

RAPID CARRIER PHASE ACQUISITION FOR
LARGE QAM SIGNAL CONSTELLATIONS

A Thesis

by

JASON DILLARD PRESTON

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 1996

Major Subject: Electrical Engineering

RAPID CARRIER PHASE ACQUISITION FOR
LARGE QAM SIGNAL CONSTELLATIONS

A Thesis

by

JASON DILLARD PRESTON

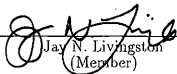
Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

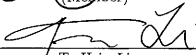
Approved as to style and content by:



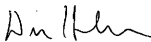
Costas N. Georghiadis
(Chair of Committee)



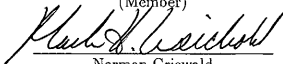
Jay N. Livingston
(Member)



Ta-Hsin Li
(Member)



Don Halverson
(Member)



Norman Griswold
(Head of Department)

December 1996

Major Subject: Electrical Engineering

ABSTRACT

Rapid Carrier Phase Acquisition for
Large QAM Signal Constellations. (December 1996)

Jason Dillard Preston, B.S. Electrical Engineering, Lamar University;
B.S. Mathematics, Lamar University

Chair of Advisory Committee: Dr. Costas N. Georghiadis

This work addresses the problem of rapid carrier-phase acquisition for QAM constellations, and specifically for the 32-QAM, 64-QAM, 128-QAM, and 256-QAM constellations, operating over an AWGN channel. It is assumed that there is no frequency or symbol timing error. Seven algorithms in all are derived and tested. The first is the maximum-likelihood (ML) estimator, which is found to be accurate but impractical to implement. A suboptimal ML type estimator is derived from the ML algorithm and is found to be simpler but still impractical. The power-law (PL) estimator is derived as the low SNR limit of the ML estimator, and is found to be very simple, but having poor error performance. A threshold is added to the PL algorithm and the performance is improved, but not enough for rapid phase acquisition for most cases. An algorithm is derived specifically for use on the cross constellations which further improves performance, but this algorithm is superseded by an estimator based on a trellis structure and the Viterbi Algorithm which is developed. Finally a suboptimal version of the trellis algorithm is developed which reduces the complexity of the trellis algorithm by two-thirds. Both the trellis and the suboptimal trellis algorithm perform well enough for phase acquisition and are simple and practical to implement.

To my very patient wife, Laura

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	A. Problem Background	1
	B. Description of Evaluation Methods Used	3
	C. The Cramer-Rao Bound	6
II	MAXIMUM LIKELIHOOD PHASE ESTIMATION	9
	A. Derivation of the Maximum Likelihood Estimate	9
	1. Implementation of the ML Algorithm	11
	2. Results for the ML Algorithm Simulations	12
	B. A Suboptimal Version of the ML Algorithm	13
	1. Implementation of the SML Algorithm	22
	2. Results for the SML Algorithm Simulations	22
III	POWER LAW PHASE ESTIMATION	30
	A. Derivation of the Power Law Estimate	30
	1. Simulation Results for the PL Algorithm	36
	B. The Power Law with Thresholding	36
	1. Simulation Results for the PLT Algorithm	42
IV	GROUPING ALGORITHM FOR CROSS CONSTELLATIONS	56
	A. Derivation of the Grouping Algorithm	56
	1. Implementation of the Grouping Algorithm	58
	2. Simulation Results for the Grouping Algorithm	61
V	TRELLIS ALGORITHM PHASE ESTIMATION	66
	A. Derivation of the Trellis Algorithm	66
	1. Reducing Computational Load for the Trellis Algorithm	70
	2. Thresholding for the Trellis Algorithm	72
	3. Results for the Trellis Algorithm	73
	B. A Suboptimal Version of the Trellis Algorithm	82
	1. Results for the Suboptimal Trellis Algorithm	83
VI	CONCLUSION	92

	Page
REFERENCES	94
VITA	96

LIST OF TABLES

TABLE		Page
I	SNR's Used in Simulations	6
II	Constellation Subset Information for 32-QAM	19
III	Constellation Subset Information for 64-QAM	20
IV	Constellation Subset Information for 128-QAM	21
V	Constellation Subset Information for 256-QAM	21
VI	Additive and Self-Noise Coefficients	36

LIST OF FIGURES

FIGURE		Page
1	The 32-QAM, 64-QAM, 128-QAM, and 256-QAM Constellations . . .	4
2	Symbol Error Probabilities for Various Constellations	5
3	A Typical Log-Likelihood Function $L(\mathbf{r} \theta)$ for 128-QAM, $\theta = 45^\circ$. .	11
4	ML Estimator Results for 32-QAM	14
5	ML Estimator Results for 64-QAM	15
6	ML Estimator Results for 128-QAM	16
7	ML Estimator Results for 256-QAM	17
8	Suboptimal ML Estimator Results for 32-QAM, $p = 0$	23
9	Suboptimal ML Estimator Results for 32-QAM, $p = 1$	24
10	Suboptimal ML Estimator Results for 64-QAM, $p = 1$	25
11	Suboptimal ML Estimator Results for 128-QAM, $p = 1$	26
12	Suboptimal ML Estimator Results for 128-QAM, $p = 2$	27
13	Suboptimal ML Estimator Results for 256-QAM, $p = 2$	28
14	Suboptimal ML Estimator Results for 256-QAM, $p = 3$	29
15	The 256-QAM Constellation Raised to the 4-th Power, Multiplied by -1	34
16	The 256-QAM Constellation Rotated $\theta = \pi/4$ and Raised to the 4-th Power, Multiplied by -1	35
17	Performance of the Power Law Algorithm for 32-QAM	37
18	Performance of the Power Law Algorithm for 64-QAM	38

FIGURE	Page
19	Performance of the Power Law Algorithm for 128-QAM 39
20	Performance of the Power Law Algorithm for 256-QAM 40
21	PLT Algorithm Performance vs. Threshold for 32-QAM, $K=300$ Symbols 44
22	PLT Algorithm Performance for 32-QAM, $T=1.25$ 45
23	PLT Algorithm Performance for 32-QAM and CRB, $T=1.25$ 46
24	PLT Algorithm Performance vs. Threshold for 64-QAM, $K=300$ Symbols 47
25	PLT Algorithm Performance for 64-QAM, $T=1.45$ 48
26	PLT Algorithm Performance for 64-QAM and CRB, $T=1.45$ 49
27	PLT Algorithm Performance vs. Threshold for 128-QAM, $K=300$ Symbols 50
28	PLT Algorithm Performance for 128-QAM, $T=1.40$ 51
29	PLT Algorithm Performance for 128-QAM and CRB, $T=1.40$ 52
30	PLT Algorithm Performance vs. Threshold for 256-QAM, $K=300$ Symbols 53
31	PLT Algorithm Performance for 256-QAM, $T=1.50$ 54
32	PLT Algorithm Performance for 256-QAM and CRB, $T=1.50$ 55
33	C' for 128-QAM Partitioned into Two 4-PSK Subconstellations 57
34	The Fourth Power of Threshold Detected Data for 128-QAM, $\theta = 0$ 59
35	The Fourth Power of Threshold Detected Data for 128-QAM, $\theta = \pi/16$ 60
36	Grouped Fourth Power Data for 128-QAM, $\theta = 0$ 62
37	Grouping Algorithm Performance for 32-QAM 64
38	Grouping Algorithm Performance for 128-QAM 65

FIGURE	Page
39	A Four State Fully Interconnected Trellis 68
40	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs T for 32-QAM, $K =$ 300 Symbols 74
41	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs K for 32-QAM 75
42	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs T for 64-QAM, $K =$ 300 Symbols 76
43	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs K for 64-QAM 77
44	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs T for 128-QAM, $K =$ 300 Symbols 78
45	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs K for 128-QAM 79
46	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs T for 256-QAM, $K =$ 300 Symbols 80
47	Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs K for 256-QAM 81
48	Suboptimal Trellis Algorithm Performance vs T for 32-QAM, $K =$ 300 Symbols 84
49	Suboptimal Trellis Algorithm Performance vs K for 32-QAM 85
50	Suboptimal Trellis Algorithm Performance vs T for 64-QAM, $K =$ 300 Symbols 86
51	Suboptimal Trellis Algorithm Performance vs K for 64-QAM 87
52	Suboptimal Trellis Algorithm Performance vs T for 128-QAM, $K = 300$ Symbols 88
53	Suboptimal Trellis Algorithm Performance vs K for 128-QAM 89
54	Suboptimal Trellis Algorithm Performance vs T for 256-QAM, $K = 300$ Symbols 90
55	Suboptimal Trellis Algorithm Performance vs K for 256-QAM 91

CHAPTER 1

INTRODUCTION

The objective of this work is to evaluate the performance of several algorithms for rapid carrier phase acquisition for large quadrature amplitude modulation (QAM) signal constellations when the data sequence is unknown and the signal is corrupted by additive white Gaussian noise (AWGN). It is assumed that there is no intersymbol interference and no frequency offset. The constellations considered in this work are the 32-QAM (cross), 64-QAM, 128-QAM (cross), and 256-QAM constellations, but the algorithms discussed in this work can be applied to other QAM signal sets.

A. Problem Background

Burst mode transmission systems, such as time division multiple access (TDMA) and slow frequency hopping (FH), are widely used in wireless digital communications [1]. The best performance in terms of error probability is achieved through the use of phase-coherent demodulation, which requires an accurate carrier phase reference at the receiver [2]. In these types of systems phase estimation for each burst is an important design consideration. Although carrier frequency and symbol timing can be accurately tracked by using a stable reference at the receiver, and by tracking over several bursts, the reference carrier phase tends to drift to such a degree that phase estimation is required for each burst [3].

Communication system designers have traditionally used a phase-locked loop (PLL) as the phase acquisition and tracking mechanism for QAM constellations, but this approach leaves much to be desired since the phase acquisition time of PLL's far

The journal model is *IEEE Transactions on Communications*.

exceeds the time required for optimal phase estimation [3]. The PLL is also subject to hang-up, which is the prolonged dwell at large phase errors. For these reasons most burst mode system designs do not use PLL's for carrier synchronization.

One prevalent phase acquisition technique which has been used in burst mode systems involves the use of a preamble, or a predetermined sequence of symbols, at the beginning of each burst to allow the receiver to use *a priori* knowledge of the transmitted sequence to estimate the phase [1]. While this scheme performs well and is very simple to implement, it has the undesirable property that energy and bandwidth are wasted on the preamble preceding each burst.

Another powerful technique which has been used in burst mode communication systems is to treat the carrier phase θ as an unknown and nonrandom parameter, and estimate it using some parameter estimation algorithm [4] [5] [6] [7] [8]. These algorithms are implemented digitally, and produce their estimate using the sampled sequence of output from the matched filter receiver. Efficient phase estimation algorithms exist for small QAM signal constellations, as well as for M -ary phase shift keying (M -PSK) constellations. For example, applications which use M -PSK constellations can easily handle the phase estimation problem by use of the M -th power algorithm, which is computationally simple and yields good estimates [9]. For large QAM constellations, however, many estimation algorithms are so complex that they are impractical for virtually all applications, and the M -th power algorithm has poor performance, especially for cross constellations.

This work will focus on finding practical phase estimation algorithms that perform well for large QAM signal sets, and that do not require a preamble.

B. Description of Evaluation Methods Used

The purpose of this work is to derive and evaluate several phase estimation algorithms on typical large $\pi/2$ rotationally symmetric QAM signal sets. The algorithms will be implemented for four constellations. These constellations are the 32-QAM, 64-QAM, 128-QAM, and 256-QAM constellations shown in Figure 1. All of the constellations have been normalized to have average energy equal to one.

Since constellations with different sizes will be considered there needs to be a common reference for simulation, so that the algorithm's performance for different constellations can be compared in a fair way. This comparison will be based on symbol error probability. Figure 2 shows an approximation, or more specifically a tight upper bound, to the symbol error probability for the four constellations considered in this paper. This approximation is given in [2] as

$$P_N \approx 2 \cdot \operatorname{erfc}\left(\sqrt{\frac{3\gamma}{2(N-1)}}\right), \quad (1.1)$$

where γ is the signal-to-noise ratio (SNR) per symbol, and N is the number of symbols in the constellation. It is reasonable to assume that, for most applications, the symbol error probability will fall between 10^{-2} and 10^{-6} , since this is a practical operating condition. The choice of SNR's used for the simulations in this work will be directed by this assumption. For instance, the SNR's that give a symbol error probability of 10^{-2} and 10^{-6} for 32-QAM are about 19dB and 24dB per symbol respectively, hence 19dB and 24dB will be used as the SNR's when simulating 32-QAM signals. The SNR's for the other constellations were chosen in a similar way. Table I shows the SNR's used in the simulations for the various constellations.

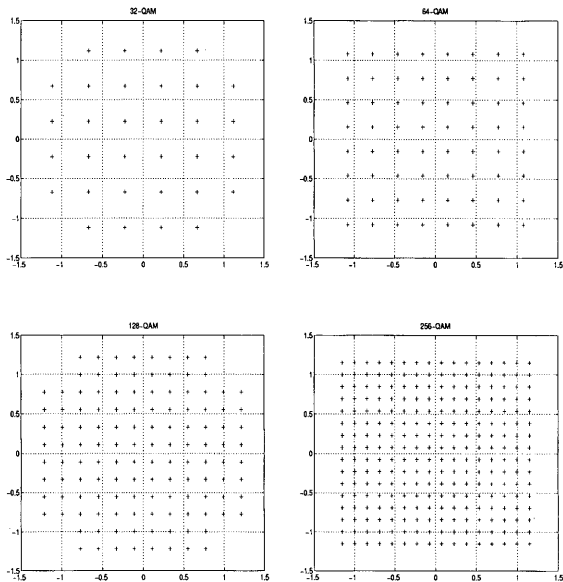


Fig. 1. The 32-QAM, 64-QAM, 128-QAM, and 256-QAM Constellations

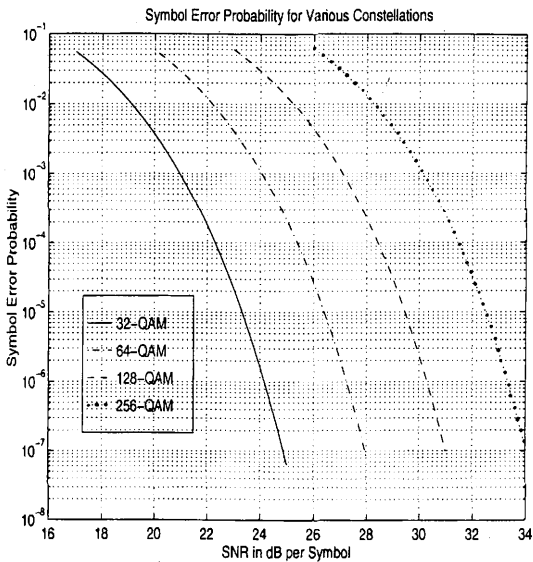


Fig. 2. Symbol Error Probabilities for Various Constellations

Table I. SNR's Used in Simulations

Signal Constellation	SNR in dB per Symbol	
	Low end	High end
32-QAM	19	24
64-QAM	22	27
128-QAM	25	30
256-QAM	28	33

C. The Cramer-Rao Bound

In this work the performance of an algorithm will be measured in terms of mean squared error with respect to the true phase offset. The Cramer-Rao bound (CRB) provides an objective reference to use in evaluating the algorithms. It is established in parameter estimation theory that the CRB is a lower bound on the variance of an unbiased estimate [10]. Therefore the phase estimation algorithms can be evaluated in terms of how close they come to achieving the CRB. In this section the CRB for the phase estimators will be derived.

Let θ represent the unknown phase offset and let $C = \{c_i\}$ be the constellation containing N symbols. It is assumed that the symbols c_i are independent and are all equally likely to be transmitted (i.i.d.). In the k -th symbol interval, the received symbol is given by

$$r(k) = m(k)e^{j\theta} + n(k), \quad (1.2)$$

where $m(k) \in C$ is the transmitted data symbol and $n(k)$ is the complex AWGN variable with $E[n(k)] = 0$ and $E[|n(k)|^2] = N_0/2$.

To obtain the CRB for any estimator of θ we must first form the likelihood function

$$f(\mathbf{r} | \theta) = \prod_{k=1}^K f(r(k) | \theta). \quad (1.3)$$

Taking the logarithm we obtain the log-likelihood function

$$L(\mathbf{r} | \theta) = \ln f(\mathbf{r} | \theta) = \sum_{k=1}^K \ln f(r(k) | \theta). \quad (1.4)$$

The variance of any estimate $\hat{\theta}$ is bounded by the CRB

$$\sigma_{\hat{\theta}}^2 \geq -\frac{1}{E \left[\frac{\partial^2 L(\mathbf{r} | \theta)}{\partial \theta^2} \right]}. \quad (1.5)$$

To simplify the derivation of a useful lower bound we will assume that the estimator has prior knowledge of the transmitted sequence $\mathbf{m} = \{m(k)\}$. This is an optimistic assumption since the estimators derived in this work assume no prior knowledge of \mathbf{m} . The resulting expression will still be a lower bound for $\sigma_{\hat{\theta}}^2$. We can now form the log-likelihood function in terms of the joint Gaussian density.

$$\begin{aligned} L(\mathbf{r} | \theta) &= \sum_{k=1}^K \ln \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{|r(k) - m(k)e^{j\theta}|^2}{N_0}\right) \\ &= K \ln \frac{1}{\sqrt{\pi N_0}} + \frac{1}{N_0} \sum_{k=1}^K -|r(k) - m(k)e^{j\theta}|^2 \\ &= K \ln \frac{1}{\sqrt{\pi N_0}} - \frac{1}{N_0} \sum_{k=1}^K (|r(k)|^2 - 2\text{Re} [r(k)m(k)e^{j\theta}] + |m(k)|^2) \\ &= K_1 + \frac{1}{N_0} \sum_{k=1}^K 2\text{Re} [r(k)m(k)e^{j\theta}] \\ &= K_1 + \frac{1}{N_0} \sum_{k=1}^K (r(k)m(k)^* e^{-j\theta} + r(k)^* m(k)e^{j\theta}), \end{aligned} \quad (1.6)$$

where K_1 includes all terms which do not depend on θ . We can now take the first

and second partial derivatives with respect to θ ,

$$\frac{\partial L(\mathbf{r} | \theta)}{\partial \theta} = \frac{1}{N_0} \sum_{k=1}^K \left(-jr(k)m(k)^* e^{-j\theta} + jr(k)^* m(k)e^{j\theta} \right) \quad (1.7)$$

$$\begin{aligned} \frac{\partial^2 L(\mathbf{r} | \theta)}{\partial \theta^2} &= \frac{1}{N_0} \sum_{k=1}^K \left(-r(k)m(k)^* e^{-j\theta} - r(k)^* m(k)e^{j\theta} \right) \\ &= -\frac{1}{N_0} \sum_{k=1}^K \left(r(k)m(k)^* e^{-j\theta} + r(k)^* m(k)e^{j\theta} \right). \end{aligned} \quad (1.8)$$

We now take the expected value, using the fact that $E[r(k)] = m(k)e^{j\theta}$ and $E[r(k)^*] = m(k)^* e^{-j\theta}$.

$$\begin{aligned} E \left[\frac{\partial^2 L(\mathbf{r} | \theta)}{\partial \theta^2} \right] &= -\frac{1}{N_0} \sum_{k=1}^K \left(E[r(k)m(k)^* e^{-j\theta}] + E[r(k)^* m(k)e^{j\theta}] \right) \\ &= -\frac{1}{N_0} \sum_{k=1}^K \left(E[r(k)] m(k)^* e^{-j\theta} + E[r(k)^*] m(k)e^{j\theta} \right) \\ &= -\frac{1}{N_0} \sum_{k=1}^K \left(|m(k)|^2 + |m(k)|^2 \right) \\ &= -\frac{2}{N_0} \sum_{k=1}^K |m(k)|^2. \end{aligned} \quad (1.9)$$

If we now note that for large K , $\frac{1}{K} \sum_{k=1}^K |m(k)|^2 \approx E[|c_i|^2]$ we can form the CRB as a function of SNR and K .

$$\begin{aligned} \sigma_\theta^2 &\geq \frac{1}{\frac{2}{N_0} K \left(\frac{1}{K} \sum_{k=1}^K |m(k)|^2 \right)} \\ &\approx \frac{1}{2K \frac{E[|c_i|^2]}{N_0}} \\ &= \frac{1}{2K\gamma}, \end{aligned} \quad (1.10)$$

where $\gamma = E[|c_i|^2]/N_0$ is the SNR. The inverse proportionality of γ and σ_θ^2 expressed in (1.10) is common in phase estimation.

CHAPTER II

MAXIMUM LIKELIHOOD PHASE ESTIMATION

The purpose of this chapter is to evaluate the performance of the maximum likelihood (ML) algorithm for phase estimation given an unknown signal sequence. The ML estimate is an important benchmark, since it is well established in parameter estimation theory that any ML estimate is asymptotically unbiased and efficient. That is, for an arbitrarily large number of observations, $r(k)$, the variance of the ML estimate will attain the Cramer-Rao bound.

A. Derivation of the Maximum Likelihood Estimate

In ML phase estimation we treat the carrier phase offset θ as an unknown nonrandom variable and estimate it using maximum likelihood techniques [3], [4], [5], [2]. The MAP estimate is the value of θ which maximizes the *a posteriori* density function

$$f(\theta | \mathbf{r}) = \frac{f(\mathbf{r} | \theta)f(\theta)}{f(\mathbf{r})}. \quad (2.1)$$

If there is no prior knowledge of θ we can assume that $f(\theta)$ is uniform. Hence the value of θ which maximizes $f(\theta | \mathbf{r})$ is identical to the value of θ which maximizes $f(\mathbf{r} | \theta)$, and is therefore the ML estimate. By monotonicity of the logarithm function, the ML estimate also maximizes $\ln f(\mathbf{r} | \theta)$, which is the log-likelihood function given in (1.4). So to produce the ML estimate $\hat{\theta}$ we need to find the value of θ that maximizes

$$\max_{\theta} L(\mathbf{r} | \theta) = \ln f(\mathbf{r} | \theta) = \sum_{k=1}^K \ln f(r(k) | \theta). \quad (2.2)$$

At this point we cannot assume the receiver has any knowledge of the transmitted symbol sequence \mathbf{m} . Instead we assume that the signals in the constellation C are all

equally likely to be transmitted, and take the expected value.

$$\begin{aligned}
 f(r(k) | \theta) &= E_C[f(r(i) | \theta, C)] \\
 &= \sum_{i=1}^N f(r(k) | \theta, c_i) P(c_i) \\
 &= \frac{1}{N} \sum_{i=1}^N f(r(k) | \theta, c_i) \\
 &= \frac{1}{N} \sum_{i=1}^N \exp\left(-\frac{1}{N_0} \cdot |r(k)e^{-j\theta} - c_i|^2\right).
 \end{aligned} \tag{2.3}$$

We can now substitute into (2.2), realizing that the $1/N$ term can be omitted since it will not affect the maximization. Hence the ML estimate, $\hat{\theta}_{ML}$, is found by maximizing the log-likelihood function

$$\max_{\theta} L(\mathbf{r} | \theta) = \sum_{k=1}^K \ln \left[\sum_{i=1}^N \exp\left(-\gamma \cdot |r(k)e^{-j\theta} - c_i|^2\right) \right]. \tag{2.4}$$

Explicit solution for a value of θ which maximizes $L(\mathbf{r} | \theta)$ is impossible, so $\hat{\theta}$ must be obtained by use of numerical methods. Therefore any algorithm that attempts to produce an ML estimate using (2.4) can only claim to produce an estimate that is within a certain tolerance of $\hat{\theta}_{ML}$. Further complicating the matter is the fact that γ is actually an unknown parameter which also needs to be estimated. In a time division multiple access system, for example, γ could be estimated from symbol error rates during previous transmission bursts, assuming the statistics of the noise in the channel do not change rapidly. For the purpose of simulating the ML phase estimator, however, this work will assume that the estimator has perfect knowledge of γ .

Since the constellations under consideration in this work are $\pi/2$ rotationally symmetric, the log-likelihood function is periodic with period $\pi/2$. A phase offset of θ is indistinguishable from one of $\theta + k\pi/2$ for any integer k , unless some knowledge of the sequence \mathbf{m} is used. Therefore it is sufficient to maximize $L(\mathbf{r} | \theta)$ for $\theta \in [0, \pi/2)$.

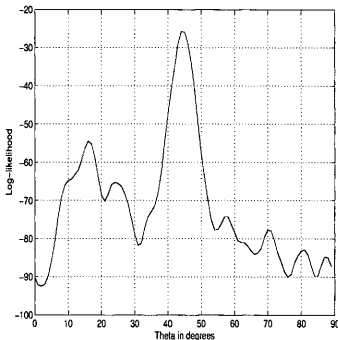


Fig. 3. A Typical Log-Likelihood Function $L(\mathbf{r} | \theta)$ for 128-QAM, $\theta = 45^\circ$

As shown in Figure 3, $L(\mathbf{r} | \theta)$ may have many local maxima. Maximization over the entire range requires first a coarse then a fine optimization, resulting in many evaluations of $L(\mathbf{r} | \theta)$ for each estimate. Furthermore, each evaluation requires K logarithm operations, and $K \cdot N$ exponentials, scalar multiplications, complex multiplications, and complex norms. For even moderate values of K , and N the computational load is quite considerable.

1. Implementation of the ML Algorithm

In this simulation, as in all of the simulations described in this work, the vector \mathbf{r} is generated by randomly choosing a signal from the constellation C , multiplying by $e^{j\theta}$ where θ is uniformly distributed on $[0, \pi/2)$, and adding the complex AWGN. The program used in the simulations for this algorithm performs a coarse optimization in the range of $[0, \pi/2)$ by evaluating $L(\mathbf{r} | \theta)$ at 2° increments. After the approximate

location of the maximum has been found the program uses a second order Newton's method algorithm to refine the estimate. After the refined estimate is produced, the squared error from the true phase offset is determined. This procedure is repeated for 1000 iterations to obtain an average performance.

The Newton's method algorithm locates the maximum within a tolerance of $5 \cdot 10^{-4}$ radians, or about $2.86 \cdot 10^{-2}$ degrees. If we assume the worst case, that is the maximum error allowed by the tolerance is introduced by the fine optimization algorithm every time an estimate is produced, the squared error introduced is about $8.21 \cdot 10^{-4}$ degrees squared. This worst case tolerance error is at least one order of magnitude below the CRB for all values of K and γ used in the ML estimator simulations presented in this work.

When θ is close to 0 or $\pi/2$ the algorithm may converge on an estimate near the opposite end of the range $[0, \pi/2)$. This is due to the $\pi/2$ periodicity of $L(\mathbf{r} | \theta)$. In these cases, $(\theta - \hat{\theta})^2$ is not a fair measure of the accuracy of the estimate. The measure of squared error used in the simulation is

$$\epsilon^2 = \min \begin{cases} (\theta - \hat{\theta})^2 \\ (\theta - \hat{\theta} + \frac{\pi}{2})^2 \\ (\theta - \hat{\theta} - \frac{\pi}{2})^2 \end{cases} \quad (2.5)$$

which takes into account this problem to produce a fair measure of accuracy.

2. Results for the ML Algorithm Simulations

The simulation results for the ML phase estimator are shown in Figures 4, 5, 6, and 7. These graphs show the mean squared error of the ML estimate, $\sigma_{\hat{\theta}}^2$, versus the vector length K . We can see from these figures that the performance of the ML estimate for

an unknown sequence comes close to achieving the CRB given by (1.10), which was derived for a known sequence.

B. A Suboptimal Version of the ML Algorithm

It is possible to significantly reduce the computational complexity of the ML algorithm with a relatively small loss in accuracy. We begin by examining (2.4), and consider what is the maximum possible contribution to the innermost summation for any particular values of k and i . This maximum contribution is given by

$$\max_{\theta} \exp\left(-\gamma \cdot |r(k)e^{-j\theta} - c_i|^2\right). \quad (2.6)$$

The maximum occurs when $\arg(r(k)e^{-j\theta}) = \arg(c_i)$, and has a value given by

$$\max_{\theta} \exp\left(-\gamma \cdot |r(k)e^{-j\theta} - c_i|^2\right) = \exp\left(-\gamma \cdot (|r(k)| - |c_i|)^2\right). \quad (2.7)$$

Note that this expression no longer depends on θ , so any signal point c_j such that $|c_j| = |c_i|$ will produce the same maximum contribution to the sum. Furthermore, this maximum contribution rapidly becomes smaller as $|r(k)| - |c_i|$ becomes larger. This indicates that most of the sum is made up of terms for which $|r(k)| - |c_i|$ is small, and that some other terms are much less significant.

To take advantage of these facts we first partition the constellation C into the L subsets, $\{S_l\}_{l=1}^L$, which contain the signal points that are at the same distance from the origin, or equivalently, that have equal energy. Also, let M_l be the number of signal points contained in S_l . Let \tilde{c}_{lm} represent the m -th signal point in S_l . We can now rewrite (2.4) in terms of the subsets

$$\max_{\theta} L(\mathbf{r} | \theta) = \sum_{k=1}^K \ln \left[\sum_{l=1}^L \sum_{m=1}^{M_l} \exp\left(-\gamma \cdot |r(k)e^{-j\theta} - \tilde{c}_{lm}|^2\right) \right]. \quad (2.8)$$

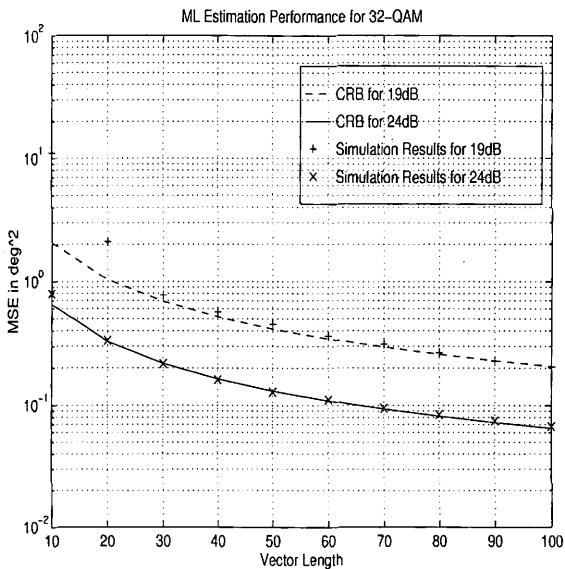


Fig. 4. ML Estimator Results for 32-QAM

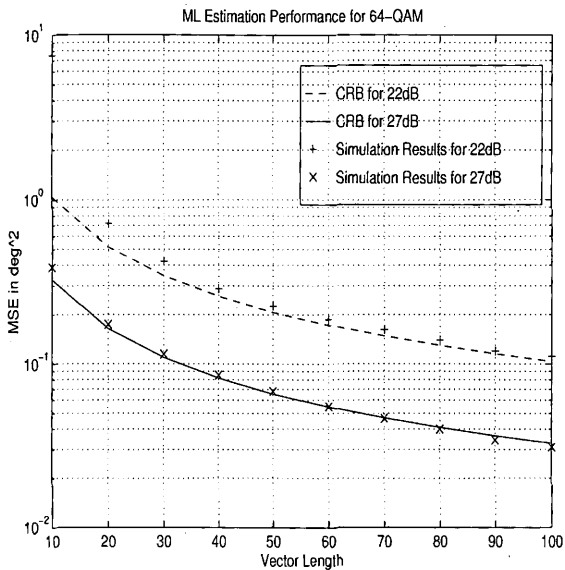


Fig. 5. ML Estimator Results for 64-QAM

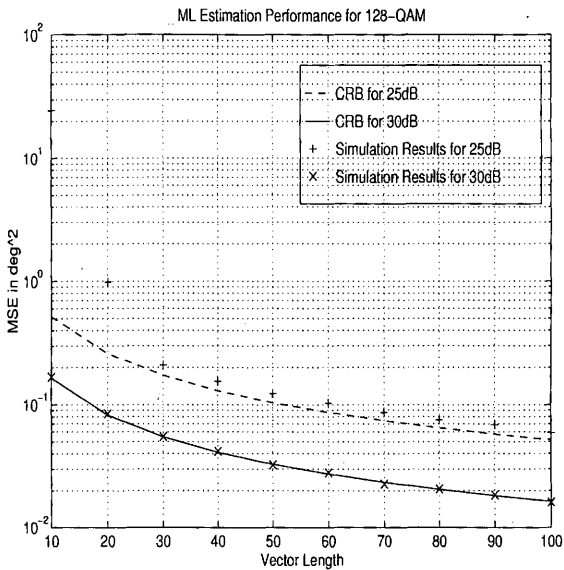


Fig. 6. ML Estimator Results for 128-QAM

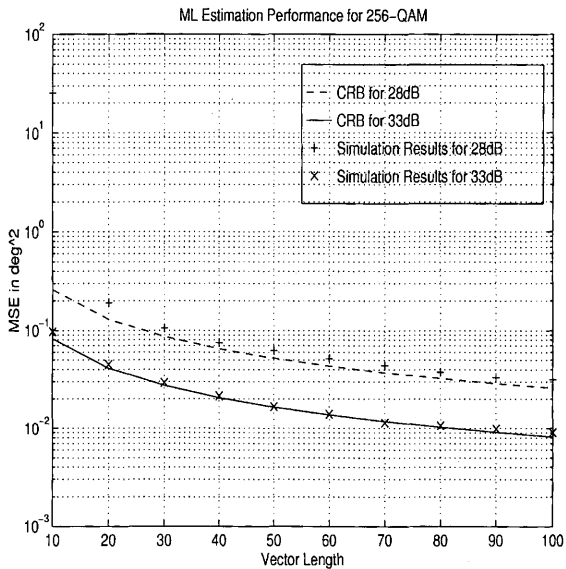


Fig. 7. ML Estimator Results for 256-QAM

This form of the log-likelihood function lends itself to the selective omission and inclusion of terms in the summation corresponding to the individual subsets. If we define $|\tilde{c}_l|$ to be the magnitude of the signal points in subset S_l , or the characteristic magnitude of S_l , a method of deciding which subsets to include can be devised. For example, if we assume that any contribution made by the innermost summation of (2.8) of less than some threshold value T is negligible, we obtain a rule that states that the subset S_l should be included if and only if $(|r(k)| - |\tilde{c}_l|)^2 < -N_0 \ln(T/M_l)$. The nonlinear nature of the logarithm function, however, makes the effect of T on $L(\mathbf{r} | \theta)$ impossible to determine.

The simulations in this work are based on another method of choosing which subsets to include. The idea of this method is to make maximum likelihood decisions about which subset $m(k)$ came from based on $r(k)$. In other words, we choose the subset S_l for which $(|r(k)| - |\tilde{c}_l|)^2$ is minimum. To decrease the effect of making decision errors we also include a certain number of subsets adjacent to the one chosen by the ML criterion. We can assume without loss of generality that the subsets are ordered such that $|\tilde{c}_1| < |\tilde{c}_2| < \dots < |\tilde{c}_L|$, and for notational convenience we can define $S_l = \emptyset$ for all $l \notin \{1, 2, \dots, L\}$. If we now let d_k be the sequence of integers such that

$$(|r(k)| - |\tilde{c}_{d_k}|)^2 < (|r(k)| - |\tilde{c}_l|)^2 \quad \text{for all } l \neq d_k \quad (2.9)$$

we can define a suboptimal log-likelihood function. The algorithm which produces an estimate based on the new log-likelihood function will be referred to as the suboptimal maximum likelihood (SML) algorithm. The suboptimal estimate, $\hat{\theta}_{SML}$, is defined as

Table II. Constellation Subset Information for 32-QAM

32-QAM ($L = 5$)		
l	M_l	$ \tilde{c}_l $
1	4	0.3162
2	8	0.7071
3	4	0.9487
4	8	1.1402
5	8	1.3038

the value of θ which maximizes

$$\max_{\theta} \tilde{L}(\mathbf{r} | \theta) = \sum_{k=1}^K \ln \left[\sum_{l=d_k-p}^{d_k+p} \sum_{m=1}^{M_l} \exp \left(-\gamma \cdot |r(k)e^{-j\theta} - \tilde{c}_{lm}|^2 \right) \right], \quad (2.10)$$

where p is the range of adjacent subsets to be included. For example, if for a particular $r(k)$ the subset whose characteristic magnitude is closest to $|r(k)|$ is S_4 , and $p = 2$, then the subsets included in (2.10) for that particular value of k will be S_2, S_3, S_4, S_5 , and S_6 .

Tables II, III, IV, and V show the values of L , M_l , and $|\tilde{c}_l|$ for the 32-QAM, 64-QAM, 128-QAM, and 256-QAM constellations respectively. The tables show that the difference between the characteristic magnitudes of neighboring subsets gets smaller as the number of points in the constellation increases. This indicates that a larger value of p may be required to produce an accurate phase estimate for the larger constellations, since the probability of choosing the wrong subset is high for larger constellations.

Table III. Constellation Subset Information for 64-QAM

64-QAM ($L = 9$)		
l	M_l	$ \tilde{c}_l $
1	4	0.2812
2	8	0.4880
3	4	0.6547
4	8	0.7868
5	8	0.8997
6	12	1.0911
7	8	1.1751
8	8	1.3274
9	4	1.5275

Table IV. Constellation Subset Information for 128-QAM

128-QAM ($L = 16$)					
l	M_l	$ \tilde{c}_l $	l	M_l	$ \tilde{c}_l $
1	4	0.1562	9	8	1.0000
2	8	0.3492	10	8	1.0467
3	4	0.4685	11	4	1.0932
4	8	0.5631	12	8	1.1370
5	8	0.6439	13	8	1.2198
6	12	0.7809	14	16	1.2591
7	8	0.8410	15	8	1.3343
8	8	0.9500	16	8	1.4399

Table V. Constellation Subset Information for 256-QAM

256-QAM ($L = 32$)											
l	M_l	$ \tilde{c}_l $	l	M_l	$ \tilde{c}_l $	l	M_l	$ \tilde{c}_l $	l	M_l	$ \tilde{c}_l $
1	4	0.1085	9	8	0.6945	17	16	1.0000	25	16	1.2127
2	8	0.2425	10	8	0.7276	18	8	1.0233	26	8	1.2696
3	4	0.3254	11	4	0.7593	19	8	1.0683	27	8	1.3061
4	8	0.3911	12	8	0.7896	20	8	1.0901	28	8	1.3416
5	8	0.4472	13	8	0.8471	21	8	1.1324	29	4	1.4100
6	12	0.5423	14	16	0.8745	22	8	1.1530	30	8	1.4266
7	8	0.5841	15	8	0.9267	23	8	1.1732	31	8	1.5244
8	8	0.6598	16	4	0.9762	24	4	1.1931	32	4	1.6270

1. Implementation of the SML Algorithm

The implementation of the SML algorithm is very similar to that of the optimal ML algorithm. The coarse optimization of $\tilde{L}(\mathbf{r} | \theta)$ is performed at 2° increments, and the fine optimization is performed using the same Newton's method algorithm as before. For this algorithm the program performed 1500 iterations to obtain an average performance.

2. Results for the SML Algorithm Simulations

The simulation results for the SML phase estimator are shown in Figures 8 through 14. Figures 8 and 9 show the performance of the SML estimator for 32-QAM when $p = 0$ and $p = 1$ respectively. When $p = 0$ there is a significant loss in accuracy from the ML estimate, but when $p = 1$ the performance of the SML estimator is nearly identical to that of the ML estimator, shown in Figure 4. Likewise, Figure 10 shows that for 64-QAM and $p = 1$ the SML algorithm has performance comparable to that of the ML algorithm shown in Figure 5. The results for 128-QAM are shown in Figures 11 and 12 for $p = 1$ and $p = 2$ respectively. When $p = 1$ there is a performance loss noticeable for lower SNRs and small vector lengths. For $p = 2$ the performance is comparable to that of the ML algorithm, shown in Figure 6. The results for 256-QAM are shown in Figures 13 and 14 for $p = 2$ and $p = 3$ respectively. When $p = 2$ there is a performance loss noticeable mainly at small vector lengths. For $p = 3$ the performance is again comparable to that of the ML algorithm, shown in Figure 7.

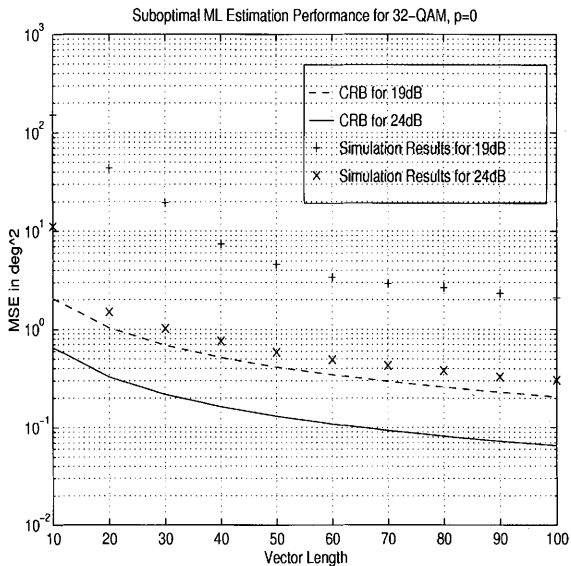


Fig. 8. Suboptimal ML Estimator Results for 32-QAM, $p = 0$

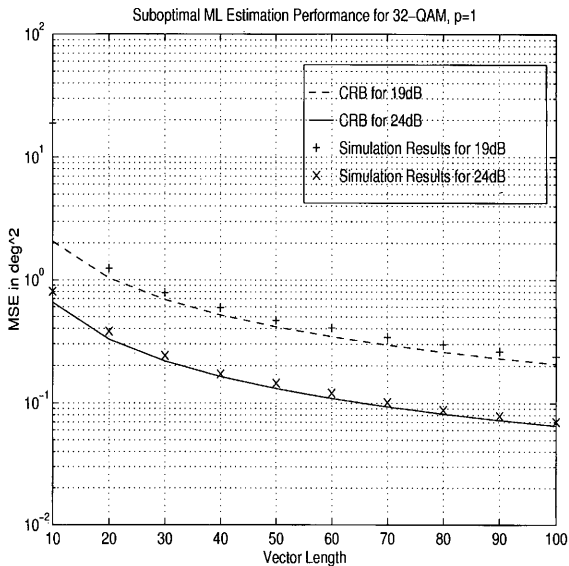


Fig. 9. Suboptimal ML Estimator Results for 32-QAM, $p = 1$

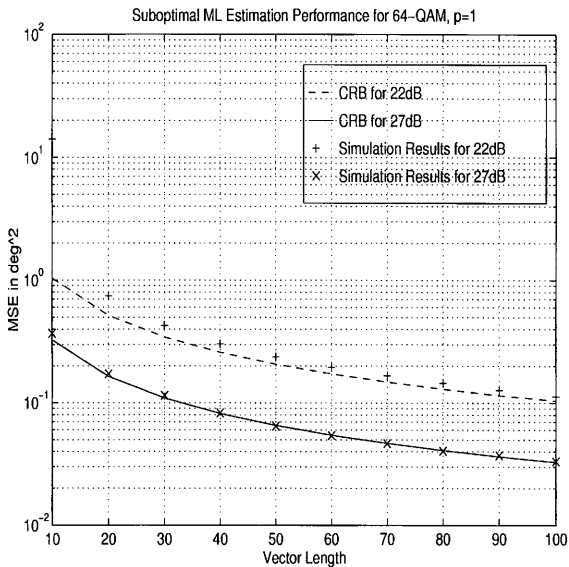


Fig. 10. Suboptimal ML Estimator Results for 64-QAM, $p = 1$

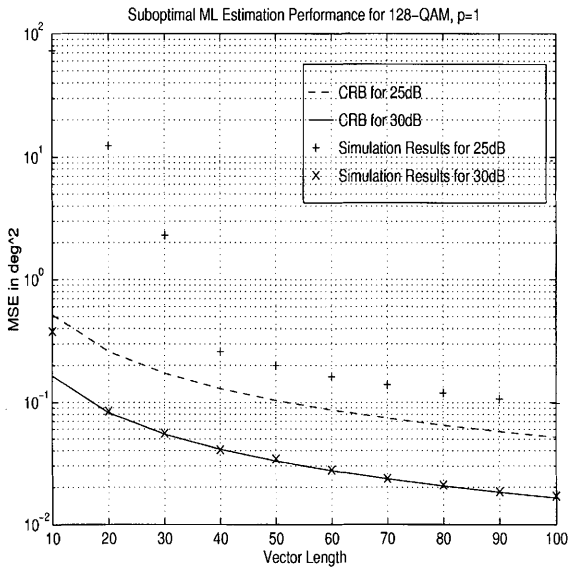


Fig. 11. Suboptimal ML Estimator Results for 128-QAM, $p = 1$

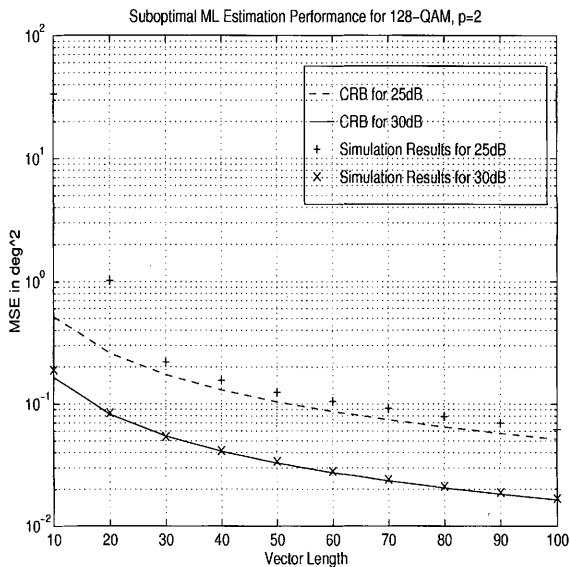


Fig. 12. Suboptimal ML Estimator Results for 128-QAM, $p = 2$

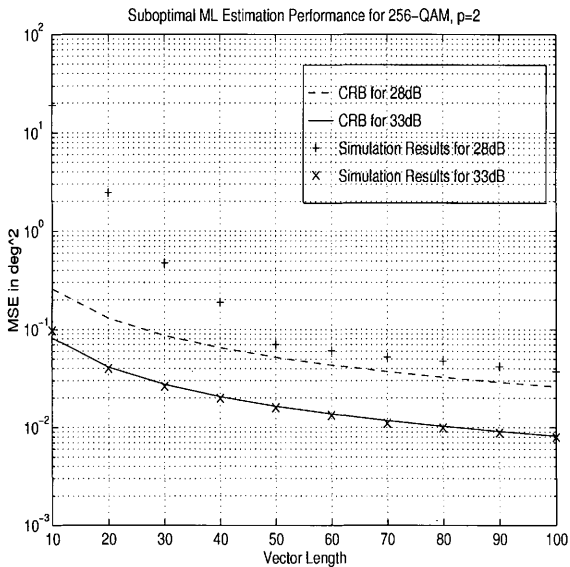


Fig. 13. Suboptimal ML Estimator Results for 256-QAM, $p = 2$

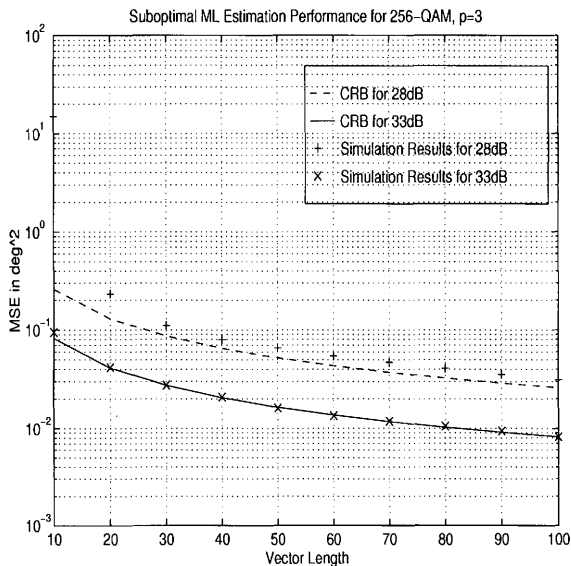


Fig. 14. Suboptimal ML Estimator Results for 256-QAM, $p = 3$

CHAPTER III

POWER LAW PHASE ESTIMATION

The purpose of this chapter is to examine the performance of the power law (PL) phase estimator when it is applied to the high order QAM constellations under consideration in this work. It is well known that the power law is a simple algorithm which produces good estimates when applied to PSK constellations. However, as shown in [9], the algorithm performs poorly for QAM constellations. This poor performance is due mainly to the effects of self-noise, or noise introduced by the transmitted sequence \mathbf{m} itself, independent of the AWGN.

The power law estimate is derived from the log-likelihood function by letting $\gamma \rightarrow 0$. The resulting estimate does not depend on γ , so this parameter does not need to be known or estimated as with the ML estimate.

A. Derivation of the Power Law Estimate

A derivation of the PL estimate is given in [9], and is repeated here in more detail for completeness. To derive the power law estimate we begin with the log-likelihood function, which gives the ML estimate $\hat{\theta}$ as the value of θ that maximizes

$$\begin{aligned} \max_{\theta} L(\mathbf{r} | \theta) &= \sum_{k=1}^K \ln E_C \left[\exp \left(-\gamma \cdot |r(k)e^{-j\theta} - c_i|^2 \right) \right] \\ &= \sum_{k=1}^K \ln E_C \left[\exp \left(-\gamma \cdot (|r(k)|^2 - 2\text{Re}\{r(k)c_i^* e^{-j\theta}\} + |c_i|^2) \right) \right] \\ &= \sum_{k=1}^K \ln E_C \left[\exp \left(-\gamma \cdot (-2\text{Re}\{r(k)c_i^* e^{-j\theta}\} + |c_i|^2) \right) \right] - \sum_{k=1}^K \gamma |r(k)|^2. \end{aligned} \quad (3.1)$$

The summation on the far right is a constant with respect to θ , and can be omitted. Thus we can obtain the ML estimate by maximizing

$$\max_{\theta} L'(\mathbf{r} | \theta) = \sum_{k=1}^K \ln l(\mathbf{r} | \theta) \quad (3.2)$$

where

$$l(\mathbf{r} | \theta) = E_C \left[\exp \left(\gamma \cdot (r(k)c_i^* e^{-j\theta} + r(k)^* c_i e^{j\theta} - |c_i|^2) \right) \right]. \quad (3.3)$$

We can now expand $l(\mathbf{r} | \theta)$ into a power series

$$\begin{aligned} l(\mathbf{r} | \theta) &= E_C \left[1 + \sum_{m=1}^{\infty} \frac{\gamma^m}{m!} \left(\gamma \cdot (r(k)c_i^* e^{-j\theta} + r(k)^* c_i e^{j\theta} - |c_i|^2) \right)^m \right] \\ &= 1 + E_C \left[\sum_{m=1}^{\infty} \gamma^m \sum_{n=0}^m \sum_{n'=0}^{m-n} A(m, n, n') (r(k)c_i^* e^{-j\theta})^n (r(k)^* c_i e^{j\theta})^{n'} (|c_i|^2)^{m-n-n'} \right] \\ &= 1 + \sum_{m=1}^{\infty} \gamma^m \sum_{n=0}^m \sum_{n'=0}^{m-n} A(m, n, n') r(k)^n r(k)^{*n'} e^{-j(n-n')\theta} E_C [c_i^{m-n} c_i^{*m-n'}], \end{aligned} \quad (3.4)$$

where $A(m, n, n') = \frac{(-1)^{m-n-n'}}{n!n'!(m-n-n)!}$. The first term along with all other terms for which $n = n'$ combine to form the phase independent part of $l(\mathbf{r} | \theta)$, which converges to 1 as $\gamma \rightarrow 0$. Hence the phase dependent part of $l(\mathbf{r} | \theta)$ is given by

$$l_d(\mathbf{r} | \theta) = \sum_{m=1}^{\infty} \gamma^m \sum_{n=0}^m \sum_{\substack{n'=0 \\ n' \neq n}}^{m-n} A(m, n, n') r(k)^n r(k)^{*n'} e^{-j(n-n')\theta} E_C [c_i^{m-n} c_i^{*m-n'}]. \quad (3.5)$$

It is easily shown that for constellations that are invariant under rotation of $2\pi/P$,

$$E[c_i^p c_i^{*q}] = 0 \quad \text{for } p - q \notin \{0, P, 2P, \dots\}, \quad (3.6)$$

hence the terms in (3.5) are nonzero only if $n - n' \in \{0, P, 2P, \dots\}$. For M -PSK constellations $P = M$, but for the QAM constellations considered in this work $P = 4$. The function $l_d(\mathbf{r} | \theta)$ is dominated by the nonzero term in (3.5) for which m is

smallest. This occurs for $m = P$, and hence we have

$$\begin{aligned} L_0'(\mathbf{r} | \theta) &= \sum_{k=1}^K \ln \left(1 + \frac{\gamma^P}{P!} (r(k)^P E[c_i^{*P}] e^{-jP\theta} + r(k)^{*P} E[c_i^P] e^{jP\theta}) \right) \quad \text{as } \gamma \rightarrow 0 \\ \hat{\theta}_{PL} &= \frac{2}{P!} \operatorname{Re} \left[E[c_i^{*P}] \sum_{k=1}^K r(k)^P e^{-jP\theta} \right]. \end{aligned} \quad (3.7)$$

This is the log-likelihood function for vanishingly small γ , and it is maximized for

$$\hat{\theta}_{PL} = \frac{1}{P} \arg \left(E[c_i^{*P}] \sum_{k=1}^K r(k)^P \right). \quad (3.8)$$

Clearly this estimate is much simpler to implement than the ML estimate. The term $E[c_i^{*P}]$ is a constant which lies on the negative real axis for the constellations under consideration. There is no need to maximize a nonlinear function, and a new estimate can easily be produced for each signalling interval using information from the previous interval.

Figures 15 and 16 give an indication of how the algorithm works. Figure 15 shows the effects of raising the points of the 256-QAM constellation to the 4-th power, and then multiplying by a negative constant. The result of this transform is a collection of 64 distinct points, since four signal points in the original constellation are mapped to a single point in the transformed set. The points along the positive real axis in Figure 15 correspond to the signal points in the original constellation which lie along the diagonal lines $y = \pm x$. The transformed set is clearly biased in the direction of these points. Figure 16 shows the results of performing the same transform on the 256-QAM constellation with a phase offset of $\theta = \pi/16$. The resulting pattern has been rotated by $\pi/4$ or 4θ , and is therefore biased in the direction of $e^{j4\theta}$. From this illustration it can be surmised that the signal points which lie on or near the positive real axis in figure 15 are good indicators of θ since they tend to drive the summation in (3.8) in the direction of $e^{j4\theta}$. Signal points which lie farther off the real axis or

in the left half of the plane are more likely to introduce error (self-noise) into the estimate. An algorithm which attempts to exploit these properties will be discussed later in this chapter.

In [9] an approximation to the MSE performance of $\hat{\theta}_{PL}$ for moderate to high γ is given as

$$\sigma_{\hat{\theta}_{PL}}^2 \approx \frac{B_1}{2K\gamma} + \frac{B_2}{K} \quad (3.9)$$

where

$$\begin{aligned} B_1 &= \frac{E[|c_i^2|]E[|c_i^{2P-1}|]}{|E[c_i^P]|^2} \geq 1 \\ B_2 &= \frac{2|E[c_i^P]|^2 E[|c_i^{2P}|] - E^2[c_i^P]E[c_i^{*2P}] - E^2[c_i^{*P}]E[c_i^{2P}]}{4P^2|E[c_i^P]|^4}. \end{aligned} \quad (3.10)$$

The two terms in (3.9) correspond to effects of additive noise and self-noise respectively. For any M -PSK constellation, $B_1 = 1$ and $B_2 = 0$, indicating that there is no self-noise present and that $\sigma_{\hat{\theta}_{PL}} \approx \sigma_{\hat{\theta}}$, the CRB. Hence under conditions of moderate to high SNR for PSK constellations the power law produces a good estimate. For QAM constellations, however, the self-noise term tends to dominate.

Table VI gives the values for B_1 and B_2 for the constellations considered in this work. When it is taken into account that the $1/2\gamma$ factor which multiplies B_1 can be on the order of 10^{-2} to 10^{-4} for typical SNRs, it is clear that the effects of self-noise tend to dominate the performance of the estimate.

It is also interesting to note that the ratio $B_1 : B_2$ is roughly half as big for the cross constellations as it is for the square constellations. This would indicate that the effects of self-noise are of greater significance for cross constellations than for square ones. This is due to the fact that the cross constellations do not have proportionately as many of the high-energy signal points which tend to produce better estimates of θ , and which introduce less self-noise.

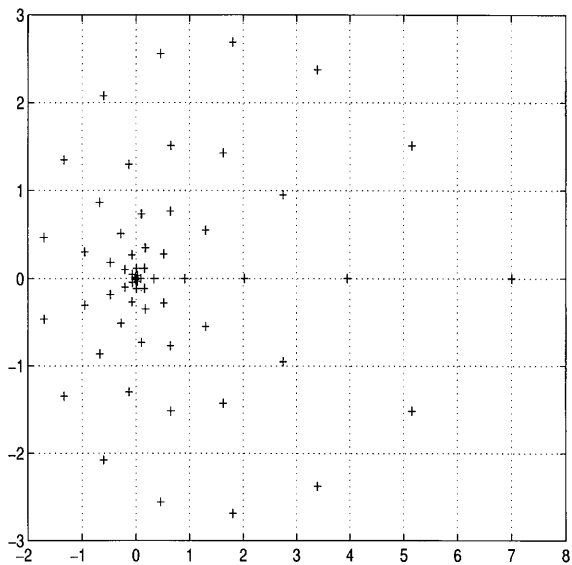


Fig. 15. The 256-QAM Constellation Raised to the 4-th Power, Multiplied by -1

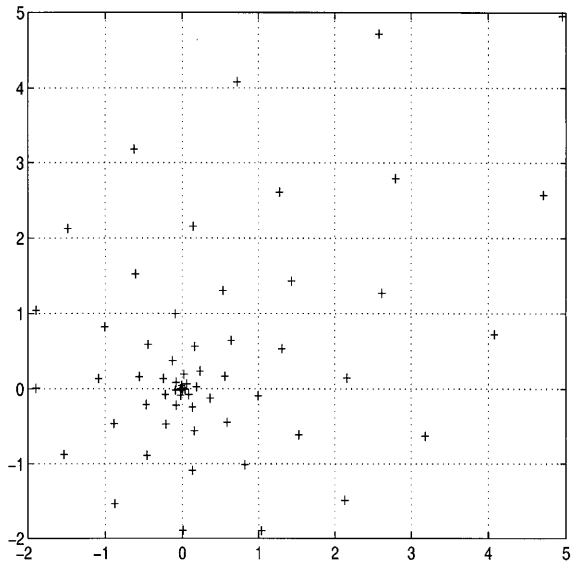


Fig. 16. The 256-QAM Constellation Rotated $\theta = \pi/4$ and Raised to the 4-th Power, Multiplied by -1

Table VI. Additive and Self-Noise Coefficients

Constellation	B_1	B_2
32-QAM	52.6316	3.1413
64-QAM	5.8081	0.1674
128-QAM	62.0675	3.7929
256-QAM	6.2685	0.2019

1. Simulation Results for the PL Algorithm

Simulations were run for the PL algorithm for 10,000 iterations. Figures 17 through 20 show the simulation results for the four constellations, along with the CRB. The approximation $\sigma_{\hat{\theta}_{PL}}^2$ is plotted only for the higher SNRs simulated because the curves for the lower and higher SNRs fall very close together. This algorithm clearly does not perform well enough to be used for rapid phase estimation for the constellations considered here. The next section will discuss a way to improve the performance of the power law estimator.

B. The Power Law with Thresholding

In some cases it is possible to produce a more accurate estimate than $\hat{\theta}_{PL}$ by omitting some terms from the summation in (3.8). As stated in the previous section, some signal points in the constellations tend to be detrimental to the accuracy of the estimate $\hat{\theta}_{PL}$, while other points introduce less self-noise and tend to help the accuracy of $\hat{\theta}_{PL}$. It would seem desirable to devise a method to produce an estimate using only the information at the receiver that tends to produce a more accurate estimate. Since the most beneficial signal points lie towards the corners of the constellation, a simple

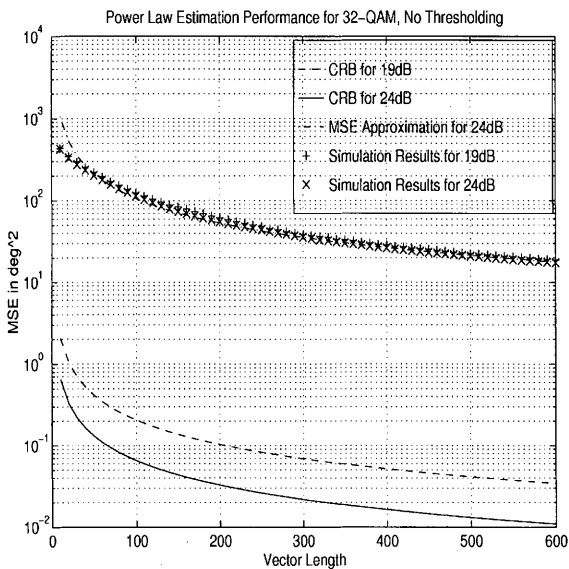


Fig. 17. Performance of the Power Law Algorithm for 32-QAM

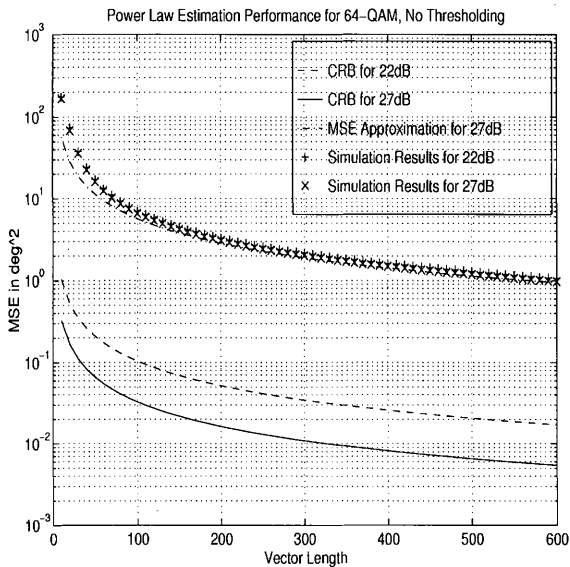


Fig. 18. Performance of the Power Law Algorithm for 64-QAM

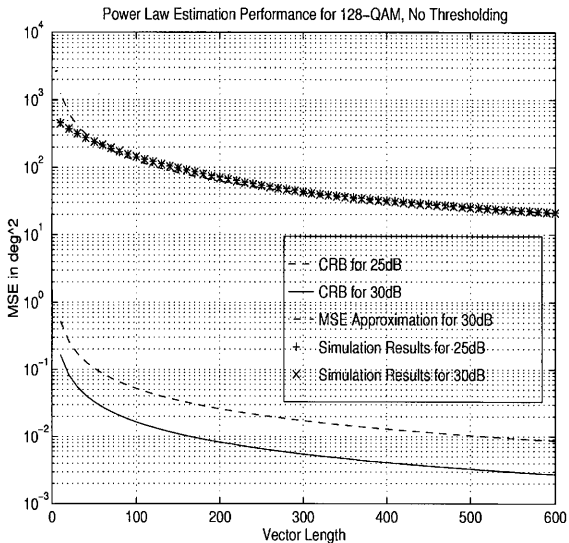


Fig. 19. Performance of the Power Law Algorithm for 128-QAM

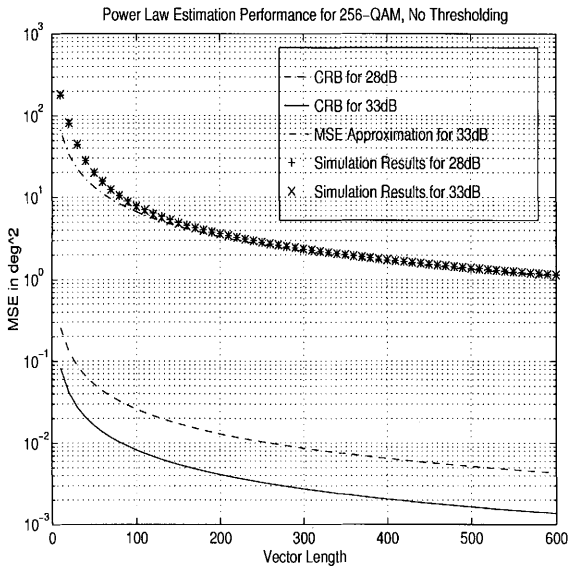


Fig. 20. Performance of the Power Law Algorithm for 256-QAM

way to partially achieve this goal is to introduce a threshold, T . Any received signals for which $|r(k)| > T$ will be used in estimating the phase offset, and the rest will not. We define $C' = \{c'_i\}$ as the subset of C containing the N' signal points that lie outside the circle of radius T centered at the origin. The new estimate is given by

$$\hat{\theta}_{PLT} = \frac{1}{P} \arg \left(E[c'_i^{*P}] \sum_{k=1}^{K'} r'(k)^P \right) \quad (3.11)$$

where $\mathbf{r}' = \{r'(k)\}$ is the vector containing all received signals for which $|r(k)| > T$, and K' is the number of symbols in \mathbf{r}' .

Under conditions of high SNR we can assume that the receiver perfectly detects the symbols which correspond to transmission of a signal point from C' . Using this assumption we can derive an approximation to the performance of $\hat{\theta}_{PLT}$ based on the approximation given in (3.9). If we assume that at least one symbol from C' has been transmitted the approximate error variance is

$$\sigma_1^2 \approx \frac{1}{E[K']} \left(\frac{B'_1}{2\gamma'} + B'_2 \right) \quad (3.12)$$

where $E[K'] = K \cdot E[P(m(k) \in C')] = K \frac{N'}{N}$, $\gamma' = \frac{E[|c'|^2]}{N_0} = \frac{E[|c'|^2]}{E[|c|^2]} \gamma$, and B'_1 and B'_2 are the constants defined in (3.10) calculated for C' . Since the constellations have been normalized we can write the approximation as

$$\sigma_1^2 \approx \frac{N}{N'K} \left(\frac{B'_1}{2\gamma E[|c'|^2]} + B'_2 \right). \quad (3.13)$$

If no symbols from C' are detected at the receiver, an estimate must be produced based on no information. The error $\theta - \hat{\theta}$ in this case is uniformly distributed along the interval $(-\pi/4, \pi/4)$. Therefore the variance for this case is $\sigma_0^2 = \pi^2/48$. To obtain the overall approximate performance we take the expected value over the two cases of no symbols from C' detected and at least one symbol from C' detected. If we let P_0 be the probability that no symbols from C' are detected, the resulting approximation

is given as

$$\begin{aligned}\sigma_{\hat{\theta}_{PLT}}^2 &\approx (1 - P_0)\sigma_1^2 + P_0\sigma_0^2 \\ &= \left(1 - \left(\frac{N - N'}{N}\right)^K\right) \frac{N}{N'K} \left(\frac{B'_1}{2\gamma E[|c'|^2]} + B'_2\right) + \left(\frac{N - N'}{N}\right)^K \frac{\pi^2}{48}. \quad (3.14)\end{aligned}$$

Since (3.14) is an approximation for the case of no errors in detecting symbols from C' , it can be expected to be more accurate when few detection errors are made. For this reason it should be evaluated at values of T that result in fewer detection errors. A set of reasonable threshold values at which to evaluate (3.14) is $T = \frac{\lfloor l \rfloor + \lfloor l+1 \rfloor}{2}$ for $l = 1, 2, \dots, L - 1$, which can be easily obtained using values listed in Tables II through V.

1. Simulation Results for the PLT Algorithm

The threshold value that produces the best performance for a particular constellation depends on the vector length. For instance, the threshold which minimizes the MSE for a vector length of 100 symbols may not be the same as the best threshold for 300 symbols. In this work the vector length at which the performance is optimized is 300 symbols. However, the optimal threshold for 100 symbols was found to differ only slightly from that of 300 symbols.

Simulations were run for the PLT algorithm for 10,000 iterations. Figure 21 shows the performance of the PLT algorithm versus T for the 32-QAM constellation, as well as $\sigma_{\hat{\theta}_{PLT}}^2$ at appropriate values of T . The difference between the approximation and the simulation results is due mainly to the effect of detection errors. According to this graph the threshold that minimizes the MSE for $K = 300$ is about $T = 1.25$. Figure 22 shows the performance of the PLT versus K and the approximation for $T = 1.25$. Figure 23 shows the performance of the PLT as well as the CRB and

the approximation $\sigma_{\hat{\theta}_{PL}}^2$. Figures 24 through 32 show similar graphs for the other constellations. These graphs indicate that the optimal threshold values are about $T = 1.45$ for 64-QAM, $T = 1.40$ for 128-QAM, and $T = 1.50$ for 256-QAM.

It is interesting to note that for 64-QAM when $T = 1.45$, C' contains only the corner points of the constellation. The performance is very good at this threshold value because no self-noise is introduced unless detection errors occur. For the 256-QAM constellation, inclusion of only the corner points does not result in the best performance for $K = 300$ because the probability of receiving useful phase information is too small. The best threshold for this case is one that includes points which introduce some self-noise, but at the same time provide the receiver with more phase information.

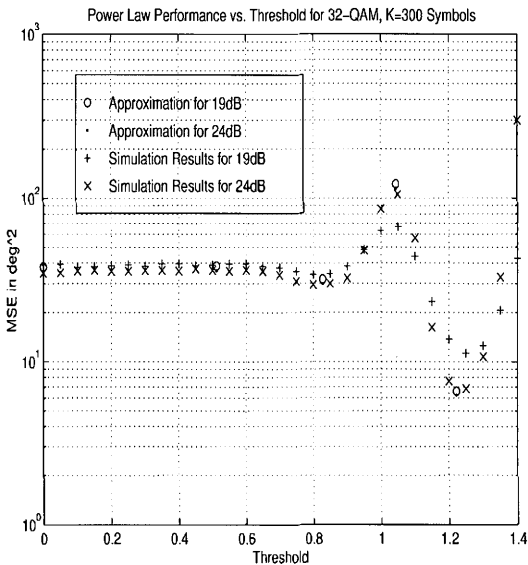


Fig. 21. PLT Algorithm Performance vs. Threshold for 32-QAM, K=300 Symbols

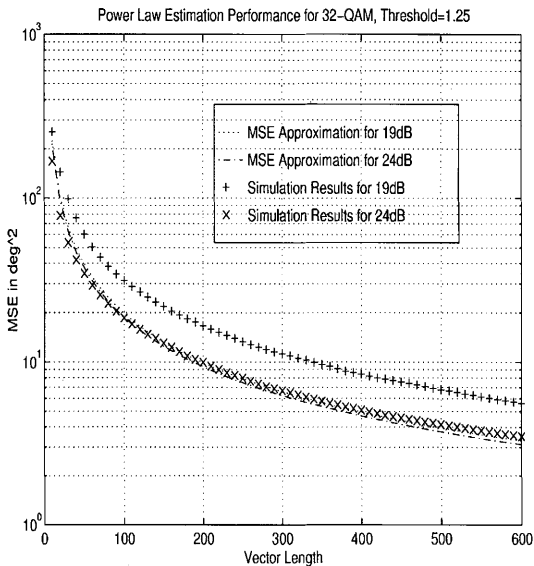


Fig. 22. PLT Algorithm Performance for 32-QAM, T=1.25

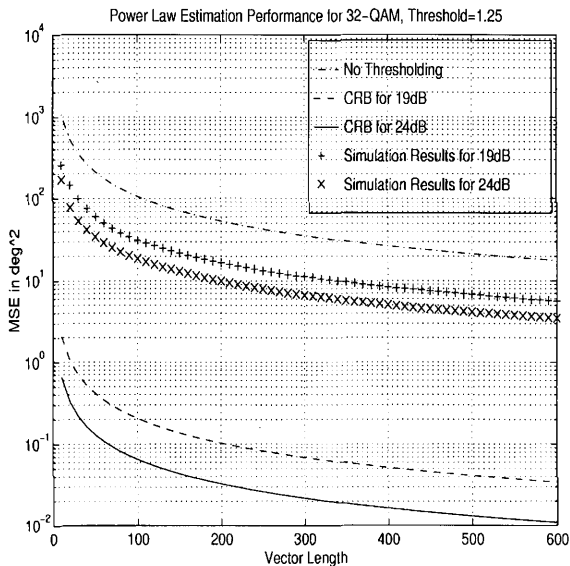


Fig. 23. PLT Algorithm Performance for 32-QAM and CRB, $T=1.25$

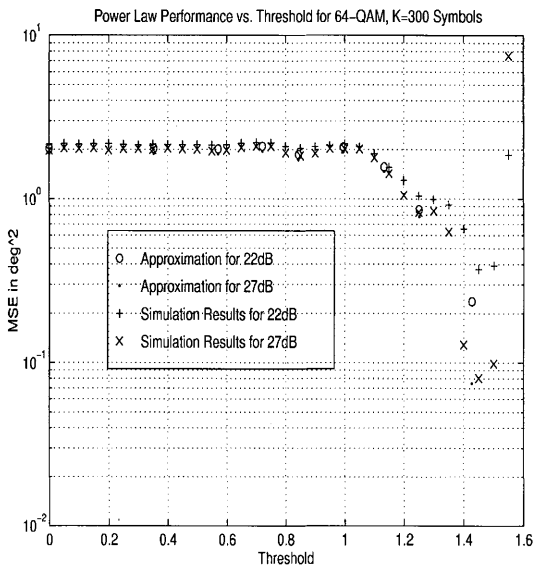


Fig. 24. PLT Algorithm Performance vs. Threshold for 64-QAM, K=300 Symbols

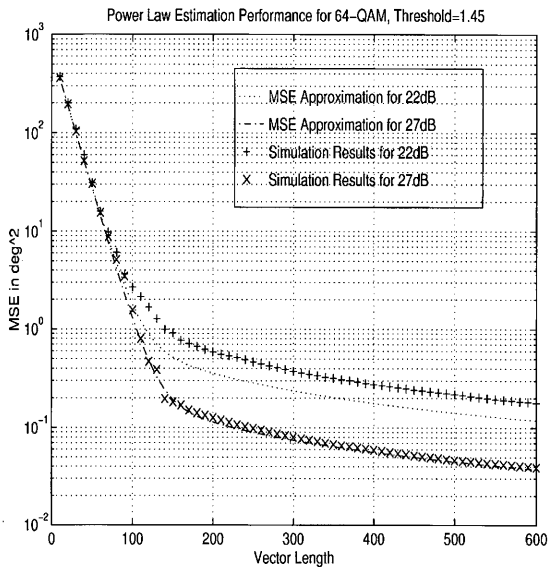


Fig. 25. PLT Algorithm Performance for 64-QAM, $T=1.45$

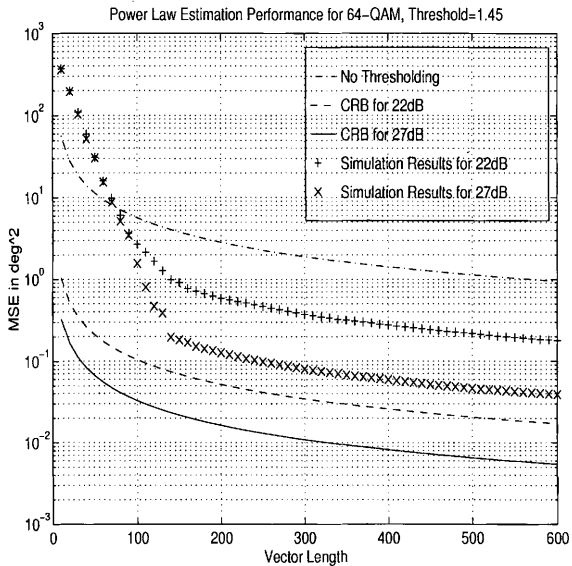


Fig. 26. PLT Algorithm Performance for 64-QAM and CRB, $T=1.45$

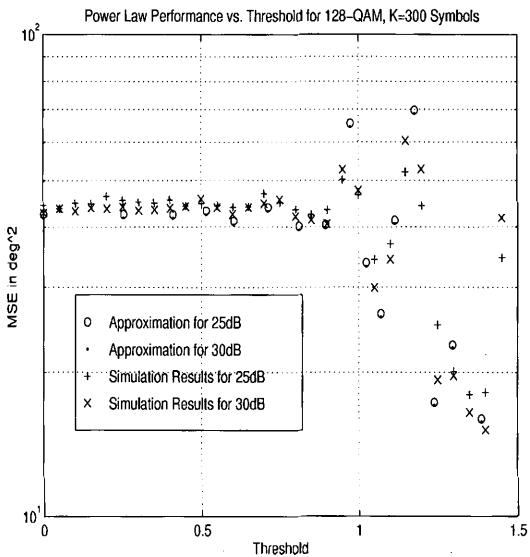


Fig. 27. PLT Algorithm Performance vs. Threshold for 128-QAM, K=300 Symbols

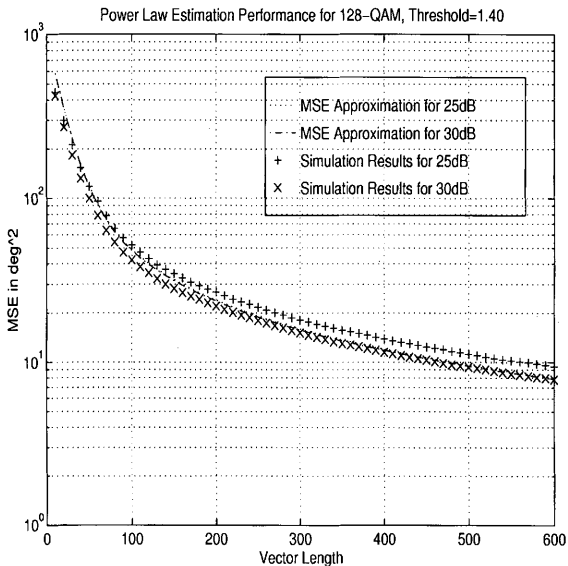


Fig. 28. PLT Algorithm Performance for 128-QAM, $T=1.40$

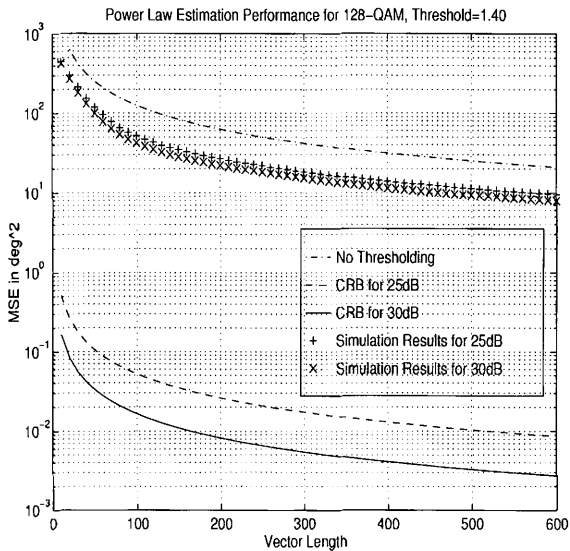


Fig. 29. PLT Algorithm Performance for 128-QAM and CRB, $T=1.40$

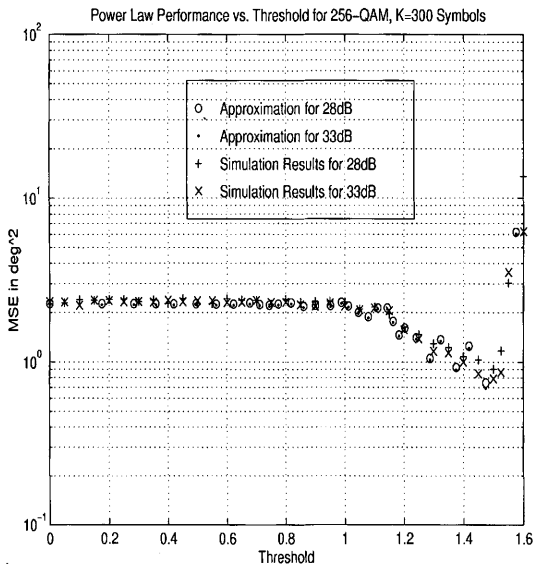


Fig. 30. PLT Algorithm Performance vs. Threshold for 256-QAM, K=300 Symbols

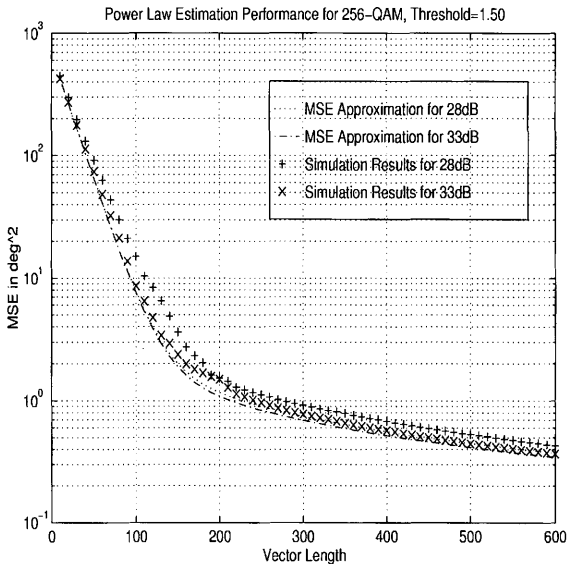


Fig. 31. PLT Algorithm Performance for 256-QAM, $T=1.50$

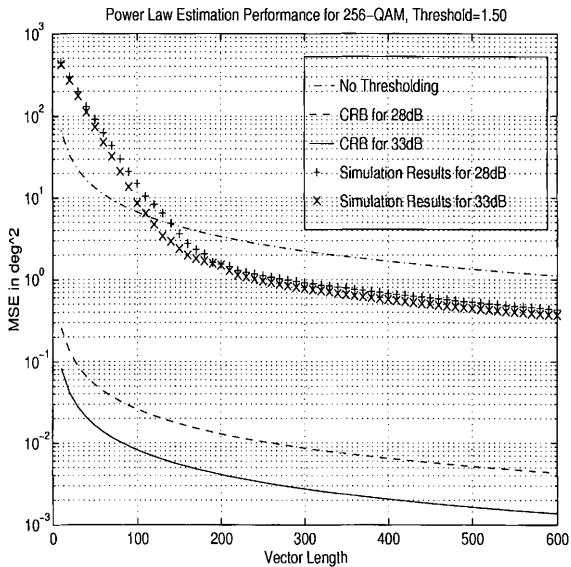


Fig. 32. PLT Algorithm Performance for 256-QAM and CRB, $T=1.50$

CHAPTER IV

GROUPING ALGORITHM FOR CROSS CONSTELLATIONS

The purpose of this chapter is to briefly describe and evaluate an algorithm that was motivated by the success of thresholding in improving the performance of the power law described in Chapter III. This algorithm is only applicable to cross constellations, and it attempts to reduce or eliminate the effects of self-noise which are inevitably introduced when using the PLT algorithm on cross constellations.

If we examine the results for the PLT algorithm for the 32-QAM and 128-QAM constellations we can see that the best performance is achieved when the algorithm attempts to detect only the signals corresponding to the outermost eight symbols. Choosing a value of T such that C' consists of these eight points results in the smallest value for B'_2 . However, the self-noise still tends to be the dominant factor for large K .

A. Derivation of the Grouping Algorithm

First, we partition the eight point constellation C' into two subconstellations C'_1 and C'_2 , such that each subset forms a PSK constellation as shown in Figure 33. If at this point the received symbols \mathbf{r}' could be classified into subsets \mathbf{r}'_1 and \mathbf{r}'_2 , corresponding to transmitted symbols from C'_1 or C'_2 , either subset could be used in equation (3.11) to produce a better estimate than $\hat{\theta}_{PLT}$. In this case we have $B'_1 = 1$, $B'_2 = 0$, and $N' = 4$. Elimination of the self noise in this way greatly enhances the accuracy of the estimate.

Further performance gains are possible, especially at smaller values of K , since only part of the phase information available has been used thus far. In the limit of

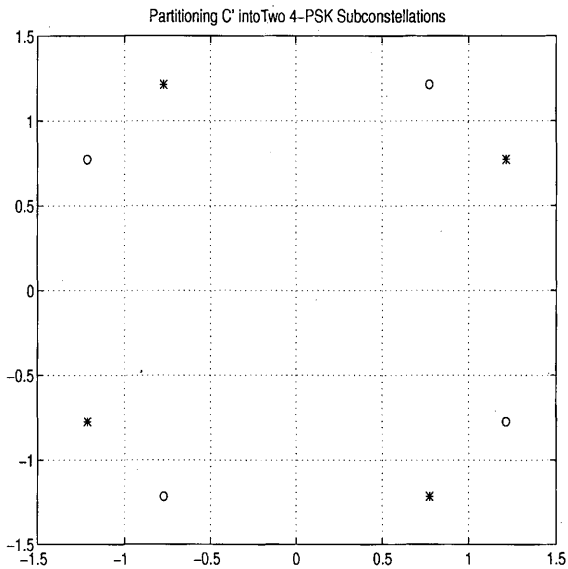


Fig. 33. C' for 128-QAM Partitioned into Two 4-PSK Subconstellations

high SNR for any 4-PSK constellation we have

$$\begin{aligned} E[c_i'^{**4}] \sum_{k=1}^{K'} r'(k)^4 &= c_i'^{**4} \sum_{k=1}^{K'} c_i'^4 e^{j4\theta} \\ &= K' |c_i'|^8 e^{j4\theta}. \end{aligned} \quad (4.1)$$

Since this sum vector always points in the direction of $e^{j4\theta}$ regardless of the original constellation, we can add two such vectors produced from symbols corresponding to different PSK constellations and produce a larger vector which points in the direction of $e^{j4\theta}$. Hence an estimate can be produced which uses all of the phase information available. This estimate is given as

$$\hat{\theta}_G = \frac{1}{4} \arg \left(E[c_{1,i}'^{**4}] \sum_{k=1}^{K'_1} r'_1(k)^4 + E[c_{2,i}'^{**4}] \sum_{k=1}^{K'_2} r'_2(k)^4 \right), \quad (4.2)$$

where K'_1 and K'_2 are the number of received symbols in \mathbf{r}'_1 and \mathbf{r}'_2 respectively. The performance of this estimate can be approximated using (3.14) using the values $B'_1 = 1$, $B'_2 = 0$, and $N' = 8$. It can be shown that in the limit of high SNR this algorithm makes optimal use of the phase information in \mathbf{r}' .

1. Implementation of the Grouping Algorithm

Although the formula for the estimate is known, there is also the problem of classifying the symbols in \mathbf{r}' into \mathbf{r}'_1 and \mathbf{r}'_2 . This task can be simplified by first taking the fourth power of the received symbols. Assuming no detection errors are made, the symbols in \mathbf{r}' will map into two regions on the complex plane, as illustrated in Figure 34. Figure 35 shows the effect of a phase offset of $\theta = \pi/16$ radians, which results in a rotation of $\pi/4$ radians when the fourth power is taken.

Once the fourth power is taken the problem involves locating the two distinct groups of points on the plane, and determining which group each point belongs to.

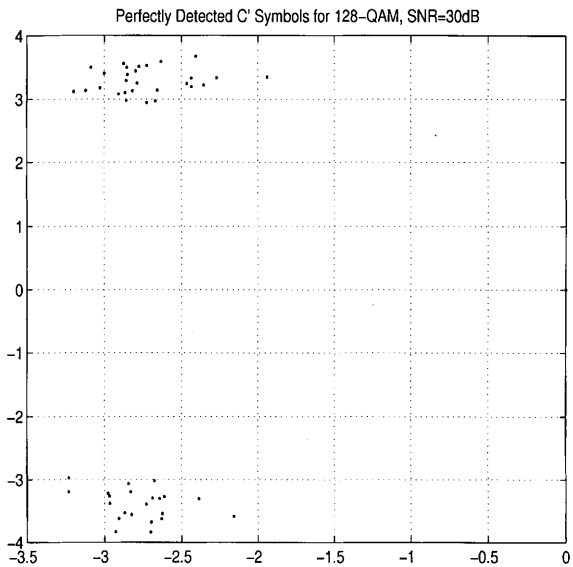


Fig. 34. The Fourth Power of Threshold Detected Data for 128-QAM, $\theta = 0$

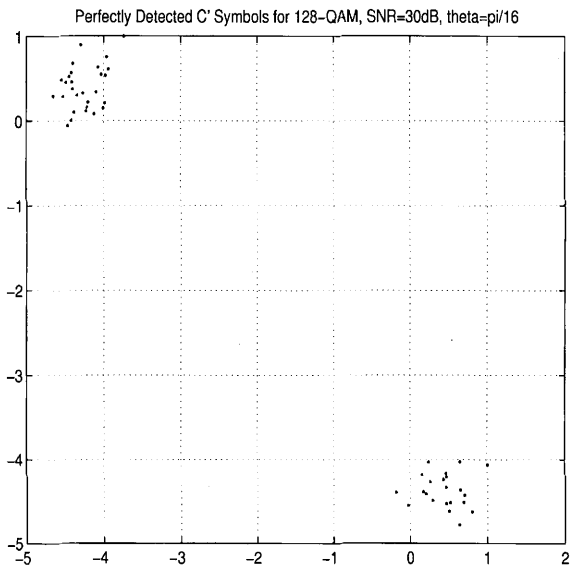


Fig. 35. The Fourth Power of Threshold Detected Data for 128-QAM, $\theta = \pi/16$

The algorithm used to perform this task is a form of the Linde-Buzo-Gray (LBG) algorithm commonly used in training vector quantizers [11]. The LBG algorithm is an iterative algorithm that effectively partitions the symbols in \mathbf{r}' into two groups such that the sum of the variance of the symbols in the groups is minimized.

Let Y_1 and Y_2 represent the location of the centroid of the two groups of points in the plane. These values can initially be set to any two distinct points, and for the program described in this work they are initially set to equal the position of the first two threshold-detected symbols. The algorithm that classifies the symbols into the groups \mathbf{r}'_1 and \mathbf{r}'_2 is as follows:

1. Place each unclassified symbol into the nearest group based on the distance to the centroids
2. Recompute the centroids based on the new groupings,
 $Y_1 = \bar{\mathbf{r}}'_1$ and $Y_2 = \bar{\mathbf{r}}'_2$
3. Reclassify all symbols into the nearest group based on the distance to the centroids
4. If any symbols changed groups, go to step 2

Figure 36 shows the results of performing this procedure on the data shown in Figure 34. Once the algorithm has converged, an estimate is produced using equation (4.2).

2. Simulation Results for the Grouping Algorithm

Simulations were run for the grouping algorithm for 1500 iterations. Figures 37 and 38 show the simulation results for 32-QAM and 128-QAM respectively. Detection errors hurt the performance of the algorithm for both constellations, but the effect of these

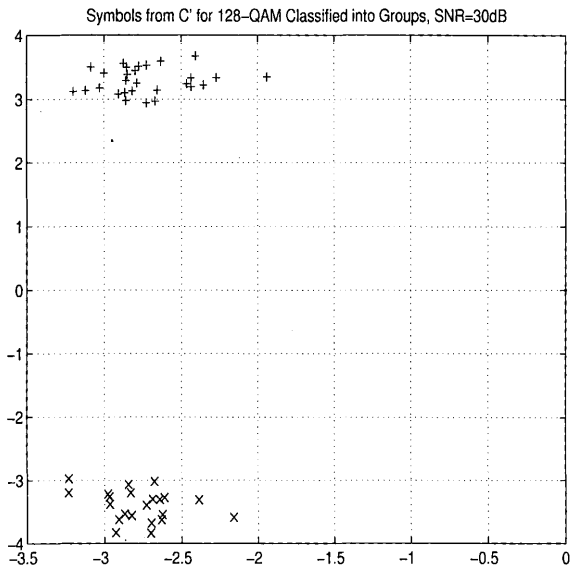


Fig. 36. Grouped Fourth Power Data for 128-QAM, $\theta = 0$

errors is more pronounced for 32-QAM. The performance for 128-QAM comes closer to the approximation. The difference between the simulation and approximation at the smaller vector lengths in Figure 38 is due to the fact that approximation does not take into account the case when symbols from only one of the subconstellations C'_i have been detected.

Even though the grouping algorithm failed to meet the expectations set forth by the approximation, it did achieve the goal of improving upon the accuracy of the PLT algorithm. Furthermore it indicates that algorithms which rely on decisions about the transmitted sequence can be feasible for phase acquisition. This type of decision based algorithm will be explored more fully in the next chapter.

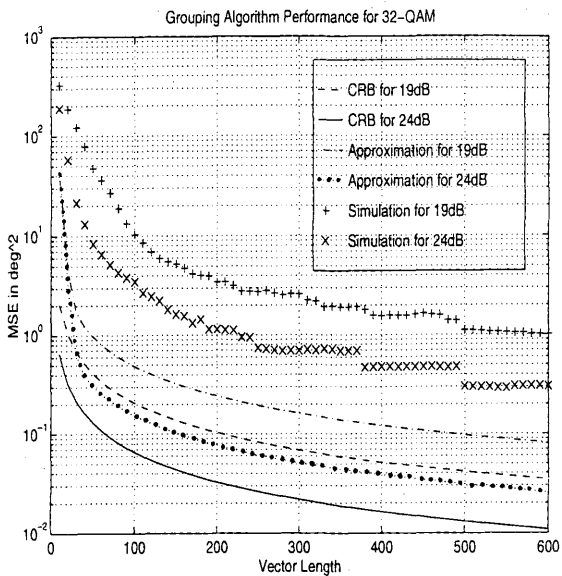


Fig. 37. Grouping Algorithm Performance for 32-QAM

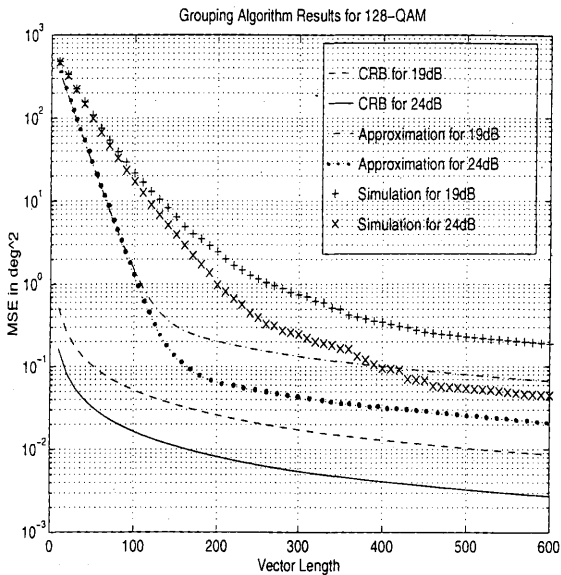


Fig. 38. Grouping Algorithm Performance for 128-QAM

CHAPTER V

TRELLIS ALGORITHM PHASE ESTIMATION

The purpose of this chapter is to evaluate the performance of an algorithm which uses the idea of a trellis and the Viterbi algorithm to produce a phase estimate. The algorithm is based on one presented by Ungerboeck in [7] for carrier phase tracking. Changes have been made to the algorithm to make it more suitable for fast carrier acquisition, and to allow its use for large constellations. It will be shown that for high SNR and large K the algorithm acts as a decision-directed estimator, which is known to perform very well for high SNR.

A. Derivation of the Trellis Algorithm

In Chapter III a derivation for the ML estimate for the case of $\gamma \rightarrow 0$ was given. Here a derivation is given for the case of $\gamma \rightarrow \infty$. Again, the ML estimate is the value of θ which maximizes the log-likelihood function

$$\max_{\theta} L(\mathbf{r} | \theta) = \sum_{k=1}^K \ln E_C \left[\exp \left(-\gamma \cdot \left| r(k)e^{-j\theta} - c_i \right|^2 \right) \right]. \quad (5.1)$$

As $\gamma \rightarrow \infty$ we have $r(k) \rightarrow m(k)e^{j\theta}$, so in the limit $r(k) = c_{j,k}e^{j\theta}$ for some $j \in \{1, 2, \dots, N\}$. All terms in $E_C[\cdot]$ for which $c_i \neq c_j$ become zero, leaving as the log-likelihood function

$$\begin{aligned} L_{\infty}(\mathbf{r} | \theta) &= -\gamma \sum_{k=1}^K \left| r(k)e^{-j\theta} - c_{j,k} \right|^2 \\ &= -\gamma \sum_{k=1}^K \left(|r(k)|^2 - 2\operatorname{Re}[r(k)c_{j,k}^*e^{-j\theta}] + |c_{j,k}|^2 \right) \\ &= 2\gamma \operatorname{Re} \left[\sum_{k=1}^K r(k)c_{j,k}^*e^{-j\theta} \right] + K_1 \end{aligned} \quad (5.2)$$

where K_1 includes all terms that do not depend on θ . This form of the log-likelihood function is maximized for

$$\hat{\theta}_\infty = \arg \left(\sum_{k=1}^K r(k) c_{j,k}^* \right). \quad (5.3)$$

In cases where the sequence $\{c_{j,k}\}$ is known, such as when a preamble is used, $\hat{\theta}_\infty$ is a good estimate, especially at higher SNR. When the transmitted sequence is unknown, however, it must be estimated. For this case the estimate becomes

$$\hat{\theta}_{DD} = \arg \left(\sum_{k=1}^K r(k) \hat{c}_k^* \right) \quad (5.4)$$

where \hat{c}_k is the estimate of the transmitted symbol for the k -th interval. This is the decision-directed phase estimate which is commonly used for carrier phase tracking after acquisition. When the phase offset is completely unknown, symbol-by-symbol estimation of the transmitted data is impossible. Under certain conditions, however, sequence decisions can be made well enough to produce an accurate phase estimate.

Consider the problem of joint estimation of $\{\hat{c}_k\}$ and $\hat{\theta}$. Let $\{\hat{\theta}_k\}$ be a sequence of phase estimates. From Bayes theorem we have

$$f(\{\hat{c}_k\}, \{\hat{\theta}_k\} | r(k)) \propto f(r(k) | \{\hat{c}_k\}, \{\hat{\theta}_k\}), \quad (5.5)$$

since all sequences $\{\hat{c}_k\}$ are equally likely and $\{\hat{\theta}_k\}$ is assumed to be uniformly distributed. The log-likelihood function to be maximized is given by

$$L(r(k) | \{\hat{c}_k\}, \{\hat{\theta}_k\}) = -\gamma \sum_{k=1}^K |r(k) e^{-j\hat{\theta}_k} - \hat{c}_k|^2. \quad (5.6)$$

Hence to maximize the log-likelihood function we want to minimize the metric

$$J(\{\hat{c}_k\}, \{\hat{\theta}_k\}) = \sum_{k=1}^K |r(k) e^{-j\hat{\theta}_k} - \hat{c}_k|^2. \quad (5.7)$$

In this work the Viterbi algorithm is used to attempt to produce the sequence of

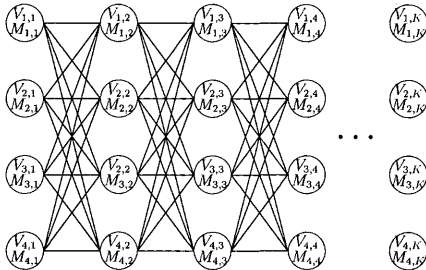


Fig. 39. A Four State Fully Interconnected Trellis

estimates $\{\hat{c}_k\}$, and the sequence $\{\hat{\theta}_k\}$ is produced using equation (5.4).

Let each symbol c_s in the constellation C correspond to a state I_s in a trellis. I_s at the k -th level of the trellis will be referred to as $I_{s,k}$. Since the symbols are i.i.d. every state in the trellis is connected to every state in the next level, as shown in Figure 39 for a simplified four state case. Let each state be associated with a vector V and a metric M . For trellis state $I_{s,k}$ these variables have values of $V_{s,k}$ and $M_{s,k}$.

The algorithm is initialized using the first received symbol $r(1)$. For each state $I_{s,1}$ the variables become

$$\begin{aligned} V_{s,1} &= r(1)c_s^* \\ M_{s,1} &= (|r(1)| - |c_s|)^2. \end{aligned} \quad (5.8)$$

Note that the optimal high SNR estimate given in (5.4) based on only the first received symbol is given by $\hat{\theta}_{s,1} = \arg(V_{s,1})$ for some state $I_{s,1}$. Also, the state corresponding to the correct estimate will most likely have the smallest metric $M_{s,1}$. Hence we choose

$\hat{\theta}_1 = \hat{\theta}_{s,1}$ for the state with the smallest metric $M_{s,1}$.

For all other k the values of $V_{s,k}$ and $M_{s,k}$ depend on the previous level in the trellis as well as the data $r(k)$. We define the metric for the transition between state $I_{s,k-1}$ to state $I_{t,k}$ as

$$d(r(k), c_t | \hat{\theta}_{s,k-1}) = |r(k)e^{-j\hat{\theta}_{s,k-1}} - c_t|^2. \quad (5.9)$$

This metric is used to determine the survivor paths through the trellis. The total path metric of a path entering state $I_{t,k}$ from state $I_{s,k-1}$ is given by

$$D(I_{s,k-1}, I_{t,k}) = M_{s,k-1} + d(r(k), c_t | \hat{\theta}_{s,k-1}). \quad (5.10)$$

Each state $I_{t,k}$ chooses as the survivor the incoming path for which the total path metric is minimized.

If we let $I_{m,k-1}$ represent the state for which $D(I_{m,k-1}, I_{t,k})$ is the minimum incoming path metric, the values assigned to V and M are

$$\begin{aligned} V_{t,k} &= V_{m,k-1} + r(k)c_t^* \\ M_{t,k} &= D(I_{m,k-1}, I_{t,k}). \end{aligned} \quad (5.11)$$

The phase estimate at time k is given by $\hat{\theta}_k = \hat{\theta}_{s,k}$ for the state $I_{s,k}$ with the smallest metric $M_{s,k}$.

Assume that at time K the state $I_{m,K}$ has the smallest total path metric $M_{m,K}$. The phase estimate at this time can be written as

$$\hat{\theta}_K = \arg \left(\sum_{k=1}^K r(k) \hat{c}_k^* \right) \quad (5.12)$$

where \hat{c}_k represents the values of c_k for the nodes $I_{s,k}$ that the minimum metric path

passed through. The value of the path metric at this node is given by

$$M_{m,K} = \sum_{k=1}^K |r(k)e^{-j\hat{\theta}_k} - \hat{c}_k|^2 \quad (5.13)$$

$$= J(\{\hat{c}_k\}, \{\hat{\theta}_k\}).$$

Hence if the Viterbi algorithm produces the correct sequence of estimates \hat{c}_k , the appropriate metric is minimized and $\hat{\theta}_K = \hat{\theta}_{DD}$.

1. Reducing Computational Load for the Trellis Algorithm

A constellation with order N will give rise to a trellis with N states. If a trellis has N states then the transition from one level to the next requires N^2 calculations of the metric given in equation (5.10), as well as N^2 comparisons to determine the survivor paths. Furthermore the $\pi/2$ phase ambiguity of the constellations results in redundancy in the trellis states. For every symbol $c_s \in C$ there are states which correspond to symbols $c_s e^{j\frac{m\pi}{2}}$ for $m \in \{0, 1, 2, 3\}$. Since received a sequence $\{r(1), r(2), \dots\}$ is indistinguishable from a sequence $\{r(1)e^{j\frac{m\pi}{2}}, r(2)e^{j\frac{m\pi}{2}}, \dots\}$ the trellis will always have at least four states with identical metrics $M_{s,k}$, corresponding to the estimates $\hat{\theta}_k + \frac{m\pi}{2}$. Only one estimate is needed, so calculating four is inefficient.

To eliminate some of the redundancy we define the constellation $Q = \{q_i\}$ to be the symbols from C which lie in the first quadrant. If we make the assumption that $\theta \in [0, \pi/2)$ then we can think of the received sequence in terms of the symbols q_i .

$$r(k) = q_i e^{j(\theta + \frac{m\pi}{2})} \quad \text{for some } i \text{ and } m. \quad (5.14)$$

The received symbols are mapped into the first quadrant by taking the angle modulus $\pi/2$ using the transform

$$r_1(k) = |r(k)| e^{j(\arg(r(k)) \bmod \frac{\pi}{2})}. \quad (5.15)$$

The trellis algorithm can now be applied using the smaller constellation Q and the transformed received data $\mathbf{r}_1 = \{r_1(k)\}$, with modifications to the distance metric to account for the effects of the modulus.

Since the state $I_{s,k-1}$ for the modified algorithm corresponds to four states in the original algorithm, the state $I_{t,k}$ must consider four possible metrics when choosing the incoming path. The metric for the path entering $I_{t,k}$ from $I_{s,k-1}$ is

$$D'(I_{s,k-1}, I_{t,k}) = M_{s,k-1} + \min_{m \in \{0,1,2,3\}} \left[d(r(k), q_t \mid \hat{\theta}_{s,k-1} + \frac{m\pi}{2}) \right]. \quad (5.16)$$

The case for $m = 2$ will never produce the minimum metric and does not need to be considered, and a value of $m = -1$ is equivalent to $m = 3$, so the metric can now be written as

$$D'(I_{s,k-1}, I_{t,k}) = M_{s,k-1} + \min_{m \in \{-1,0,1\}} \left[d(r(k), q_t \mid \hat{\theta}_{s,k-1} + \frac{m\pi}{2}) \right]. \quad (5.17)$$

The three possible incoming metrics correspond to three vectors $r_1(k)e^{j(\hat{\theta}_{s,k-1} + \frac{m\pi}{2})}$. If \hat{m} is the value of m that minimizes $D'(I_{s,k-1}, I_{t,k})$ the values assigned to the variables are

$$\begin{aligned} V_{t,k} &= V_{s,k-1} + r_1(k)e^{-j(\hat{\theta}_{s,k-1} + \frac{\hat{m}\pi}{2})} q_t^* \\ M_{t,k} &= D'(I_{s,k-1}, I_{t,k}). \end{aligned} \quad (5.18)$$

This modified algorithm produces an estimate with the same accuracy as before with a significant reduction in the computational load. It requires the calculation of three metrics for each state, and the trellis contains $N/4$ states, therefore $\frac{3}{16}N^2$ metrics and comparisons are required for each transition between levels. The computational load can still be considerable for large constellations, however. For example, $\frac{3}{16}(256)^2 = 12288$ metrics to compute for every level in the trellis.

2. Thresholding for the Trellis Algorithm

It will be shown in the results that the algorithm performs poorly without further modifications. Decision errors early in the trellis can cause the path corresponding to incorrect decisions to be chosen as the survivor path instead of the correct path. When this happens it is unlikely that the algorithm will recover, and the estimate produced will not be accurate.

Better performance can be achieved with the use of a threshold. As in chapter III we introduce a threshold T and form the vector \mathbf{r}' which consists of all $r(k)$ for which $|r(k)| > T$. This allows the use of a subset of the constellation Q , resulting in further reduction in the computational load. Unlike in chapter III, however, we assume that some detection errors will be made, and design the algorithm to use this incorrectly detected data in the estimate. To do this we define Q' to be the symbols $\{q_i \in Q\}$ such that $|q_i| > T - \frac{d_m}{2}$ where d_m is the minimum distance between symbols in Q . This choice of Q' places an upper bound on the probability that a symbol in \mathbf{r}' does not correspond to a state in the trellis. For the lower SNRs considered in this work the probability of this type of thresholding error will be less than about 10^{-2} . If we let $N_{Q'}$ represent the number of symbols in Q' , then the algorithm requires $\frac{3}{16}N_{Q'}^2$ metrics and comparisons for each transition between levels. Also fewer transitions are required since the algorithm only moves to the next level when a symbol from \mathbf{r}' is detected.

An approximation to the performance of this algorithm can be derived in a similar way to equation (3.14) based on the CRB. Assuming that there are no unmanageable detection errors and no symbol decision errors the resulting approximation for the

thresholded trellis algorithm is given by

$$\sigma_{\theta_{TT}}^2 \approx \left(1 - \left(\frac{N - N_T}{N}\right)^K\right) \frac{N}{N_T K} \left(\frac{1}{2\gamma E[|c|^2]}\right) + \left(\frac{N - N_T}{N}\right)^K \frac{\pi^2}{48} \quad (5.19)$$

where N_T is the number of symbols in C whose magnitude is greater than T .

3. Results for the Trellis Algorithm

Simulations for the trellis algorithm were run for 10,000 iterations. Figures 40 and 41 show the simulation results and the approximation $\sigma_{\theta_{TT}}^2$ versus T for 32-QAM. Figure 40 shows that the algorithm performs poorly at most threshold values for the lower SNR's, as well as some lower threshold values for the higher SNR's. This poor performance is due mainly to decision errors. There is a threshold value of about $T = 1.30$ at which the performance for both the low and high SNR is acceptable. Figure 41 shows the performance versus K at $T = 1.30$.

Figures 42 and 43 show similar graphs for 64-QAM. From Figure 42 we see that in this case there is a choice of acceptable thresholds. Letting $T = 1.2$ results in a trellis with eight states. Letting $T = 1.3$ results in a trellis with five states, but also results in a small performance loss. Choosing the higher threshold reduces the amount of computation required to produce the estimate by about 60 percent. In cases like this the higher threshold is preferred.

Figures 44 and 45 show the simulation results and $\sigma_{\theta_{TT}}^2$ for 128-QAM. Figure 44 indicates that an appropriate choice for the threshold is a value of about $T = 1.30$, which results in a ten-state trellis.

Figures 46 and 47 show the simulation results and $\sigma_{\theta_{TT}}^2$ for 256-QAM. Figure 46 indicates that an appropriate choice for the threshold is a value of about $T = 1.40$, which results in a four-state trellis.

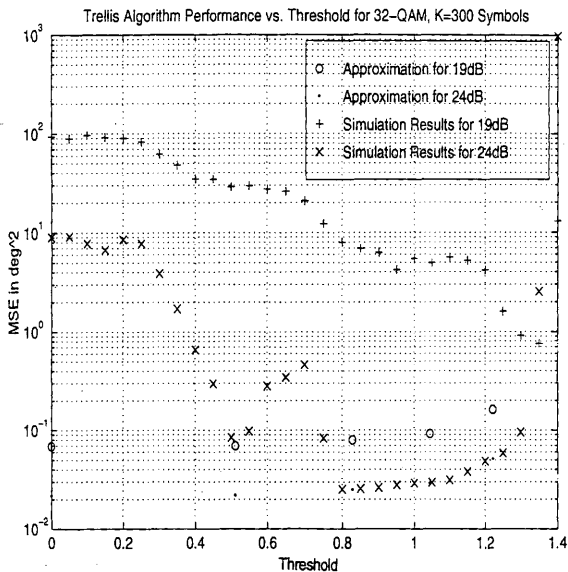


Fig. 40. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs T for 32-QAM, $K = 300$ Symbols

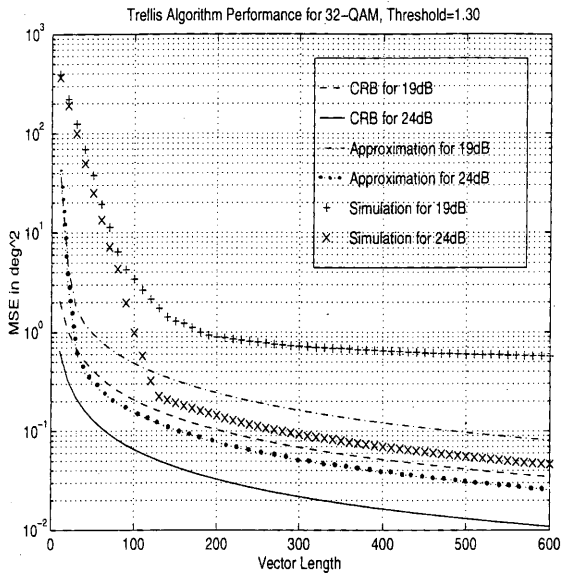


Fig. 41. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs K for 32-QAM

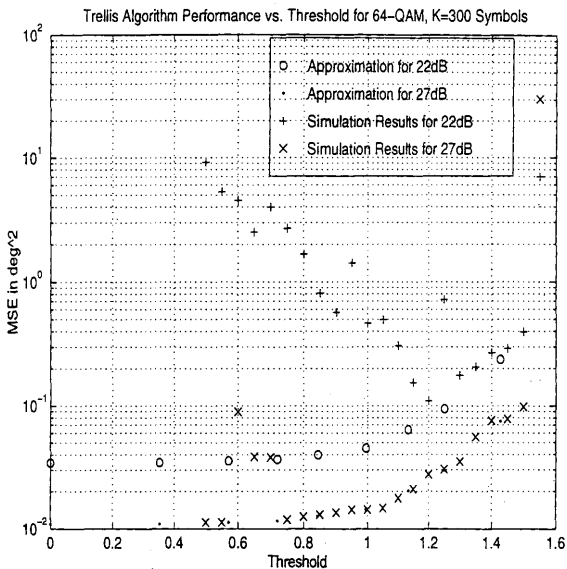


Fig. 42. Trellis Algorithm Performance and $\sigma_{\hat{\theta}_{TT}}^2$ vs T for 64-QAM, $K = 300$ Symbols

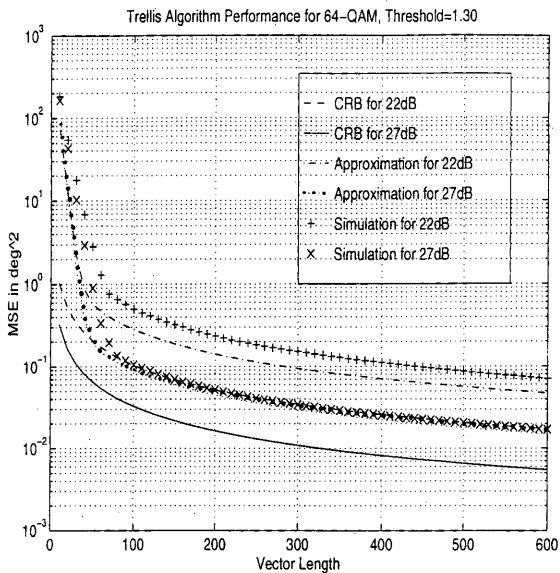


Fig. 43. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs K for 64-QAM

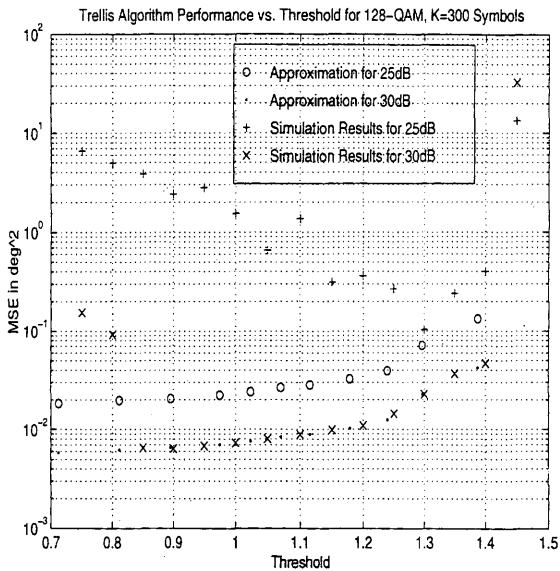


Fig. 44. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs T for 128-QAM, $K = 300$ Symbols

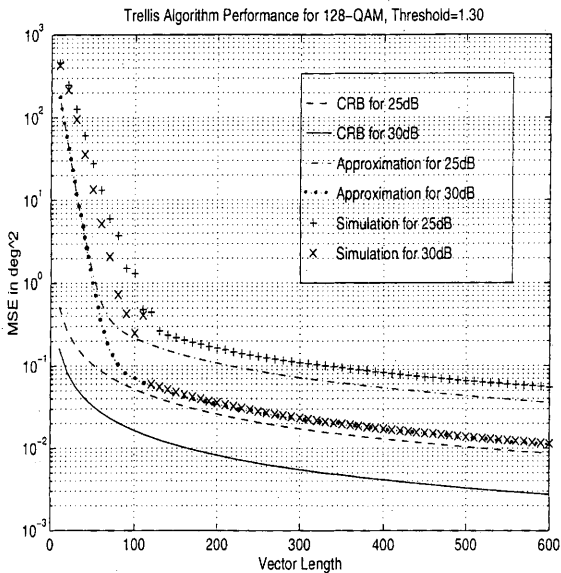


Fig. 45. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs K for 128-QAM

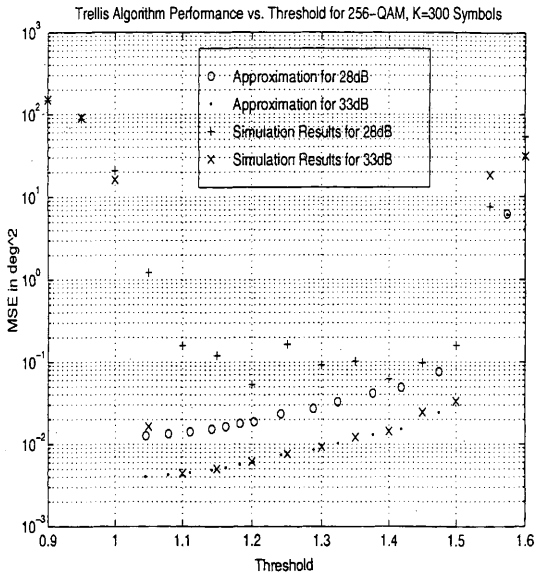


Fig. 46. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs T for 256-QAM, $K = 300$ Symbols

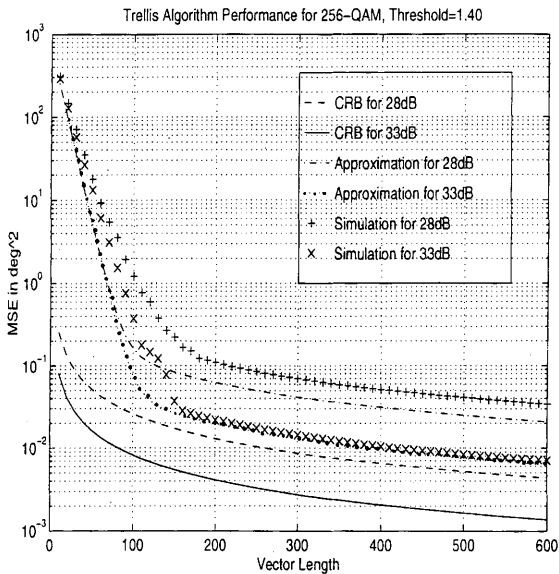


Fig. 47. Trellis Algorithm Performance and $\sigma_{\theta_{TT}}^2$ vs K for 256-QAM

B. A Suboptimal Version of the Trellis Algorithm

A further reduction in complexity can be achieved with a small loss in performance if we apply the trellis algorithm to the fourth power of the received data. Let $Q_4 = \{q_i^4\}$ and $\mathbf{r}_4 = \{r(k)^4\}$. The algorithm can be applied using the simpler metric given in equation (5.10), with the variables M and V taking values described in (5.11). The estimate for each state $I_{s,k}$ is given by

$$\hat{\theta}_{s,k} = \frac{1}{4} \arg(V_{s,k}). \quad (5.20)$$

Thresholding is again applied to reduce complexity and achieve better performance. The use of the euclidean distance metric on the transformed data results in a higher probability of decision errors, which causes some degradation in performance. However, since the simpler metric can be used this algorithm requires about one third of the computation to produce an estimate.

In the limit of high SNR, the estimate given by the path with the minimum metric is

$$\begin{aligned} \hat{\theta}_K &= \frac{1}{4} \arg\left(\sum_{k=1}^K r(k)^4 c_k^{4*}\right) \\ &= \frac{1}{4} \arg\left(\sum_{k=1}^K (c_k e^{j\theta})^4 c_k^{4*}\right) \\ &= \frac{1}{4} \arg\left(e^{j4\theta} \sum_{k=1}^K |c_k|^8\right) \\ &= \theta. \end{aligned} \quad (5.21)$$

Hence in the limit of high SNR the algorithm produces a perfect estimate.

1. Results for the Suboptimal Trellis Algorithm

Simulations for the suboptimal trellis algorithm were run for 10,000 iterations. The results are shown in Figures 48 through 55. Figures 48, 50, 52, and 54 show the performance of the suboptimal trellis algorithm versus T for the four constellations. A value of $T = 1.325$ was chosen for 32-QAM to provide better overall performance, still resulting in a four-state trellis. The other threshold values did not change from before. Figures 49, 51, 53, and 55 show the performance versus K for the four constellations at the appropriate values of T .

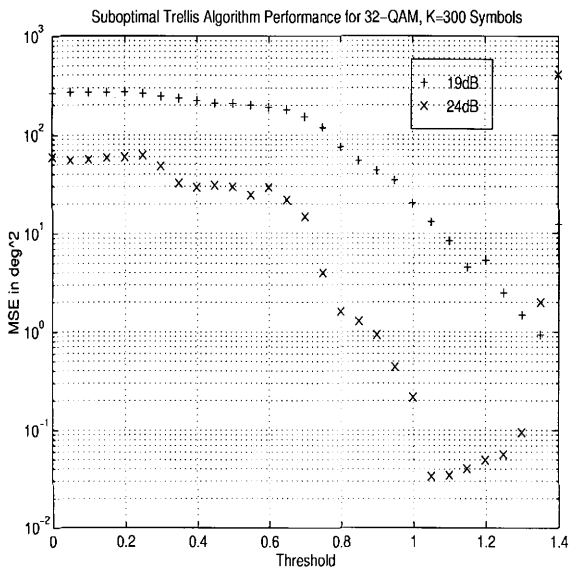


Fig. 48. Suboptimal Trellis Algorithm Performance vs T for 32-QAM, $K = 300$ Symbols

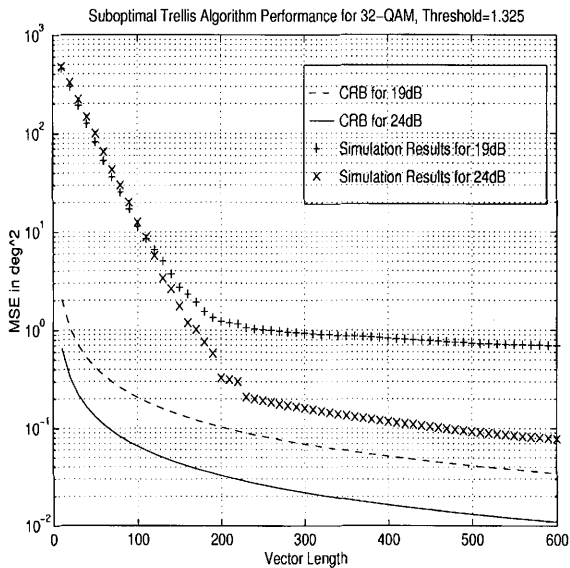


Fig. 49. Suboptimal Trellis Algorithm Performance vs K for 32-QAM

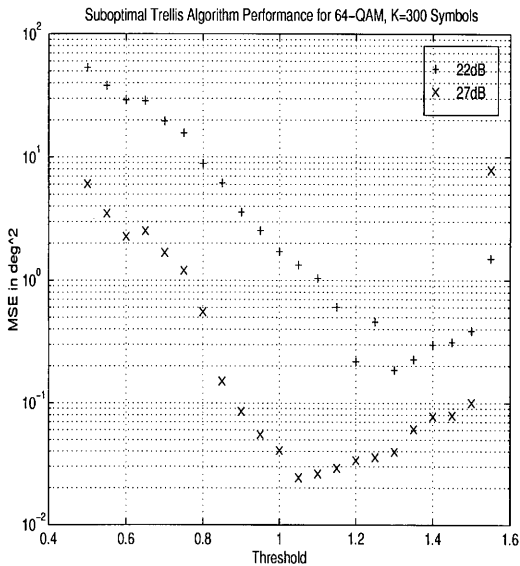


Fig. 50. Suboptimal Trellis Algorithm Performance vs T for 64-QAM, $K = 300$ Symbols

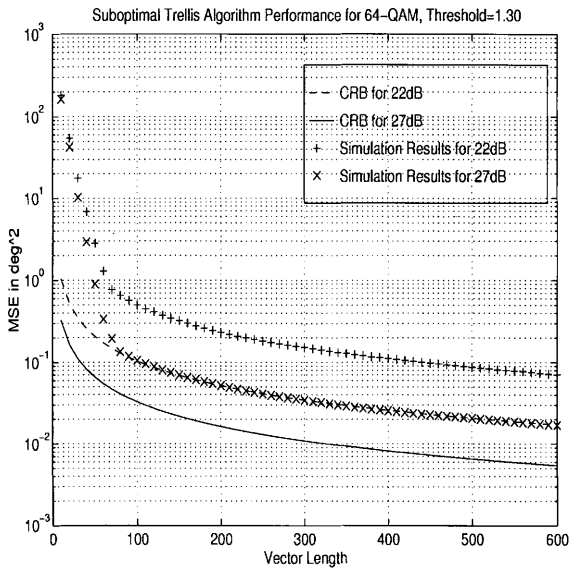


Fig. 51. Suboptimal Trellis Algorithm Performance vs K for 64-QAM

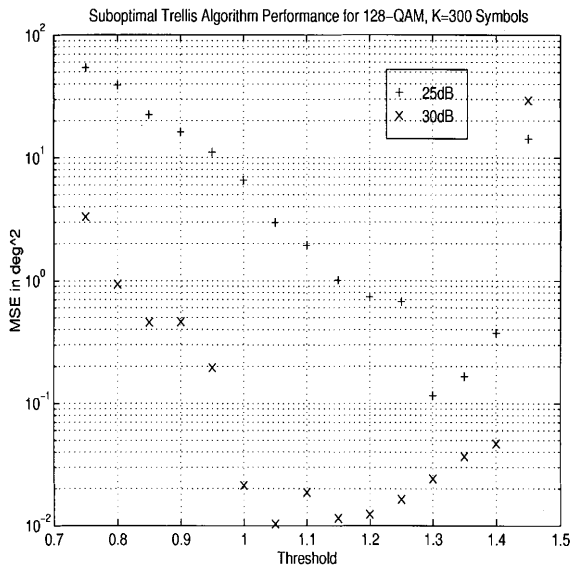


Fig. 52. Suboptimal Trellis Algorithm Performance vs T for 128-QAM, $K = 300$ Symbols

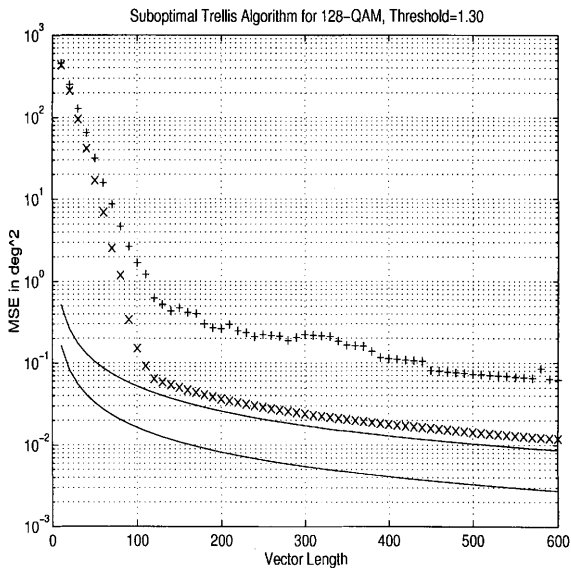


Fig. 53. Suboptimal Trellis Algorithm Performance vs K for 128-QAM

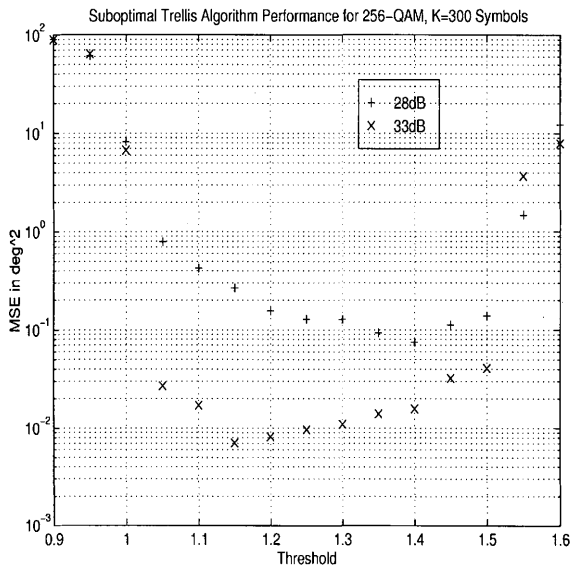


Fig. 54. Suboptimal Trellis Algorithm Performance vs T for 256-QAM, $K = 300$ Symbols

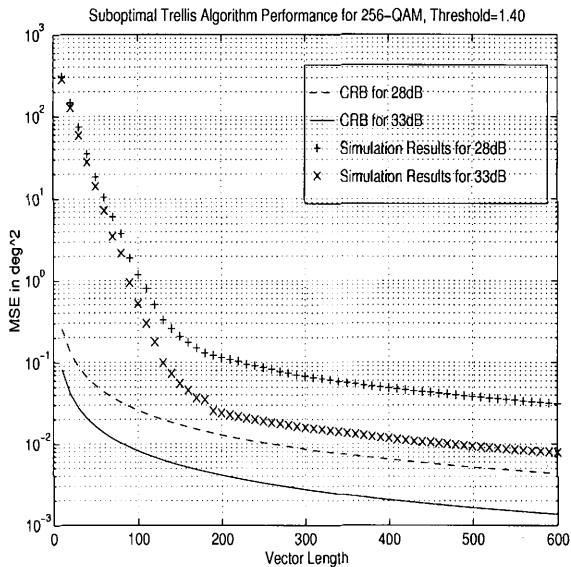


Fig. 55. Suboptimal Trellis Algorithm Performance vs K for 256-QAM

CHAPTER VI

CONCLUSION

Seven carrier-phase estimation algorithms for QAM constellations have been derived and evaluated in this work. Most of them were found to be inadequate in some way for the purpose of rapid carrier-phase acquisition.

The ML algorithm discussed in Chapter II was found to perform well relative to the CRB. The complexity of the algorithm, however, makes it an impractical solution for virtually all applications requiring the use of large QAM signal sets. The need to estimate the SNR is another drawback. In an effort to produce a less complex estimate using ML techniques, the SML algorithm was derived. It was found that the SML algorithm could also produce a good estimate, but was still too complex to be useful.

Chapter III discussed the PL estimator, which is obtained by maximizing the low SNR limit of the log-likelihood function. This estimate is easily computed, but was found to have very poor performance relative to the CRB. The PLT algorithm used a threshold along with the PL algorithm to produce an estimate using a subset of the received data. For all of the constellations considered, the error performance of the PLT estimate was better than that of the PL estimate for an appropriate threshold value. However, in most cases the performance was still not good enough to make the PLT algorithm useful. The exception is the case of the 64-QAM constellation, for which the mean-squared error performance of the PLT estimate comes within one to two orders of magnitude of the CRB, for a vector length of about 150 symbols.

Chapter IV briefly described an algorithm which is designed for use with cross constellations. It uses the LBG algorithm to attempt to make decisions on a thresholded subset of the received symbols. Although this algorithm performed better

than the PLT algorithm, its reliance on the iterative LBG algorithm makes it less attractive.

In Chapter V an algorithm based on the idea of a trellis was discussed. This algorithm used the Viterbi algorithm to make decisions on a thresholded subset of the received symbols, which allows a simple estimate to be produced. This algorithm was shown to perform well for all constellations, with appropriate choice of threshold values. A modified version of the trellis algorithm is presented, which performs operations on the fourth power of the received symbols. This allows use of a simpler metric and reduces the computation required by two-thirds. Both the trellis and modified trellis algorithms were found to be useful and practical for rapid carrier-phase estimation for all of the constellations considered in this work.

REFERENCES

- [1] M. P. Fitz and W. C. Lindsey, "Decision Directed Burst-Mode Carrier Synchronization Techniques," *IEEE Transactions on Communication*, vol. 40, pp. 1644-1653, October 1992.
- [2] J. G. Proakis, *Digital Communications*. New York, New York: McGraw Hill, second ed., 1989.
- [3] A. J. Viterbi and A. M. Viterbi, "Nonlinear Estimation of PSK-Modulated Carrier Phase with Application to Burst Digital Transmission," *IEEE Transactions on Information Theory*, vol. IT-29, pp. 543-551, July 1983.
- [4] P. Y. Kam, "Maximum Likelihood Carrier Phase Recovery for Linear Suppressed-Carrier Digital Data Modulations," *IEEE Transactions on Communications*, vol. COM-34, pp. 522-527, June 1986.
- [5] P. Y. Kam, "Maximum-Likelihood Digital Data Sequence Estimation Over the Gaussian Channel with Unknown Carrier Phase," *IEEE Transactions on Communications*, vol. COM-35, pp. 764-767, July 1987.
- [6] H. Kobayashi, "Simultaneous Adaptive Estimation and Decision Algorithm for Carrier Modulated Data Transmission Systems," *IEEE Transactions on Communication Technology*, vol. COM-19, pp. 268-280, June 1971.
- [7] G. Ungerboeck, "New Applications for the Viterbi Algorithm: Carrier Phase Tracking in Synchronous Data-Transmission Systems," *Proceedings of the IEEE National Telecommunication Conference*, pp. 734-738, December 1974.

- [8] O. Macchi and L. L. Scharf, "A Dynamic Programming Algorithm for Phase Estimation and Data Decoding on Random Phase Channels," *IEEE Transactions on Information Theory*, vol. IT-27, pp. 581–595, September 1981.
- [9] M. Moeneclaey and G. de Jonghe, "ML-Oriented NDA Carrier Synchronization for General Rotationally Symmetric Signal Constellations," *IEEE Transactions on Communications*, vol. 42, pp. 2531–2533, August 1994.
- [10] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York, New York: McGraw Hill, third ed., 1991.
- [11] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.
- [12] L. E. Franks, *Signal Theory*. Stroudsburg, PA: Dowden & Culver, revised ed., 1981.
- [13] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, 1983.
- [14] L. W. Couch, *Digital and Analog Communication Systems*. New York, New York: McMillan Publishing Company, fourth ed., 1993.
- [15] N. K. Jablon, "Joint Blind Equalization, Carrier Recovery, and Timing Recovery for High-Order QAM Signal Constellations," *IEEE Transactions on Signal Processing*, vol. COM-28, pp. 1867–1875, November 1980.
- [16] D. N. Godard, "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems," *IEEE Transactions on Communication*, vol. 40, pp. 1644–1653, October 1992.

VITA

Mr. Jason Dillard Preston was born on November 16, 1970 in Port Arthur, Texas. He received his B. S. in Electrical Engineering from Lamar University in 1993, and his B. S. in Mathematics from Lamar University in 1994. He currently resides at 4105 Alexandria Drive, Austin TX.