth efficiencies under 8PSK with code rates 1/4 and 1/3 are lower than that of QPSK with code rate 2/3, but they require a higher SNR in order to maintain 10^{-3} BER. Thus, the 8PSK modulation with code rates 1/4 and 1/3 are neglected.

Under a fading environment, the SNR γ will vary. If γ is known at the receiver, then the proper ACM mode can be selected using SNR boundaries from Table 2.

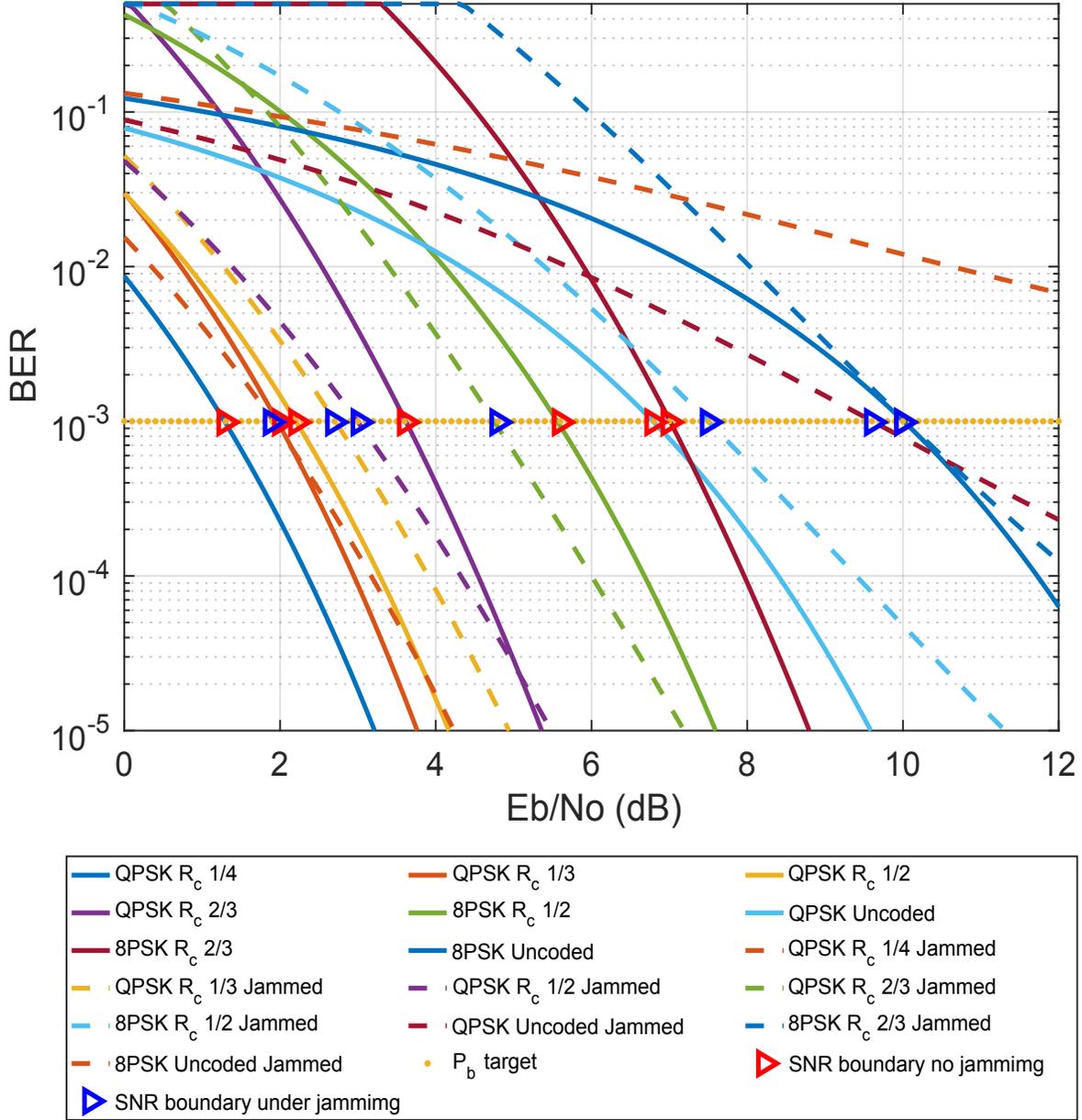


Figure 2. Bit error rate $P_b(\gamma)$ versus $\gamma = E_b / N_0$ in dB under AWGN for 10 ACM modes plotted, assuming no jamming and jamming with $E_b / N_j = 10$ dB. Crossing points at $P_b(\gamma) = 10^{-3}$ (i.e., lower boundaries of γ_i for ACM mode i) are marked by triangles in red and blue for no jamming and jamming cases, respectively.

TABLE 2

SNR BOUNDARIES FOR TEN ACM MODES UNDER AWGN OR AWGN
PLUS JAMMING WHEN $\frac{E_b}{N_j} = 10$ dB AND $\beta = 1$

ACM Mode Index i	ACM Mode	Bandwidth Efficiency $\frac{R(\gamma_i)}{B} = (\log_2 M) R_c$ (bits / (s · Hz))	SNR Lower Bound γ_i in dB for ACM Mode i , i.e., $P_b(\gamma_i) = 10^{-3}$ $\gamma_i \leq \gamma < \gamma_{i+1}$	
			No Jamming	Under Jamming
0	No Transmission	0	–	–
1	QPSK + R_c 1/4	0.5	$\gamma_{b,1} = 1.3$	$\gamma_{b,1,J} = 1.9$
2	QPSK + R_c 1/3	0.66	$\gamma_{b,2} = 2.0$	$\gamma_{b,2,J} = 2.7$
3	QPSK + R_c 1/2	1	$\gamma_{b,3} = 2.2$	$\gamma_{b,3,J} = 3.0$
4	QPSK + R_c 2/3	1.33	$\gamma_{b,4} = 3.6$	$\gamma_{b,4,J} = 4.8$
5	8PSK + R_c 1/4	0.75	$\gamma_{b,5} = 4.6$	$\gamma_{b,5,J} = 6.1$
6	8PSK + R_c 1/3	1	$\gamma_{b,6} = 5.3$	$\gamma_{b,6,J} = 7.2$
7	8PSK + R_c 1/2	1.5	$\gamma_{b,7} = 5.6$	$\gamma_{b,7,J} = 7.5$
8	QPSK + R_c 1 (no coding)	2	$\gamma_{b,8} = 6.8$	$\gamma_{b,8,J} = 9.6$
9	8PSK + R_c 2/3	2	$\gamma_{b,9} = 7.0$	$\gamma_{b,9,J} = 10.0$
10	8PSK + R_c 1 (no coding)	3	$\gamma_{b,10} = 10.0$	$\gamma_{b,10,J} = \infty$ cannot be achieved

3.2 Case B: SNR Boundaries under Jamming with JSI

When the transmitted signal is jammed with $E_b / N_j = 10$ dB and jamming probability $\beta = 1$, for example, the required crossing boundary point $\gamma_{b,i,J}$ for the ACM mode to achieve the same $P_b(\gamma) = 10^{-3}$ is shifted to a higher SNR than $\gamma_{b,i}$ because the effective symbol energy-to-jamming-plus-noise ratio (SJNR) is $E_s / (N_0 + \beta N_j)$ under AWGN and jamming (but no fading).

This SJNR can be rewritten as

$$SJNR = \frac{E_s}{\beta N_J} \frac{E_s}{N_0} \left/ \left(\frac{E_s}{\beta N_J} + \frac{E_s}{N_0} \right) \right. \quad (6)$$

It is interesting to know the shifting amount $(10\log_{10} \gamma_{b,i,J}) - (10\log_{10} \gamma_{b,i})$ in dB in order for the transmitter to use the same ACM mode under jamming with known JSI. This amount indicates how much the transmitter needs to increase its transmitter power P_T or SNR E_s/N_0 in equation (6) because $E_s = P_T T$ and the one-side AWGN power spectral density level N_0 and symbol time T are constant. If the transmitter cannot change its transmit power, then this shifting amount indicates selecting a proper ACM mode to achieve the same target BER. This paper assumes a constant transmit power, i.e., no power adaptation but rather a rate adaptation because typical satellite systems use constant power.

As can be seen previously in Figure 2, the crossing points at $P_b(\gamma_{b,i,J}) = 10^{-3}$, which are marked by triangles in blue color, are the lower SNR boundaries for ACM mode i under jamming. Note that the horizontal axis in Figure 2 is not SJNR but SNR $\gamma_b = E_b/N_0$. The symbol SNR boundary $\gamma_{i,J} = E_s/N_0$ under jamming can be expressed in terms of known JSI $(E_s/N_J, \beta)$ and the SNR boundary $\gamma_i = E_s/N_0$ obtained under no jamming from $P_b(\gamma_i) = P_b(SJNR(\gamma_{i,J})) = 10^{-3}$ as

$$\gamma_{i,J} = \frac{E_s}{\beta N_J} \gamma_i \left/ \left(\frac{E_s}{\beta N_J} - \gamma_i \right) \right. \quad (7)$$

This is obtained from equation (6) by setting

$$\gamma_i = SJNR(\gamma_{i,J}) = \frac{E_s}{\beta N_J} \gamma_{i,J} \left/ \left(\frac{E_s}{\beta N_J} + \gamma_{i,J} \right) \right. \quad (8)$$

where $\gamma_i = (E_s/N_0)$ is $\log_2 M$ multiplied by the crossing points $\gamma_{b,i}$ at $P_b(\gamma) = 10^{-3}$ under only AWGN but no jamming. This is because both BERs are approximately a function of the tail probability $Q(\alpha)$ for both Cases A and B.

Again, the SJNR γ_J will vary under a fading environment. If γ_J is known at the receiver, i.e., channel state information (E_b/N_0 , E_b/N_J , and β) is known, then the proper ACM mode can be selected using Table 2 for the jamming case because Table 2 lists the SNR boundaries $\gamma_{b,i,J}$ at the fifth column for the jamming case, using the blue marks shown previously in Figure 2. Note also that 8PSK with no coding cannot be used because $P_b(\gamma) = 10^{-3}$ cannot be achieved, even at $\gamma_{b,10,J} = \infty$. Note that the gap between $\gamma_{i,J}$ and γ_i in dB, i.e., $(10\log_{10} \gamma_{b,i,J}) - (10\log_{10} \gamma_{b,i})$ in dB, by using equation (7), matches closely with the gap between $\gamma_{b,i,J}$ and $\gamma_{b,i}$ in dB in Table 2. Furthermore, from equation (7), a simple theorem can be stated.

Theorem 1: The ACM mode i can be used only if $E_s/\beta N_J > \gamma_i$. Else, the ACM mode i cannot be employed.

Proof of Theorem 1: This is because the denominator in equation (7) becomes negative. For example, ACM mode $i = 10$ in Table 2 cannot be employed when $E_b/N_J = 10$ dB, and $\beta = 1$.

3.3 Case C: SNR Boundaries under Jamming without JSI

If the channel is jammed but JSI is unknown, then a proper ACM mode cannot be selected directly from the fifth column of Table 2 because γ_J is unavailable, i.e., E_b/N_J and β are unknown. Therefore, we propose to use the boundaries γ_i obtained for Case A (unjammed case) and employ the BER criteria of whether a trial ACM mode satisfies or not. When the trial ACM mode satisfies $P_b(\gamma_J) \leq 10^{-3}$, the required BER during the current test interval (which is less

than the channel coherence time that can be a frame of multiple symbols), then the trial ACM mode is moved to the next ACM mode of higher bandwidth efficiency for the next test transmission. If the chosen ACM mode still satisfies the BER requirement, then the next ACM mode is moved to again. If the new trial ACM mode does not satisfy the BER requirement, then the ACM mode is moved down, and so on.

Figure 3 shows the flow chart of the proposed algorithm in detail. Since only the boundary γ_i obtained for Case A (unjammed case) is available, the algorithm will find which ACM mode is appropriate to use boundary γ_i for Case C with unknown received SJNR γ_j by using the simple BER test.

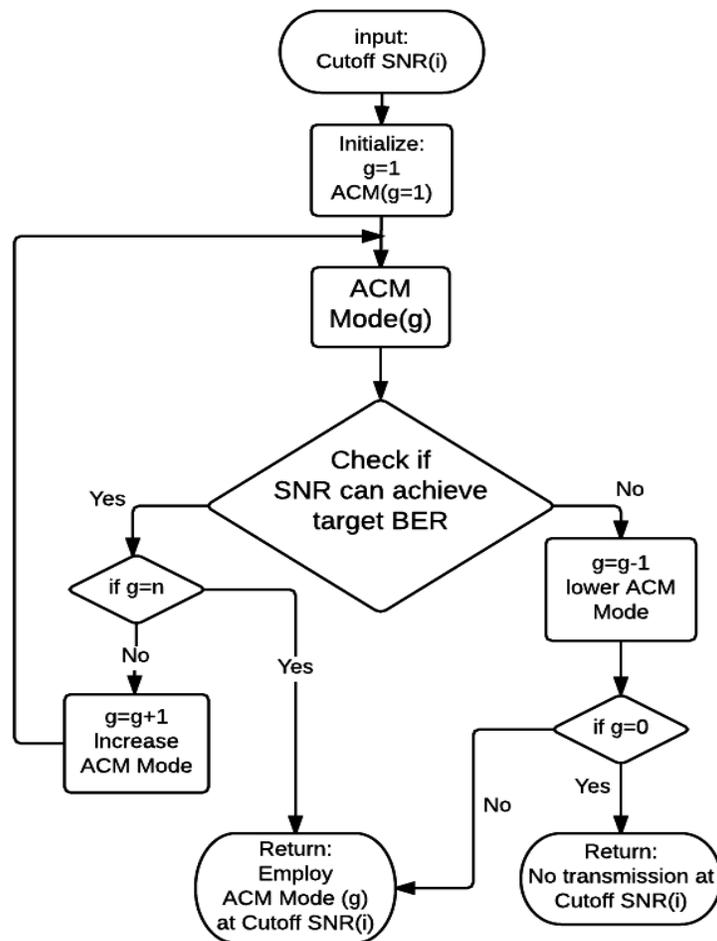


Figure 3. Flow chart of proposed ACM mode selection when JSI is unavailable.

As can be seen in Figure 3, variable i denotes the index of the trial SNR boundary (or ACM mode index), g is the chosen ACM mode index at the end of each loop for a given boundary γ_i test, and n is the number of possible modes, e.g., 8 in Table 2. Therefore, the maximum number of possible trial frames is 8 in this case. The following explanation is presented to demonstrate the process of the proposed algorithm using the example in Table 2. The flow chart starts at the first input cutoff SNR, $\gamma_{b,1} = 1.3$ dB. The receiver sets the index to $g = 1$ and initializes it using the first ACM mode, which is QPSK modulation with a code rate 1/4, even under a jamming environment, because the receiver has no JSI. However, at this modulation, the receiver cannot achieve 10^{-3} BER because of the jamming signal presence, e.g., $E_b/N_j = 10$ dB. Refer to the broken red line in Figure 2 and column 5 of Table 2. This needs to be at least 1.9 dB to achieve the target BER 10^{-3} . The receiver lowers the trial ACM index (i.e., $g = g - 1$), which becomes 0 ACM mode, i.e., no transmission for $i = 1$ boundary, i.e., γ_1 . Hence, the algorithm outputs no transmission ACM mode for region 1, i.e., $\gamma_1 \leq \gamma_j < \gamma_2$.

Now, the receiver will make a decision for the next AWGN boundary $\gamma_2 = 2.0$ dB. Then the receiver checks to see if index g has reached the last ACM mode before it increases the trial ACM index (i.e., $g = g + 1$). Again, the receiver checks the cutoff SNR γ_2 with the QPSK modulation with code rate 1/3. However, it detects that the BER is higher than 10^{-3} at $\gamma_2 = 2.0$ dB under the jamming scenario in Table 2. Refer to column 5 of Table 2 and Figure 2. At least 2.7 dB is needed to achieve the target BER for the jamming signal presence with $E_b/N_j = 10$ dB. Then the receiver lowers the ACM index to the previous mode, which is, again, the QPSK modulation with code rate 1/4. Thus, the modulation that will be used for region 2, i.e., $\gamma_2 \leq \gamma_j < \gamma_3$, is QPSK modulation with code rate 1/4.

This process is continued. At $\gamma_6 = 6.8$ dB, the receiver starts making decisions from ACM mode 1 and will continue increasing the modulation while the BER 10^{-3} can be achieved. The decision process continues until the ACM mode with 8PSK modulation and code rate 1/2. Then it detects that the BER is higher than 10^{-3} at $\gamma_6 = 6.8$ dB. Refer to column 5 of Table 2 and Figure 2. At least 7.5 dB is needed to achieve the target BER. The receiver then lowers the ACM mode to the less-efficient-bandwidth ACM mode. This process continues until the last boundary, e.g., γ_8 , is decided. Therefore, the proposed algorithm can indicate what ACM mode should be used, even if the jamming state information is not available.

The decision based on boundaries is made for calculating the bandwidth efficiency and data rate. In practice, the algorithm presented in Figure 3 can accept an instantaneous γ and select the proper ACM modulation based on the given γ to maximize the bandwidth efficiency and data rate.

Table 3 lists the selected ACM mode based on the SNR $\gamma_{b,i,J}$ boundaries from Case A. The ACM modes listed in column 2 of Table 3 are selected based on the proposed algorithm to achieve the target BER 10^{-3} under presence of the jamming signal with $E_b/N_j = 10$ dB and jamming probability $\beta = 1$.

The average bandwidth efficiency is computed as

$$\frac{\bar{R}(\gamma)}{B} = \sum_{i=1}^{N_{\text{mode}}} \frac{R(\gamma_i)}{B} \Pr(\text{mode } i \text{ is selected}) \quad (9)$$

$$\Pr(\text{mode } i \text{ is selected}) = \int_{\gamma_i}^{\gamma_{i+1}} p_{\gamma}(\gamma) d\gamma, \quad (10)$$

where N_{mode} is the number of possible ACM modes, B is the channel bandwidth equal to the inverse of the symbol interval $1/T$, and $p_{\gamma}(\gamma)$ is the probability density function of Rician fading in (1) with Rician factor $K = 5$ and the average received SNR $\bar{\gamma}$.

TABLE 3

SELECTED ACM MODES FOR CASE C USING AWGN SNR BOUNDARIES UNDER AWGN PLUS JAMMING WHEN $\frac{E_b}{N_j} = 10$ dB AND $\beta = 1$

ACM Mode Index i	ACM Mode	Bandwidth Efficiency $\frac{R(\gamma_i)}{B} = (\log_2 M) R_c$ (bits / (s · Hz))	SNR Lower Bound γ_i in dB for ACM Mode i , i.e., $P_b(\gamma_i) = 10^{-3}$ $\gamma_i \leq \gamma < \gamma_{i+1}$
			No Jamming
0	No Transmission	$\frac{R(\gamma_0)}{B} = 0$	–
1	No Transmission	$\frac{R(\gamma_1)}{B} = 0$	$\gamma_{b,1} = 1.3$
2	QPSK + R_c 1/4	$\frac{R(\gamma_2)}{B} = \log_2 4 * \frac{1}{4} = 0.5$	$\gamma_{b,2} = 2.0$
3	QPSK + R_c 1/4	$\frac{R(\gamma_3)}{B} = \log_2 4 * \frac{1}{4} = 0.5$	$\gamma_{b,3} = 2.2$
4	QPSK + R_c 1/2	$\frac{R(\gamma_4)}{B} = \log_2 4 * \frac{1}{2} = 1$	$\gamma_{b,4} = 3.6$
5	QPSK + R_c 2/3	$\frac{R(\gamma_5)}{B} = \log_2 4 * \frac{2}{3} = 1.33$	$\gamma_{b,7} = 5.6$
6	QPSK + R_c 2/3	$\frac{R(\gamma_6)}{B} = \log_2 4 * \frac{2}{3} = 1.33$	$\gamma_{b,8} = 6.8$
7	QPSK + R_c 2/3	$\frac{R(\gamma_7)}{B} = \log_2 4 * \frac{2}{3} = 1.33$	$\gamma_{b,9} = 7.0$
8	8PSK + R_c 2/3	$\frac{R(\gamma_8)}{B} = \log_2 8 * \frac{2}{3} = 2$	$\gamma_{b,10} = 10.0$

The average BER is also taken over the Rician fading as

$$\bar{P}_b = \sum_{i=1}^{N_{\text{mode}}} P_b(\gamma | \text{mode } i) \Pr(\text{mode } i \text{ is selected}) \quad (11)$$

where $P_b(\gamma | \text{mode } i)$ is computed using equation (4) for $\gamma_i \leq \gamma < \gamma_{i+1}$, $i = 1, \dots, N_{\text{mode}}$.

CHAPTER 4

NUMERICAL RESULTS

This chapter discusses the numerical results of the ACM bandwidth efficiency and ACM BER under the Rician fading of factor $K = 5$. Both the no-jamming case and the jamming case with $E_b / N_j = 10$ dB and jamming probability $\beta = 1$ are considered.

Figure 4 shows the instantaneous spectral efficiency versus SNR $\gamma = E_b / N_0$ in dB for the no-jamming case (Case A), jamming with JSI case (Case B), and jamming without JSI case (Case C). No fading is assumed. Observe that there are six noticeable steps and one unnoticeable step at γ equal to 7 dB for Case A. This is because ACM modes 8 and 9 produce the same bandwidth efficiency, which is consistent with the results in Table II. Therefore, it would be better to not employ mode 9 if the goal is to achieve a higher bandwidth efficiency.

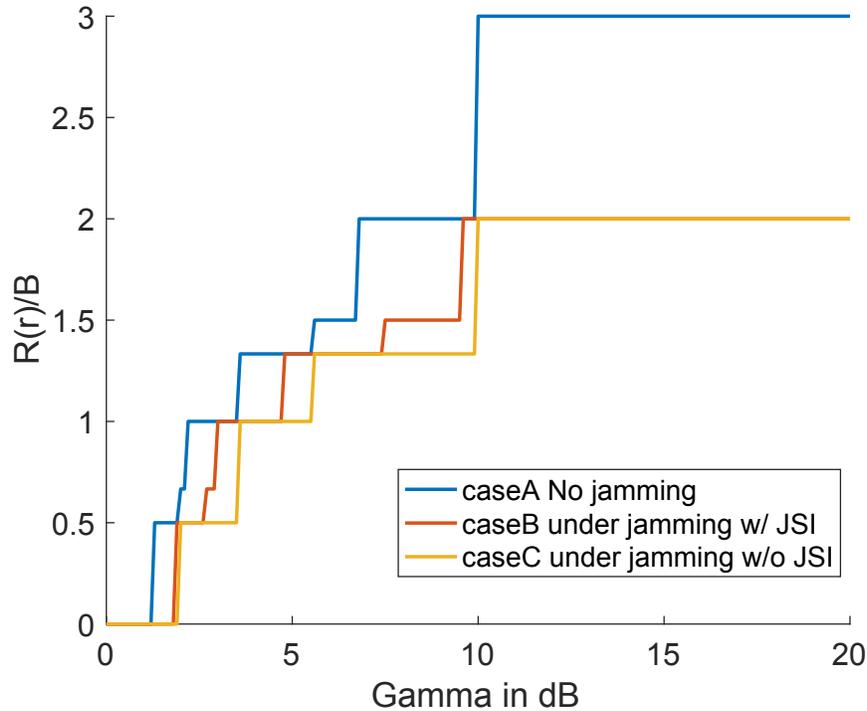


Figure 4. Instantaneous spectral efficiency versus $\gamma = E_b / N_0$ in dB for no jamming (Case A), jamming with JSI (Case B), and jamming without JSI (Case C). No fading is considered.

However, as shown previously in Figure 2, mode 9 can achieve a lower BER than mode 8 at a higher SNR. For example, if the target BER is 10^{-4} , then it would be better to employ mode 9. Two ACM modes 5 and 6 are not employed as stated earlier because they show worse bandwidth efficiency and require a higher SNR than mode 4. Also, observe that the maximum instantaneous bandwidth efficiency is 3 bits/s/Hz, which can be achieved with 8PSK of no coding when the instantaneous SNR is greater than 10 dB.

Observe also in Figure 4 that Cases B and C under jamming show similar performance, but the maximum reachable instantaneous bandwidth efficiency is only 2 bits/s/Hz. This is because mode 10 shows a BER floor higher than 10^{-3} in Figure 1, even for a very strong $E_s/N_0 = 10$, and mode 10 cannot be used. The 2 bits/s/Hz can be achieved by employing QPSK with no coding.

Observe also in Figure 4 that Case C (without JSI) shows a performance close to Case B (with JSI), even if JSI is not available. This is surprising and encouraging. Note also that the proposed algorithm does not need channel estimation, which simplifies the receiver processing significantly. The time delay caused by searching for a proper ACM mode with BER criteria may be acceptable in many applications, e.g., a disruption/delay-tolerant network (DTN) [10]-[12]. It will take less than an eight-frame interval delay to update the ACM mode if there are eight possible ACM modes in total, because each frame size can be sufficient to generate a BER test, e.g., 100,000 bits per frame, and eight frames take only 80 ms for a data rate of 10 Mbps. Therefore, the proposed algorithm shown in Figure 3 may be usable for future MILSATCOM waveform designs.

Figure 5 shows the average bandwidth efficiency versus the average received SNR $\bar{\gamma}$ and presents the results of Cases A, B, and C. Observe again that Case C (without JSI) shows a

performance close to Case B (with JSI), even if JSI is not available, and Case C shows almost the same performance as Case B when $\bar{\gamma} \geq 20$ dB for a given scenario of jamming in this research.

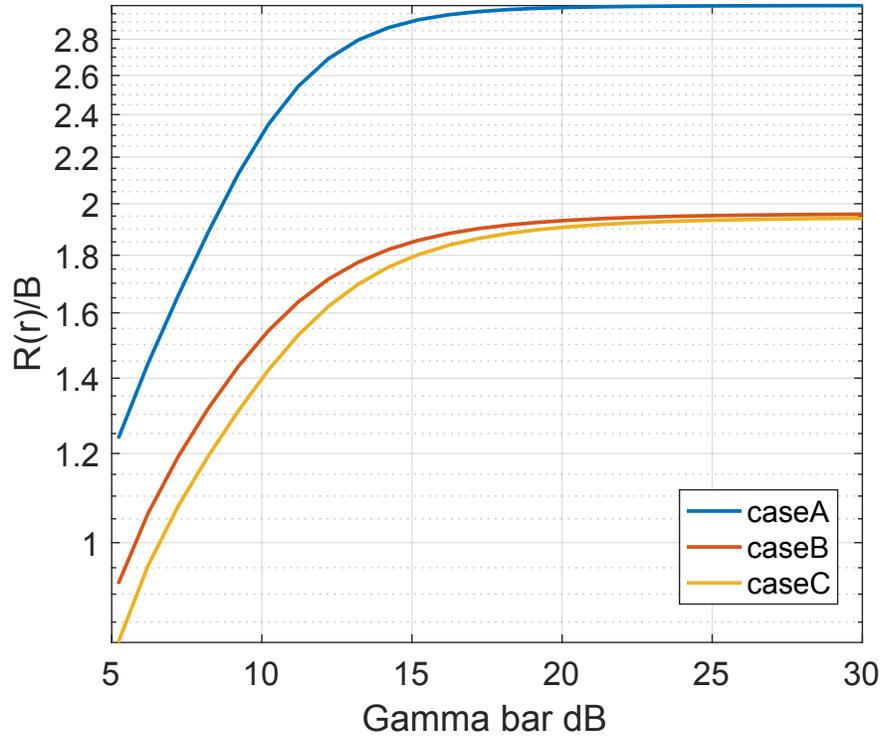


Figure 5. Average bandwidth efficiency comparisons of Cases A, B, and C. Average is taken over Rician fading of Rician factor $K = 5$.

Figure 6 shows the average BER versus $\bar{\gamma}$ and compares the results for Cases A, B, and C. Again, the average is taken over the Rician fading. Observe in Figure 6 that Case A shows the smallest (best) average BER when $\bar{\gamma}$ is sufficiently higher than 20 dB because Cases B and C show the BER floor due to jamming for a high SNR region. Observe that when the average SNR is lower than 20 dB, Case C shows a smaller BER than Case B and even Case A. This is because in Case A, all eight of the ACM modes are considered, while in Case B, the 8PSK modulation with no coding is deleted since the BER target cannot be achieved under jamming. And in Case C (refer to Table 4), only four of the ACM modes are used: QPSK with code rates of 1/4, 1/2, and 2/3, and 8PSK with a code rate of 2/3. In addition, the first region in Case C is considered as

a no-transmission region because the lower SNR boundary cannot achieve the target BER. Thus, the average BER of Case B is higher than that of Case C, but Case B has higher bandwidth efficiency than Case C. And since the instantaneous BER under fading is set to be less than or equal to a maximum allowable BER 10^{-3} , the average BER of Case B is acceptable. This paper focuses more on bandwidth efficiency criteria than BER.

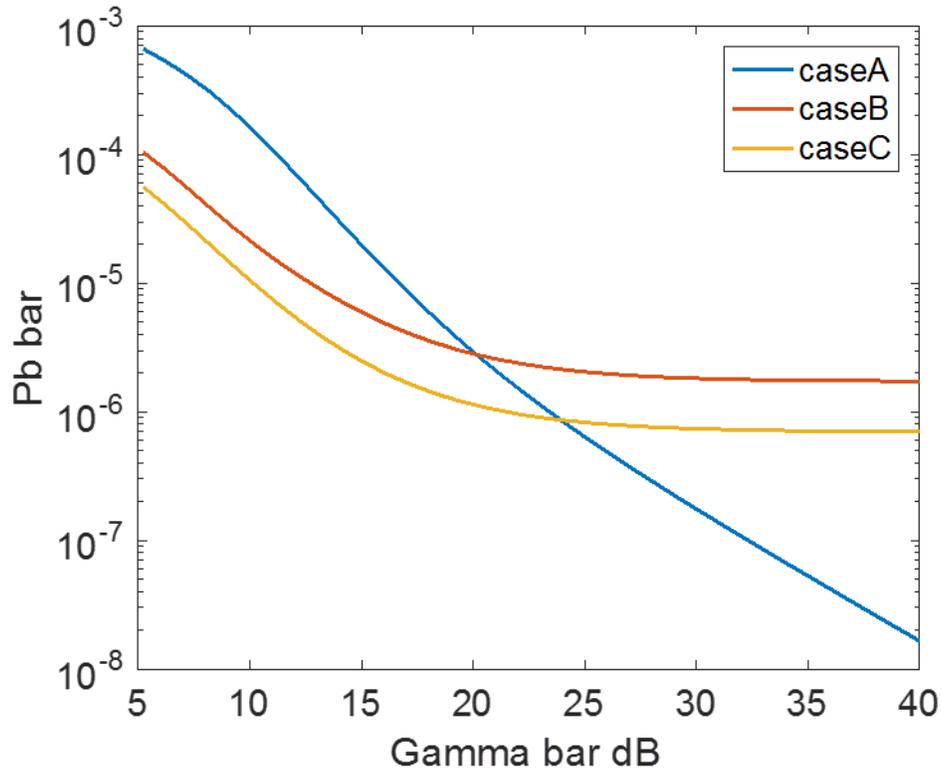


Figure 6. Average BER comparisons of Cases A, B, and C. Average is taken over Rician fading of Rician factor $K = 5$.

Observe also in Figure 6 that the average BER of Cases B and C are bounded (flat lines) as the SNR is increasing, while that of Case A is consistently decreasing. This is because Cases B and C are under jamming, while Case A is under no jamming. The probability density function of the instantaneous SNR in a Rician fading channel of Case A, written as equation (1), is a function of $\bar{\gamma}$, while the probability density function of SJNR in a Rician fading channel for Cases B and C is a function of $\bar{\gamma}_J$.

Figure 7 shows a comparison between $\bar{\gamma}$ and $\bar{\gamma}_J$ whose relationship is described in equation (3). Observe that as E_b/N_0 increases, $\bar{\gamma}$ is proportionally increasing since it is a function of E_b/N_0 , while $\bar{\gamma}_J$ is saturated because it is limited by $E_b/N_J = 10$ dB. Since $\bar{\gamma}_J$ is slowly increasing compared to $\bar{\gamma}$, it affects the average BER of Cases B and C and causes them to flatten, as shown in Figure 5. In other words, under a jamming environment, the average BER is limited by $E_s/N_J = \log_2(M)E_b/N_J$.

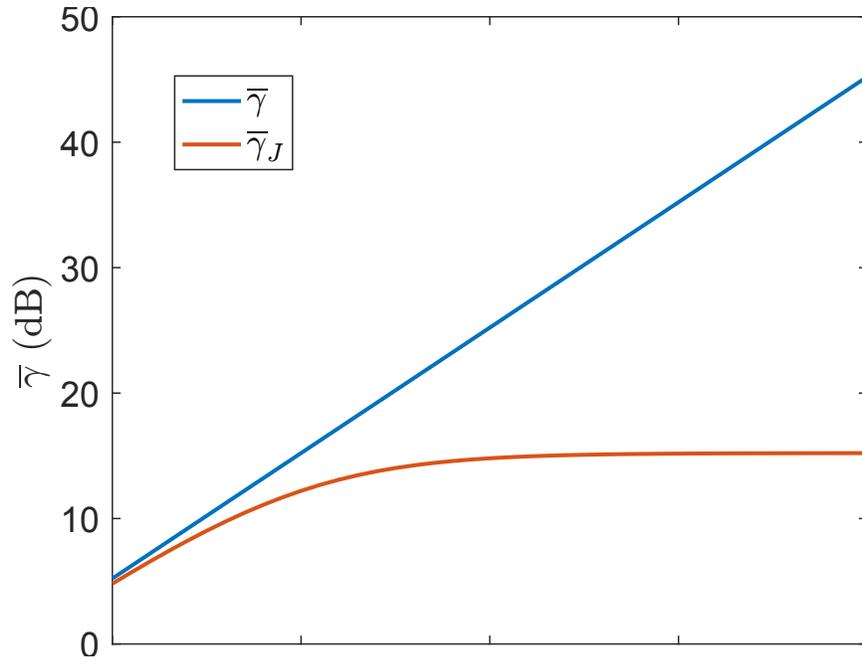


Figure 7. Comparison between average SNR with no jamming case and with jamming case.

Figure 8 presents the BER versus $\gamma_b = E_b/N_0$ for the second set of twenty ACM modes under AWGN. These ACMs are combinations of two modulations—QPSK and 8PSK—and the LDPC FEC of 12 code rates [9, p. 13]. The crossing points at $P_b(\gamma) = 10^{-5}$ marked by red triangles are the lower boundaries of γ_i for ACM mode i under AWGN when there is no jamming.

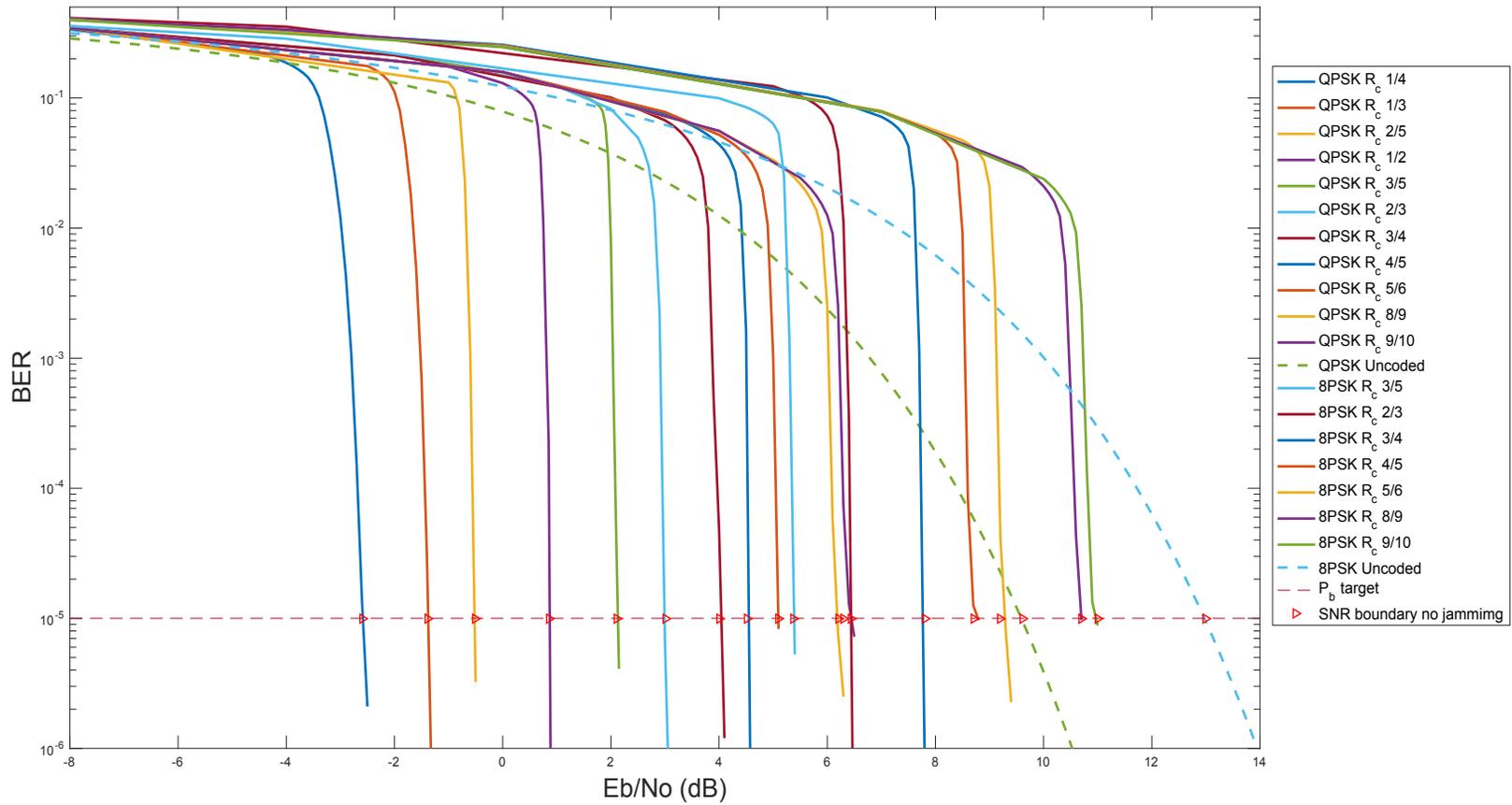


Figure 8. Plots of bit error rate $P_b(\gamma)$ versus $\gamma = E_b / N_0$ in dB for 20 ACM modes applied FEC LDPC coding under AWGN, assuming no jamming. Crossing points at $P_b(\gamma) = 10^{-5}$ (i.e., lower boundaries of γ_i for ACM mode i) are marked by triangles in red. LDPC code word length used in simulation is $n = 64,800$, as stated in DVB-S2 [9, p. 13].

Table 4 lists the corresponding 20 modes of the adaptive code modulations. The SJNR from column 5 in Table 4 is computed from equation (7).

TABLE 4

SNR BOUNDARIES FOR 20 ACM MODES APPLIED FEC LDPC CODING UNDER AWGN OR AWGN PLUS JAMMING WHEN $\frac{E_b}{N_j} = 10$ dB and $\beta = 1$

ACM Mode Index i	ACM Mode	Bandwidth Efficiency $\frac{R(\gamma_i)}{B} = (\log_2 M) R_c$ (bits / (s · Hz))	SNR Lower Bound γ_i in dB for ACM Mode i , i.e., $P_b(\gamma_{b,i}) = 10^{-5}$ $\gamma_i \leq \gamma < \gamma_{i+1}$	
			No Jamming	Under Jamming
0	No Transmission	0	–	–
1	QPSK + $R_c 1/4$	0.5	$\gamma_{b,1} = -2.8$	$\gamma_{b,1,J} = -2.57$
2	QPSK + $R_c 1/3$	0.66	$\gamma_{b,2} = -1.5$	$\gamma_{b,2,J} = -1.18$
3	QPSK + $R_c 2/5$	0.8	$\gamma_{b,3} = -0.59$	$\gamma_{b,3,J} = -0.19$
4	QPSK + $R_c 1/2$	1	$\gamma_{b,4} = 0.85$	$\gamma_{b,4,J} = 1.41$
5	QPSK + $R_c 3/5$	1.2	$\gamma_{b,5} = 2.1$	$\gamma_{b,5,J} = 2.87$
6	QPSK + $R_c 2/3$	1.33	$\gamma_{b,6} = 3.0$	$\gamma_{b,6,J} = 3.97$
7	QPSK + $R_c 3/4$	1.5	$\gamma_{b,7} = 3.9$	$\gamma_{b,7,J} = 5.12$
8	QPSK + $R_c 4/5$	1.6	$\gamma_{b,8} = 4.5$	$\gamma_{b,8,J} = 5.94$
9	QPSK + $R_c 5/6$	1.66	$\gamma_{b,9} = 5.0$	$\gamma_{b,9,J} = 6.65$
10	QPSK + $R_c 8/9$	1.77	$\gamma_{b,10} = 6.1$	$\gamma_{b,10,J} = 8.37$
11	QPSK + $R_c 9/10$	1.8	$\gamma_{b,11} = 6.3$	$\gamma_{b,11,J} = 8.72$
12	QPSK no coding	2	$\gamma_{b,12} = 9.6$	$\gamma_{b,12,J} = 20$
13	8PSK + $R_c 3/5$	1.8	$\gamma_{b,13} = 5.38$	$\gamma_{b,13,J} = 7.22$
14	8PSK + $R_c 2/3$	2	$\gamma_{b,14} = 6.43$	$\gamma_{b,14,J} = 8.94$
15	8PSK + $R_c 3/4$	2.25	$\gamma_{b,15} = 7.72$	$\gamma_{b,15,J} = 11.61$
16	8PSK + $R_c 4/5$	2.4	$\gamma_{b,16} = 8.61$	$\gamma_{b,16,J} = 14.23$
17	8PSK + $R_c 5/6$	2.5	$\gamma_{b,17} = 9.2$	$\gamma_{b,17,J} = 16.95$
18	8PSK + $R_c 8/9$	2.66	$\gamma_{b,18} = 10.51$	$\gamma_{b,18,J} = 19.48$
19	8PSK + $R_c 9/10$	2.7	$\gamma_{b,19} = 10.82$	$\gamma_{b,19,J} = 17.64$
20	8PSK no coding	3	$\gamma_{b,20} = 13$	$\gamma_{b,20,J} = \infty$ cannot be achieved

Observe that modes 10, 11, and 12 (QPSK with 8/9 and 9/10 code rate and QPSK no coding) have a higher cutoff SNR, while their bandwidth efficiencies are lower than mode 12 (8PSK with

3/5 code rate). Thus, for the no-jamming case (Case A), modes 10 to 12 are deleted. For Case B, modes 18 to 20 have a cutoff SNR (no jamming) lower than $E_s/\beta N_j$. These cutoff values $\gamma_{b,i,j}$ cannot satisfy the target BER = 10^{-5} from equation (7) or Theorem 1. Modes 10 to 12 have lower bandwidth efficiency but higher cutoff SNR values than mode 13. Therefore, a total of six ACM modes in Case B will be deleted.

For Case C, it is assumed that the BER after LDPC decoding is still a type of $Q(\alpha)$ function. For calculation purposes, the jamming signal $E_b/N_j = 10$ dB is used; however, the receiver does not have the jamming state information in this case. The calculation uses the proposed algorithm from Figure 3 to determine the ACM modes for each region by testing the cutoff SNR obtained from Case A in Table 4. Because finding the true BER using LDPC decoding simulation is too time consuming, equation (7) is used to determine minimum SNR that can achieve the target BER. The receiver compares the input cutoff SNR (obtained from Case A) one by one, with the computed SJNR using equation (7).

Figure 9 shows the instantaneous spectral efficiency versus SNR $\gamma = E_b/N_0$ in dB for the ACM with LDPC coding for Cases A, B, and C. Observe similar observations as those made for Figure 4. Figure 10 shows the average bandwidth efficiency versus the average received SNR $\bar{\gamma}$ for the ACM with LDPC. Similar observations can be seen in Figure 5. However, there is a small gap between Case C and Case B when $\bar{\gamma}$ is high. Also, observe that the overall average bandwidth efficiency of the LDPC ACM is higher than that of the convolutional FEC ACM, even if a stricter BER equal to 10^{-5} is applied to the LDPC ACM. This is because the LDPC FEC ACM can have higher code rate than the convolutional FEC ACM considered in this paper.

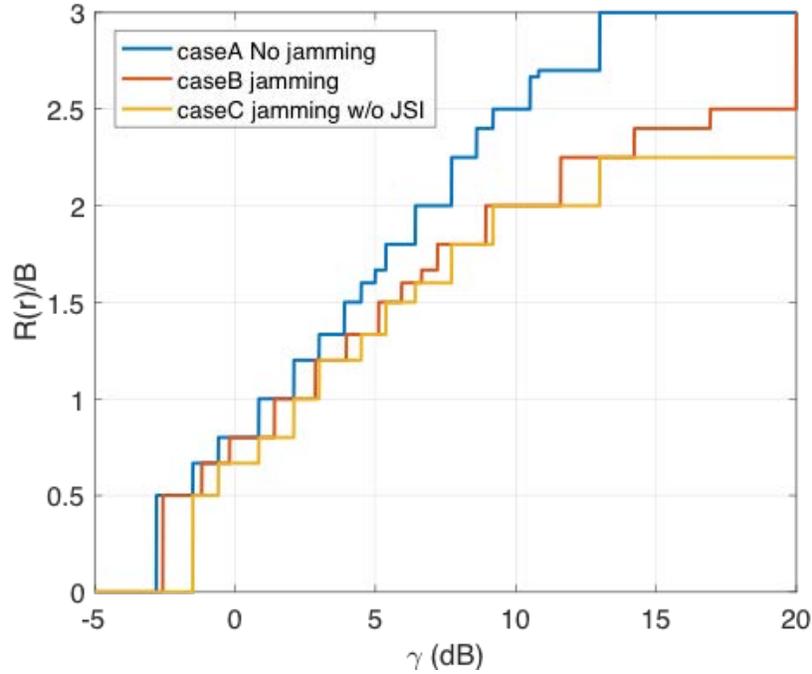


Figure 9. Instantaneous spectral efficiency versus $\gamma = E_b / N_0$ in dB applied FEC LDPC from Table 4, for no jamming (Case A), jamming with JSI (Case B), and jamming without JSI (Case C). No fading is considered.

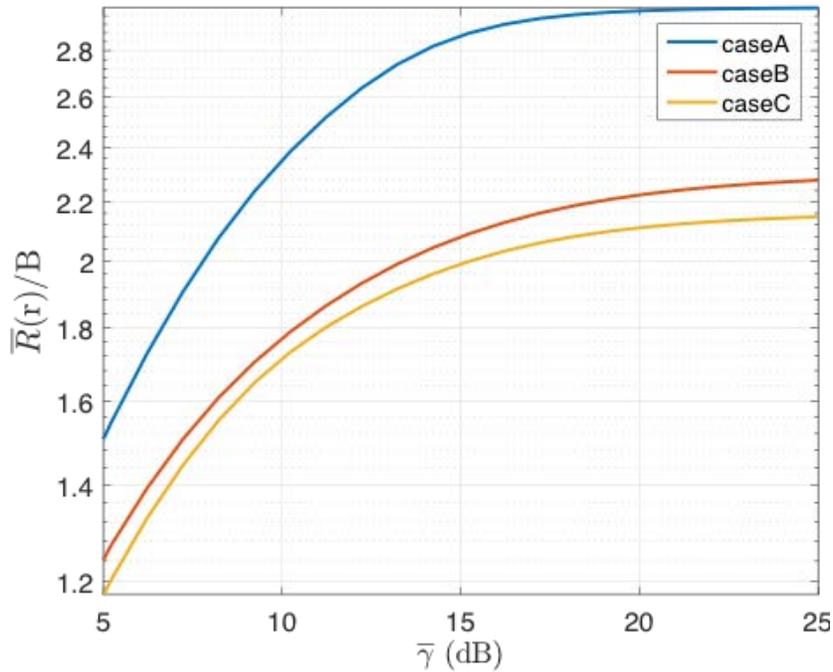


Figure 10. Average bandwidth efficiency comparisons of Cases A, B, and C applied FEC LDPC from Table 4. Average is taken over Rician fading of Rician factor $K = 5$.

CHAPTER 5

CONCLUSIONS

This research studied an adaptive code modulation to enhance the bandwidth and power efficiency in a bandwidth and power constrained communication system environment. The results here can be useful for future satellite communication system designs against jamming environments. The proposed BER-based ACM algorithm, even with no jamming state information, can show bandwidth efficiency performance close to that of ACM mode with JSI; however, the ACM under jamming with JSI shows the best bandwidth efficiency performance compared to other cases. The average BER of the proposed BER-based ACM algorithm shows the best BER performance out of the three ACM schemes (ACM under AWGN, ACM under jamming with no JSI, ACM under jamming with JSI) when the average SNR is small, such as 20 dB, and the BER floor occurs and is subjected to a jamming environment.

For future work, it is desirable to design an ACM where the average BER is within a small upper and lower BER bound from the target BER.

REFERENCES

REFERENCES

- [1] Jason D. Ellis and Michael B. Pursley, “Integration of Adaptive Modulation and Channel Coding with Fountain Coding for Packet Radio Systems,” *IEEE Transactions on Communications*, vol. 63, no. 5. pp. 1510–1521, Feb. 2015.
- [2] Michael B. Pursley and Thomas C. Royster, “Adaptive-Rate Nonbinary LDPC Coding for Frequency-Hop Communications,” 2008 IEEE Military Communications Conference, November 16–19, 2008.
- [3] A. Goldsmith, *Wireless Communication*, 5th ed., Cambridge University Press, 2012.
- [4] Brian J. Wolf and Jacob C. Huang, “Implementation and Testing of the Protected Tactical Waveform (PTW),” IEEE Military Communications Conference, Tampa, FL, October 25–28, 2015.
- [5] Thomas C. Royster and James Streitman, “Performance Considerations for Protected Wideband Satcom,” IEEE Military Communications Conference, Tampa, FL, October 25–28, 2015.
- [6] Matthew Glaser, Kelly Greiner, Bryan Hilburn, Jacob Justus, Christopher Walsh, William Dallas, Joseph Vanderpoorten, and Jo-Chieh Chuang, “Protected MILSATCOM Design for Affordability Risk Reduction (DFARR),” IEEE Military Communications Conference, San Diego, CA, November 18–20, 2013.
- [7] K. L. B. Cook, “Current Wideband MILSATCOM Infrastructure and the Future of Bandwidth Availability,” *IEEE Aerospace and Electronics Systems Magazine*, pp. 23–28, 2008.
- [8] Digital Video Broadcasting (DVB)—Interaction Channel Satellite Distribution Systems European Standard (Telecommunications Series), ETSI EN 301 790 v1.4.1, 2005-09.
- [9] Digital Video Broadcasting (DVB)—Second Generation Framing Structure, Channel Coding And Modulation Systems For Broadcasting, Interactive Services, News Gathering And Other Broadband Satellite Applications (DVB-S2), Final Draft ETSI EN 302 307 v1.2.1, 2009-04.
- [10] Scott Burleigh, “Contact Graph Routing,” NASA/JPL, California Institute of Technology, URL: <https://tools.ietf.org/html/draft-burleigh-dtnrg-cgr-01>, July 8, 2010 [cited February 4, 2016].
- [11] Scott Burleigh, “The Ring Road: CubeSats + DTN—A Proposed Architecture for Cheap, Reliable Network Services for Remote Regions” Space Technology Innovations Conference, URL: <https://www.youtube.com/watch?v=4RusR2PiqPs>, January 24, 2014 [cited February 20, 2016].

REFERENCES (continued)

- [12] “The SDS Bundle Protocol Specification,” Recommended Standard, CCSDS 734.2-B-1, Blue Book, September 2015.
- [13] John Cioffi, “System Design with Codes,” in *Digital Communication: Signal Processing Course*, 2016, retrieved from <http://web.stanford.edu/group/cioffi/book/chap10.pdf>
- [14] T. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*, Wiley-Interscience, 2005.
- [15] J. Proakis, *Digital Communications*, 5th edition, McGraw-Hill, New York, 2001.

APPENDICES

APPENDIX A

MAIN MATLAB CODE USED IN CALCULATIONS

```
Clc; clear; tic;
%-----Parameter-----%
Pb=10^-3; % Target Probability of bit error
step=.1; % define Step for Eb/No
ebnodb = (0:step:40); % Eb/No range in dB
EbNo = 10.^(ebnodb/10); % Convert Eb/No from dB to numerical

EbNj_dB = 10; % define jamming signal

EbNj = 10.^(EbNj_dB/10);

%The effective symbol energy-to-jamming plus noise ratio (SJNR) from
%equation(6)
EbNoJam = (1./((1./EbNo)+(1/EbNj)));
EbNoJam_db = 10.*log10(EbNoJam);

K=5; %Rician K factor
%coefficient power (h) under Ricain fading from equation (2)
Numerator = sqrt(pi/2)*exp(-K/2)*((1+K)*(besseli(0,K/2))+K*(besseli(1,K/2)));

EbNodb = (ebnodb(1):1:ebnodb(end));
ebno = 10.^(EbNodb/10);
ebnoJam = (1./((1./ebno)+(1/EbNj)));

%Average of the received symbol SNR from equation (3)
%No jamming
gamma_bar = Numerator.*ebno;
gamma_bar_dB = 10.*log10(gamma_bar);
%Under jamming
gamma_bar_Jam = Numerator.*ebnoJam;
gamma_bar_Jam_dB = 10.*log10(gamma_bar_Jam);

%----- Transfer Function for each code rate as define in table I -----%

%Rc = 1/4
trellis1_4 = poly2trellis(8,[367 323 275 271]);
spect1_4 = distspec(trellis1_4,4);

%Rc = 1/3
trellis1_3 = poly2trellis(7,[171 165 133]);
spect1_3 = distspec(trellis1_3,4);

%Rc = 1/2
trellis1_2 = poly2trellis(7,[155 177]);
spect1_2 = distspec(trellis1_2,4);

%Rc = 2/3
trellis2_3 = poly2trellis([5 4],[5 23 27; 17 15 13]);
spect2_3 = distspec(trellis2_3,4);

%-----Unjammed QPSK-----%
%Calculating Bit Error Rate for each code rate
```

```

uncodedBER_QPSK = berawgn(ebnodb, 'psk', 4, 'nondiff');
codedBER1_4_QPSK = bercoding(ebnodb, 'conv', 's', 1/4, spect1_4);
codedBER1_3_QPSK = bercoding(ebnodb, 'conv', 's', 1/3, spect1_3);
codedBER1_2_QPSK = bercoding(ebnodb, 'conv', 's', 1/2, spect1_2);
codedBER2_3_QPSK = bercoding(ebnodb, 'conv', 's', 2/3, spect2_3);

%Calculating The SNR boundaries for each code rate using function SNRopt
UncodeSNR_QPSK = SNRopt(uncodedBER_QPSK, Pb, ebnodb);
codeSNR1_4_QPSK = SNRopt(codedBER1_4_QPSK, Pb, ebnodb);
codeSNR1_3_QPSK = SNRopt(codedBER1_3_QPSK, Pb, ebnodb);
codeSNR1_2_QPSK = SNRopt(codedBER1_2_QPSK, Pb, ebnodb);
codeSNR2_3_QPSK = SNRopt(codedBER2_3_QPSK, Pb, ebnodb);

%-----Unjammed 8PSK-----%
%Calculating Bir Error Rate for each code rate
uncodedBER_8PSK = berawgn(ebnodb, 'psk', 8, 'nondiff');
codedBER1_2_8PSK = bercodingmod(ebnodb, 'conv', 's', 1/2, spect1_2, 8);
codedBER2_3_8PSK = bercodingmod(ebnodb, 'conv', 's', 2/3, spect2_3, 8);

%Calculating The SNR boundaries for each code rate using function SNRopt
UncodeSNR_8PSK = SNRopt(uncodedBER_8PSK, Pb, ebnodb);
codeSNR1_2_8PSK = SNRopt(codedBER1_2_8PSK, Pb, ebnodb);
codeSNR2_3_8PSK = SNRopt(codedBER2_3_8PSK, Pb, ebnodb);

%-----Jamming QPSK-----%
JuncodedBER_QPSK = berawgn(EbNoJam_db, 'psk', 4, 'nondiff');
JcodedBER1_4_QPSK = bercoding(EbNoJam_db, 'conv', 's', 1/4, spect1_4);
JcodedBER1_3_QPSK = bercoding(EbNoJam_db, 'conv', 's', 1/3, spect1_3);
JcodedBER1_2_QPSK = bercoding(EbNoJam_db, 'conv', 's', 1/2, spect1_2);
JcodedBER2_3_QPSK = bercoding(EbNoJam_db, 'conv', 's', 2/3, spect2_3);
%
JUncodeSNR_QPSK = SNRopt(JuncodedBER_QPSK, Pb, ebnodb);
JcodeSNR1_4_QPSK = SNRopt(JcodedBER1_4_QPSK, Pb, ebnodb);
JcodeSNR1_3_QPSK = SNRopt(JcodedBER1_3_QPSK, Pb, ebnodb);
JcodeSNR1_2_QPSK = SNRopt(JcodedBER1_2_QPSK, Pb, ebnodb);
JcodeSNR2_3_QPSK = SNRopt(JcodedBER2_3_QPSK, Pb, ebnodb);
%
% %-----Jamming 8PSK-----%
JuncodedBER_8PSK = berawgn(EbNoJam_db, 'psk', 8, 'nondiff');
JcodedBER1_2_8PSK = bercodingmod(EbNoJam_db, 'conv', 's', 1/2, spect1_2, 8);
JcodedBER2_3_8PSK = bercodingmod(EbNoJam_db, 'conv', 's', 2/3, spect2_3, 8);

JUncodeSNR_8PSK = SNRopt(JuncodedBER_8PSK, Pb, ebnodb);
JcodeSNR1_2_8PSK = SNRopt(JcodedBER1_2_8PSK, Pb, ebnodb);
JcodeSNR2_3_8PSK = SNRopt(JcodedBER2_3_8PSK, Pb, ebnodb);

%-----Construct input for case A-----%
%Arrange calculated SNR boundaries in orders
cutoffSNR = [ UncodeSNR_QPSK ...
codeSNR1_4_QPSK...
codeSNR1_3_QPSK ...
codeSNR1_2_QPSK...
codeSNR2_3_QPSK...
UncodeSNR_8PSK...
codeSNR1_2_8PSK ...
codeSNR2_3_8PSK...
];

```

```

cutoff_SNR=unique(cutoffSNR(:));

spectrumUJ = [ log2(4)*1/4 ...
  log2(4)*1/3 ...
  log2(4)*1/2 ...
  log2(4)*2/3 ...
  log2(8)*1/2 ...
  log2(4) ...
  log2(8)*2/3 ...
  log2(8)...
  ];

struct = [ spect1_4;...
  spect1_3;...
  spect1_2;...
  spect2_3;...
  spect1_2;...
  spect2_3;...
  spect2_3;...
  spect2_3...
  ];

code_rate = [1/4;...
  1/3;...
  1/2;...
  2/3;...
  1/2;...
  4;...
  2/3;...
  8;...
  ];

%-----Construct input for case B-----%

%Arrange SNR in orders
JcutoffSNR = [ JUncodeSNR_QPSK ...
JcodeSNR1_4_QPSK...
JcodeSNR1_3_QPSK ...
JcodeSNR1_2_QPSK...
JcodeSNR2_3_QPSK...
JcodeSNR1_2_8PSK ...
JcodeSNR2_3_8PSK...
JUncodeSNR_8PSK
  ];

Jcutoff_SNR=unique(JcutoffSNR(:));

spectrumJ = [ log2(4)*1/4 ...
  log2(4)*1/3 ...
  log2(4)*1/2 ...
  log2(4)*2/3 ...
  log2(8)*1/2 ...
  log2(4) ...
  log2(8)*2/3 ...
  log2(8) ...
  ];

```

```

Jstruct = [ spect1_4 ...
  spect1_3 ...
  spect1_2 ...
  spect2_3 ...
  spect1_2 ...
  spect2_3 ...
  spect2_3 ...
  spect2_3 ...
];

Jcode_rate = [1/4 ...
  1/3 ...
  1/2 ...
  2/3 ...
  1/2 ...
  2/3 ...
  4 ...
  8 ...
];

%-----Construct input for case C-----%

cutoff_SNR1 = cutoff_SNR;

%Algoritm from Fig.2
if Jcutoff_SNR(1)>cutoff_SNR(1)
  spectrumUJ1(1) = 0
  struct1(1) = struct(1)
  code_rate1(1) = 0
else
  spectrumUJ1(1) = spectrumUJ(1)
  struct1(1) = struct(1)
  code_rate1(1) = code_rate(1);
end

for i=2:length(spectrumUJ)
  g=i
  while cutoff_SNR(i)<Jcutoff_SNR(g)
    g=g-1
  end
  spectrumUJ1(i) = spectrumUJ(g)
  struct1(i) = struct(g)
  code_rate1(i) = code_rate(g)
end

%-----Computing -----%

for r=1:length(gamma_bar)
  [UAvgSpectral(r) ,UAvgTruPut(r) , Pb_bar_case1(r,:) ] =
  computingAvg(ebnodb,gamma_bar(r) ,cutoff_SNR,spectrumUJ,struct,code_rate,K) ;
  [JAvgSpectral(r) , JAvgTruPut(r) , Pb_bar_case2(r,:) ] =
  computingAvg(ebnodb,gamma_bar_Jam(r) ,Jcutoff_SNR,spectrumJ,Jstruct,Jcode_rate
  ,K) ;
  [JAvgSpectralcase3(r) , JAvgTruPutcase3(r) , Pb_bar_case3(r,:) ] =
  computingAvgcase3(ebnodb,gamma_bar_Jam(r) ,cutoff_SNR1,spectrumUJ1,struct1,cod
  e_rate1,K) ;

```

```

end

%-----Plotting-----%
%Figure1
figure
semilogy(ebnodb, [codedBER1_4_QPSK; codedBER1_3_QPSK; codedBER1_2_QPSK;
codedBER2_3_QPSK; ...
codedBER1_2_8PSK; uncodedBER_QPSK; codedBER2_3_8PSK; uncodedBER_8PSK; ...
JcodedBER1_4_QPSK; JcodedBER1_3_QPSK; JcodedBER1_2_QPSK; JcodedBER2_3_QPSK;
... JcodedBER1_2_8PSK; JuncodedBER_QPSK; JcodedBER2_3_8PSK;
JuncodedBER_8PSK])
grid on
legend('QPSK R_c 1/4', 'QPSK R_c 1/3', 'QPSK R_c 1/2', 'QPSK R_c 2/3'...
, '8PSK R_c 1/2', 'QPSK Uncoded', '8PSK R_c 2/3', '8PSK Uncoded', ...
'QPSK R_c 1/4', 'QPSK R_c 1/3', 'QPSK R_c 1/2', 'QPSK R_c 2/3'...
, '8PSK R_c 1/2', 'QPSK Uncoded', '8PSK R_c 2/3', '8PSK Uncoded')
title('Pb of QPSK and 8PSK modulation under AWGN no jamming case with no
coded, 1/4,1/3,1/2 and 2/3 code rate')
xlabel('Eb/No (dB)')
ylabel('BER')

%Figure 3
figure
semilogy(gamma_bar_dB, [UJAvgSpectral; JAvgSpectral; JAvgSpectralcase3])
grid on
legend('case1', 'case2', 'case3')
title('Average Spectrum Efficiency')
xlabel('Gamma bar dB')
ylabel('R(r)/B')

%Figure 4
figure
semilogy(gamma_bar_dB, [UjAvgTruPut; JAvgTruPut; JAvgTruPutcase3])
grid on
legend('case1', 'case2', 'case3')
title('Average Through Put')
xlabel('Gamma bar dB')
ylabel('R(r)/B')

%Figure 5
figure
semilogy(gamma_bar_dB, [sum(Pb_bar_case1,2) sum(Pb_bar_case2,2)
sum(Pb_bar_case3,2)])
grid on
legend('case1', 'case2', 'case3')
title('Avg BER (Pb bar)')
xlabel('Gamma bar dB')
ylabel('Pb bar')
toc;

```

APPENDIX B

MATLAB CODE USED IN COMPUTING AVERAGE SPECTRUM EFFICIENCY, AVERAGE THROUGHPUT AND AVERAGE OF PROBABILITY ERROR.

```

function [AvgSpectralsum, AvgTruPutSum, PrUJ, Pb_bar] =
computingAvg(ebnodb, gamma_bar, cutoff_SNR_dB, spectrumUJ, struct, code_rate, K)

% Function computingAvg uses in computing Average spectrum efficiency,
Average throughput and average of probability error.
% % ----- input ----- %
% ebnodb           : Given array of Eb/No in dB values
% gamma_bar       : Average SNR in dB values from given Eb/No
% cutoff_SNR_dB   : SNR Boundaries
% spectrumUJ      : Dmin
% struct
% code_rate
% K
% % ----- output ----- %
% AvgSpectralsum   : Return an array of average spectrum efficiency of each
                    given average SNR
% AvgTruPutSum    : Return an array of average throughput of each given
                    average SNR

% Pb_bar          : Return the average of BER

tol = 1e-4 / gamma_bar^4;
tol(tol>1e-4) = 1e-4;
tol(tol<eps) = eps;

if tol>10^-11
    a=10000;
else
    a=150000;
end

if cutoff_SNR_dB(end)==ebnodb(end)
    cutoff_SNR_dB=cutoff_SNR_dB(1:end-1);
end
%Probability Density Function of Rician Fading as define in equation (1)
IntPDFRician = @(x) ((K+1)/gamma_bar) .*exp(-K-
((K+1)/gamma_bar) .*x) .* (besseli(0, 2*sqrt(K*((K+1)/gamma_bar) .*x)));
cutoff_SNR = 10.^(cutoff_SNR_dB/10); % convert SNR dB to numerical

%-----Calculate Probability of transmission for each region-----%
for i = 2:length(cutoff_SNR_dB)
    PrUJ(i-1) = integral(IntPDFRician, cutoff_SNR(i-1), cutoff_SNR(i));
end
PrUJ(i) = integral(IntPDFRician, cutoff_SNR(i), a);

PrUJ(PrUJ<0)=0;

%-----Calculate Average Spectrum Efficiency -----%
AvgSpectralsum = 0;

```

```

for i = 1:length(PrUJ)
    AvgSpectral = spectrumUJ(i)*PrUJ(i);
    AvgSpectralsum =AvgSpectralsum+AvgSpectral;
end

%-----Calculate Average Probability of Bit Error -----%

for n=1:(length(cutoff_SNR_dB)-1)
    Step=0.05;
    EbNo1_dB = cutoff_SNR_dB(n):Step:cutoff_SNR_dB(n+1);
    EbNo1 = 10.^(EbNo1_dB/10);

    if mod(code_rate(n),1)==0
        for m=1:length(EbNo1_dB)
            Pb_bar_case(m) = berawgn(EbNo1_dB(m), 'psk', code_rate(n), 'nondiff') *...
                ((K+1)/gamma_bar) * exp(-K-
                ((K+1)/gamma_bar) * EbNo1(m)) * (besseli(0, 2*sqrt(K*((K+1)/gamma_bar) * EbNo1(m))))
            ;
            end
            count1=count1+1;
            Pb_bar(n) = sum(Pb_bar_case)/length(EbNo1_dB);
        else
            for m=1:length(EbNo1_dB)
                Pb_bar_case(m) =
berawgn(EbNo1_dB(m), 'conv', 's', code_rate(n), struct(n)) *...
                ((K+1)/gamma_bar) * exp(-K-
                ((K+1)/gamma_bar) * EbNo1(m)) * (besseli(0, 2*sqrt(K*((K+1)/gamma_bar) * EbNo1(m))))
            ;
            end
            count2=count2+1;

            Pb_bar(n) = sum(Pb_bar_case)/length(EbNo1_dB);
        end
    end

    EbNo1_dB = cutoff_SNR_dB(n):Step:ebnodb(end);
    EbNo1 = 10.^(EbNo1_dB/10);
    if mod(code_rate(n+1),1)==0
        for m=1:length(EbNo1_dB)
            Pb_bar_case(m) =
berawgn(EbNo1_dB(m), 'psk', code_rate(n+1), 'nondiff') *...
                ((K+1)/gamma_bar) * exp(-K-
                ((K+1)/gamma_bar) * EbNo1(m)) * (besseli(0, 2*sqrt(K*((K+1)/gamma_bar) * EbNo1(m))))
            ;
            end
            count1=count1+1;
            Pb_bar(n+1) = sum(Pb_bar_case)/length(EbNo1_dB);
        else
            for m=1:length(EbNo1_dB)
                Pb_bar_case(m) =
berawgn(EbNo1_dB(m), 'conv', 's', code_rate(n+1), struct(n+1)) *...
                ((K+1)/gamma_bar) * exp(-K-
                ((K+1)/gamma_bar) * EbNo1(m)) * (besseli(0, 2*sqrt(K*((K+1)/gamma_bar) * EbNo1(m))))
            ;
            end
            count2=count2+1;

```

```

    Pb_bar(n+1) = sum(Pb_bar_case)/length(EbNo1_dB);
end

%-----Calculate Average Through Put -----%

Min = min(length(Pb_bar),length(PrUJ));
AvgTruPutSum = 0;
for i = 1:Min
    AvgTruPut = spectrumUJ(i)*PrUJ(i)*(1-Pb_bar(i));
    AvgTruPutSum =AvgTruPutSum+AvgTruPut;
end
end

```

APPENDIX C

MATLAB CODE USED IN OPTIMIZING SNR VALUE FOR TARGET BER

```
function SNR = SNROpt(GivenBER,Pb,SNRdb)
%Function SNROpt is used in optimizing SNR value for target BER
% %-----input parameter-----%
%GivenBER : Bit error rate from given SNR
%Pb      : Target BER
%SNRdb   : Array of SNR values in dB

% %-----output parameter-----%
%SNR     : Target SNR

y1 = abs(GivenBER-Pb);
y2 = unique(y1(:));
%
ddd = find(y1==y2(1));
eee = find(y1==y2(2));
SNR= mean([SNRdb(ddd) SNRdb(eee)]);
end
```