Spectrum Sensing and Blind Automatic Modulation Classification in Real Time

Michael Paul Steiner

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Electrical and Computer Engineering

Tamal Bose

Jeffrey Reed

S. M. Hasan

April 28, 2011

Blacksburg, Virginia

Keywords: modulation classification; spectrum sensing; real time; blind synchronization

Spectrum Sensing and Blind Automatic Modulation Classification in Real-Time

Michael Steiner

ABSTRACT

This paper describes the implementation of a scanning signal detector and automatic modulation classification system. The classification technique is a completely blind method, with no prior knowledge of the signal's center frequency, bandwidth, or symbol rate. An energy detector forms the initial approximations of the signal parameters. The energy detector used in the wideband sweep is reused to obtain fine estimates of the center frequency and bandwidth of the signal. The subsequent steps reduce the effect of frequency offset and sample timing error, resulting in a constellation of the modulation of interest. The cumulant of the constellation is compared to a set of known ideal cumulant values, forming the classification estimate.

The algorithm uses two platforms that together provide high speed parallel processing and flexible run-time operation. High-rate spectral scanning using an energy detector is run in parallel with a variable down sampling path; both are highly pipelined structures, which allows for high data throughput. A pair of processing cores is used to record spectral usage and signal characteristics as well as perform the actual classification.

The resulting classification system can accurately identify modulations below 5 dB of signal-to-noise ratio (SNR) for some cases of the phase shift keying family of modulations but requires a much higher SNR to accurately classify higher-order modulations. These estimates tend toward classifying all signals as binary phase shift keying because of limits of the noise power estimation part of the cumulant normalization process. Other effects due to frequency offset and synchronization timing are discussed.

Dedication

To my Mother, for fixing my atrocious grammar in the early morning.

To my Father, for teaching me at an early age the important question of "why?"

To Gautham, for listening to my nonsensical rantings about compile times and noise power estimation.

Acknowledgment

I thank National Instruments for their support throughout my research. They provided much of the instrumentation: an NI interposer board, an NI 5781 Adapter Module, an NI 7695 FPGA Module, and an NI 8130 Real-Time Module. Each piece of hardware was interfaced through NI's LabVIEW software. Also, I appreciate the financial support provided by National Instruments.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AM	amplitude modulated signal
AMC	automatic modulation classification
ASK	amplitude shift keying
AWGN	additive Gaussian white noise
BPSK	binary phase shift keying
$\widetilde{\boldsymbol{C}}_{\boldsymbol{nk}}$	calculated feature for each modulation
$\overline{\boldsymbol{C}}_{\boldsymbol{nk}}$	stored ideal value for each modulation
$\boldsymbol{C}_{\boldsymbol{p(p\text{-}q)}}$	cumulant with p equal to the cumulant order and q equal to the number of non-conjugated inputs to the cumulant function
CNR	carrier-to-noise ratio
Δ	distance factor
$\boldsymbol{D}$	decimation factor
$\overline{\boldsymbol{D}}_{\boldsymbol{nk}}$	linear distance between the calculated feature and stored ideal value
$\overline{\boldsymbol{D}}_{\boldsymbol{T}}$	total distance from each modulation
DFT	discrete Fourier Transform
DPSK	differential phase shift keying
DQPSK	differential quadrature phase shift keying
$\hat{\boldsymbol{\varepsilon}}$	timing error
FFT	fast Fourier Transform
FIFO	first in–first out queue
FIR	finite impulse response
FM	frequency modulated signal
FPGA	field programmable gate array
FSK	frequency shift keying
$\boldsymbol{h}$	polynomial coefficient
H-T	host-to-target
$\boldsymbol{I}$	interpolation factor
ISI	intersymbol interference
ISM	industrial scientific and medical
LL	log-likelihood
M-ASK	M-ary amplitude shift keying
ML	maximum likelihood
M-PSK	M-ary phase shift keying
M-QAM	M-ary quadrature amplitude modulation
MSK	minimum shift keying
$\boldsymbol{n}$	number of multiplications
OFDM	orthogonal frequency division multiplexing
OOK	on–off keying
$\boldsymbol{P}_{\mathbf{d}}$	probability of correct detection
$\boldsymbol{P}_{\mathbf{fa}}$	probability of false alarm
PAM	pulse amplitude modulation

PLD	phase-locked detector
PLL	phase-locked loop
PSK	phase shift keying
QAM	quadrature amplitude modulation
qLLR	quasi-log-likelihood ratio
QPSK	quadrature phase shift keying
RAM	random access memory
RF	radio frequency
RRC	root-raised cosine
SNR	signal-to-noise ratio
SPS	samples per symbol
T	symbol period
x_Δ	time duration between input sample point and desired resampled instant

Chapter 1

Introduction

1.1 Motivations for Research

Communication systems are moving toward smarter and more adaptive technologies in an attempt to improve user performance in terms of minimizing transmission power, minimizing bandwidth usage, and increasing reliability. To that end, systems are being developed which have flexible transmission characteristics, such as output power, encoding scheme, and modulation type. While improving performance, these devices create a more dynamic spectrum which will become increasingly difficult to navigate as more devices are added. In addition, legacy systems still exist and are likely to continue for the duration of the foreseeable future. To this effect, the ability to automatically and blindly determine modulation types allow low-overhead ad hoc communication links and identification and avoidance of primary users.

Automatic modulation classification (AMC) is not a new topic of discussion; this method has been of interest since analog modulations were the primary method of wireless communications. Entrance into the digital realm has created a plethora of new modulating techniques, each suitable for different environmental conditions and throughputs. In this aspect, AMC has become both increasingly necessary and more difficult.

Most AMC systems implemented to date have been for a specific purpose, such as identifying the modulation of a satellite link which adapts to channel conditions [1]. The advance of smart and cognitive radios has created a need for more wideband spectrum awareness. This situation drastically increases the complexity of the classification problem, as now there are almost no limits on the types of signals that may be of interest. This paper describes a method of scanning and recording signal information over a wide (100-MHz) bandwidth. As a fully realized system, many practical issues are

unavoidable, such as frequency offset due to the limited a priori knowledge and non-Gaussian noise due to the existence of other signals, particularly spread spectrum signals, in the band. As a result, a much more complex problem is created than the underlying classification algorithm upon which the system is based.

This paper looks at the expected results based on the ideal theoretical approach and then moves into the design and implementation of a practical system that performs the signal classification in real time. The design details two main processes: scanning the spectrum to detect signals; gathering and preparing samples to be used in the classifier. Other aspects, including tracking signals over time and the classification process itself, are also discussed.

Part of what makes this classification system new is the platform upon which it is supported. Gathering and processing data in real-time can be an exhausting task, in terms of both processing power and initial programming. The hardware platform selected for this application includes both a Vertex-5 field programmable gate array (FPGA) and a dual-core real-time processor. These allow for a high degree of determinism and parallel data processing. In addition, the LabVIEW programming software is used to develop the algorithms for the hardware across the platform. Because of the innate graphical design method employed by the software, utilization of the parallel processing capabilities, particularly on the FPGA, was simple.

1.2 Thesis Overview

In the following chapter, a background in AMC techniques is discussed, with a section devoted to the study of cumulants as used in classification. Chapter 3 demonstrates several of the issues likely to be encountered when implementing a classification system. This chapter not only echoes some of the work mentioned in Chapter 2 but also forms a basis for discussion and explanation of design choices in Chapter 4. Chapter 4 details the implementation of the system. First, the hardware is discussed in terms of benefits and limitations. Second, how the signal is expected to move as it navigates through the scanning and classification algorithms is discussed. Third, in addition to the receiver structure, the design of the transmitter, which was needed for testing purposes, is

described. Chapter 5 details the testing methodology and results for the system. Tests are broken down between the two halves of the system: the classifier is tested in a manner similar to the theoretical work performed in Chapters 2 and 3; the characteristics of the wideband scanner and signal feature estimation algorithms are determined. Chapter 6 provides a summary of the design and reiterates the findings about the performance of the classification system.

Chapter 2

Literature Review

2.1 Automatic Modulation Classification (AMC)

The problem of signal classification has been extensively studied for several decades. Early interest focused on analog signals and basic digital techniques, such as the on–off keying (OOK) used to formulate Morse code [2] [3]. The simple scheme proposed by [3] used the ratio of the variance to the mean of the signal envelope to distinguish among double-side band amplitude-modulated signals (AMs), single-side band AMs, and frequency-modulated signals (FMs) effectively at carrier-to-noise ratios (CNRs) of 12 dB or higher. The prevalence of digital modulations in the current era of communications has led to more sophisticated and creative techniques of signal classification.

Most current classification techniques can be separated into two categories: decision theoretic and pattern recognition [4] [5] [6] [7] [8] [9] [10]. Decision-theoretic methods use statistical properties of the signal type to form an estimate based on the maximum likelihood (ML) of a given modulation. These methods tend to suffer from phase, frequency, and symbol timing errors common in asynchronous environments and are more difficult to implement [5]. However, decision-theoretic methods are optimal if the conditions for these numerous assumptions are met [11]. Pattern recognition methods are often used due to the simplicity of implementation with, in many cases, only a small degradation in performance compared to the optimal case [11].

In the following sections, much of the work that has been done in AMC is presented. The sections are divided into decision-theoretic and feature-based recognition methods. The work involving cumulants and related classifiers is segregated to a section of its own where it is inspected in greater detail due to its relevance to this study.

2.1.1 Decision-Theoretic Methods

The authors of [12] developed a ML-based classifier for two sets of 16-point quadrature amplitude modulations (16-QAM) modulations. The log-likelihood (LL) function is determined by using the probability of the set of received symbols matching the symbols of a particular constellation. In the simulations performed, the pulse shape was assumed to be square, with ideal frequency and timing recovery. With these assumptions, good classification (>95% correct) was achieved at signal-to-noise ratios (SNRs) ranging from 3 to 7dB depending upon the number of samples used for the estimate, which ranged from 100 to 1000. Of particular note is the derivation of correct classification with a various number of samples at arbitrary SNRs. Their work shows that, as the number of sample taken for an estimate increases to infinity, the probability of correct classification tends toward 1 regardless of the SNR.

The ML approach is continued in [13] where it is extended to the classification of orthogonal frequency division multiplexing (OFDM). The classification model is used to improve the throughput in a time division duplex system between two nodes. The AMC allows each node to form a bit allocation table based on the perceived modulation and a training sequence. The algorithm assumes a synchronous environment, as well as a few parameters particular to the time division duplex arrangement of the OFDM setup, including the signal center frequency and bit rate, and was able to exploit reciprocity of the communications link. A Doppler effect of up to 10 Hz was added in simulation and had a negligible effect on performance. The conclusion of the effort shows that AMC techniques are applicable to eliminate the need to signal the bit allocation table across the OFDM link.

Comparison of the ML method between the coherent and non-coherent cases is undertaken in [14]. The approach is almost identical with that of [12] but with the addition of phase shift keying (PSK) signals and higher order QAMs. The distinction in performance between the coherent and non-coherent cases was found to be approximately 3 dB with an error rate of 10% at an SNR of 13 dB in the non-coherent

situation. The increase in SNR necessary to achieve acceptable results from those of [12] can be at least partially attributed to the larger number of modulations tested.

The authors of [7] introduce a ML algorithm in the form of a quasi-log-likelihood ratio (qLLR) rule which is an approximation of the ML test. The classifier is limited to the two-case test of binary phase shift keying (BPSK) and quadrature phase shift keying (QPSK); however, both the synchronous and asynchronous cases are probed. Detector statistics are also developed using this method for the symbol-non-coherent and carrier-non-coherent cases. The qLLR classifier is compared to a square-law-based classifier and a phase-based classifier, with the qLLR classifier outperforming the other two and achieving good classification below 0 dB. These simulations were run under the assumption of perfect knowledge of the transmitted pulse shape at the receiver and ideal channel conditions with no intersymbol interference (ISI). However, these assumptions often cannot be realized in practice.

2.1.2 Feature-Based Recognition Methods

Liedtke [**15**] proposed an AMC method based on feature detection for amplitude shift keying (ASK), frequency shift keying (FSK), and PSK signals. The classifier requires estimates of the center frequency and bandwidth, although the precise bandwidth need not be known because the signal is fed through a set of parallel filters whereby the best estimate of the modulation retroactively identifies the best filter match. The feature used as a classification parameter is the phase change which is recorded as a histogram. The thresholds for distinction between the histograms for various modulations are derived from the ML criteria. Liedtke [15] was also the first to mention using an energy detector as a wideband search tool when classifying signals over a wide bandwidth.

The authors of [16] and [17] describe classifiers based on the wavelet transform. The wavelet transform determines localized frequency information and is therefore able to detect frequency transients. Both of the proposed classifiers use wavelet tree decomposition, which deconstructs the input signal into narrower frequency bands over several stages. In [16], the output of the wavelet tree is used to form an energy histogram

that is compared to known signal profiles. A hierarchical approach is used in [17], which also uses a much smaller wavelet tree. Both algorithms require at least a 10-dB SNR for reliable classification.

A classification method was presented in [1] that describes a very different method of classification than those thus far presented, although this method still falls into the class of feature-based recognition methods. The classification technique is based on the tracking and acquisition of a phase-locked loop (PLL), which is common among communication receivers as a method of carrier and phase tracking of a received signal. In this instance, a multimode PLL is used to lock on to the received signal, with the locking mode providing the knowledge of the modulation type. The multimode PLL consists of a bank of phase-locked detectors (PLDs), each designed for a specific modulation. The proposed system included PLDs for PSKs through order 8, which meets the criteria for adaptive downlinks in the satellite channels of interest. The inherent advantage of this system is that it naturally compensates for carrier frequency and phase mismatch. However, the system is limited to PSKs and may have difficulty locking as the modulation order increases.

2.2 Cumulants and Related Strategies

Classification using cumulants can take two forms. The first is a one-shot distance method comparing all of the calculated cumulants to the ideal value and the second uses a hierarchical structure of comparisons to narrow the suspected modulation to a single type. The one-shot distance method is very simple and easy to expand as more modulation types are added to the classifier. The hierarchal method is often able to identify a family of modulations (QAM, PSK, ASK, etc.) more accurately than isolating a specific constellation. This family classification can be beneficial in many situations as it can provide an initial level of user identification.

2.2.1 Cumulant Theory

Extensive research into the viability of cumulants as the basis for modulation classification has been undertaken due to some of the fundamental characteristics of the statistic. A few of these desirable features include robustness against phase rotation and Gaussian noise and simplicity of computation [4] [5] [18] [19] [20] [21] [22] [23] [24].

Cumulants are directly related to moments through the characteristic function by taking the natural logarithm before evaluating the derivative at zero [18] as seen in equation 2-1.

$$\Phi_X(\omega) = \ln\left(\sum_{n=0}^{\infty} m_n \frac{(i\omega)^n}{n!}\right) \qquad \text{(Equation 2-1)}$$

This has the practical application of allowing cumulants to be formulated by first computing moments, which is a simple process, and then combining them to form the cumulant value. In the case of fourth order cumulants of zero mean signals, the cumulant can be computed simply as [22]

$$cum(w,y,x,z) = E[wxyz] - E[wx]E[yz] - E[wy]E[xz] - E[wz]E[yx] \qquad \text{(Equation 2-2)}$$

where w, x, y, and z are input samples. For complex inputs, three fourth order cumulants are of interest, based on the conjugation of the input samples used in equation 2-2. In this work, cumulants will be denoted as $C_{p(p-q),}$ with p equal to the cumulant order and q equal to the number of non-conjugated inputs to the cumulant function.

Ideal cumulant values are calculated with the assumption of the signal containing unit energy. Therefore it is necessary to scale the calculated cumulant to unit power before comparing to the ideal value. In the presence of Gaussian noise, the noise power must be accounted for when performing the scaling, as shown in equation 2-3,

$$\tilde{C}_{4(4-q)} = \frac{\hat{C}_{4(4-q)}}{\left(\hat{C}_{21} - \hat{C}_{21g}\right)^2} \qquad \text{(Equation 2-3)}$$

where $\hat{C}_{21}$ is the estimated total power and $\hat{C}_{21g}$ is the estimated noise power. In practical applications, the noise power can be found by measuring when the signal is not present or in a nearby band that is signal free.

2.2.2 Approaches to Cumulant-Based Classification

Cumulants are often used in two forms: the regular cumulant [4] [20] [21] [22] and the cyclic variant [18] [19] [23] [24]. The cyclic cumulant uses a delay vector when computing the cumulant, which in turn is represented by a Fourier series. This step adds complexity to the classification algorithm; however, the delay removes any small carrier offset problems that occur when the estimate of the center frequency of the signal is incorrect [24] [5]. In either case, cumulants above order 4 are not affected by Gaussian noise, if the assumption is made that an appropriate record length is used in the computation.

Both methods can take similar approaches to the classification problem once the cumulant statistic is computed. The simplest case is a one-shot method comparing all of the calculated cumulants to the ideal value; the second case uses a hierarchical structure of comparisons to narrow the suspected modulation to a single type. The one-shot method is very simple and easy to expand as more modulation types are added to the classifier. The hierarchal method is often able to identify a family of modulations [QAM, M-ary phase shift keying (M-PSK), M-ary amplitude shift keying (M-ASK), etc.] more accurately than isolating a specific constellation [5] [20] [21] [22]. This family classification can be beneficial in many situations as it can provide an initial level of user identification. The following are some examples of theoretical implementations of classifiers based on cumulants.

The authors of [20] use fourth-order cumulants to distinguish between PSKs of order 2 through 8, including differential QPSK (DQPSK). The classification structure is a small hierarchy where two normalized cumulants are considered. The difficulty in this experiment is to find distinction between the 8-PSK and DQPSK signals, which share a constellation. This distinction is achieved through a phase differential algorithm.

Although the conditions of carrier frequency offset and symbol timing are considered, they are not implemented in the simulation where all conditions are considered ideal including square pulse shaping. The net result is the ability to properly classify 8PSK and DQPSK at SNRs of 10 dB in the two-class case. Of particular note in this experiment is the determination that, while additive Gaussian white noise (AWGN) is not apparent in the higher order cumulants, the noise does have an effect on the calculation of phase and differentiation technique.

In [5], a hierarchical scheme is proposed to classify digital modulations beyond just MPSKs. In this case, the thresholds separating modulation type for each calculated cumulant are derived from the variance of the statistic. Although this process leads to asymmetrical thresholds, they are easily implemented into a hierarchical plan. Also presented is quantitative analysis of the minimum number of samples (with one sample per symbol) to correctly identify modulations with an accuracy of 90, 95, and 99% in the two-class case. These estimates range from the tens of samples for tests such as the BPSK vs. 4-pulse amplitude modulation (PAM) case to tens of thousands of samples for the 16-QAM vs. 64-QAM cases. These numbers are calculated in the ideal noiseless case, although they are stated as "gross overestimates" of the necessary number even at low SNRs. The influence of common practical issues included frequency mismatch and timing jitter. The simulation results show that performance remains high under the condition that phase jitter is less than 10° and the carrier frequency offset is less than 0.02%. Synchronization error also causes a significant decline in performance after reaching 10% offset. Other performance issues are considered, which leads to the understanding that the classification system works well in most conditions if the SNR is above 10 dB for the multiclass case.

A more complex hierarchical scheme is presented in [4] where cumulants up to order 8 are considered by the classifier. The hierarchy includes 15 modulations from the M-ASK, M-PSK, and both square and star M-ary quadrature amplitude modulation (M-QAM) families. As a simplification, the threshold between different modulations is assumed to be the midpoint between the ideal values. The noted advantage of the

hierarchical structure in the experiment is its ability to gain a general classification for the signals even if the subclass is indeterminable. The simulations show that the structure is effective at generating the general class of modulation even at 5 dB; however, in the subclass cases of MASK and 16-QAM vs. 256-QAM, at least 15 dB of SNR is required to separate the modulation order with 1000 (ideal) samples.

2.2.3 Distance Method

The distance method is formed by determining the linear distance between each calculated feature, $\tilde{C}_{nk}$, and the stored ideal value for each modulation, $\bar{C}_{nk}$ by

$$\bar{D}_{nk} = \sqrt{\left(\tilde{C}_{nk} - \bar{C}_{nk}\right)^2} \qquad \text{(Equation 2-4)}$$

The total distance from each ideal modulation is then found by summing the distances from each cumulant by

$$\bar{D}_T = \sum_N \sum_K \bar{D}_{nk} \qquad \text{(Equation 2-5)}$$

where the index of the smallest value in $\bar{D}_T$ corresponds to the best modulation match.

This classifier is not ideal in the sense that it is assumed that the variance of $\tilde{C}_{nk}$ is the same for all modulations, which is not true. The authors of [5] use the ideal variance of $\tilde{C}_{nk}$ based on the number of input samples to form thresholds for a hierarchical classifier. Using actual variances in the distance calculation greatly increases the complexity of the one-shot distance method. For many modulations that are likely to be misclassified due to close values of $\hat{C}_{nk}$ (such as QAM modulations), the variances are also similar. Because of this similarity, any improvement using the variance is expected to be minimal and therefore disregarded in favor of reduced complexity.

A benefit of the simple nature of the one-shot distance method is the ease at which new modulations can be added to the classifier. All that must be done is append the ideal cumulant information with the data for the new constellation. In this manner, a classifier may be easily trained to identify new modulations by simply performing a long averaging sequence (to get a good estimate of the moments) in a high SNR environment. Although the accuracy of the classifier is likely to degrade as constellations begin crowding the number line, the adaptability of this method lends well to cognitive radio.

The authors of [24] present a classifier based on fourth through eighth order cyclic cumulants. Contrary to the hierarchical methods previously introduced, this algorithm generates a distance vector between the ideal cumulant value and the calculated value. The vector with the minimum overall length corresponds to the modulation type. Although the complexity of the classifier is greatly simplified over other structures, the algorithm requires tens of thousands of samples to maintain a high accuracy. This situation is due to all modulations being considered simultaneously but is especially confounded because of the introduction of 256-QAM, which has very similar properties to those of other QAM modulations. Also of note is that, as the order of modulation increases, more samples are needed because more symbols are available and the distribution of samples must be approximately equal from all symbols when determining the cumulant.

Spooner [23] presents a cyclic cumulant classifier with similar parameters as the one in [24]; however, the focus is instead upon multisignal environments. Again, the Euclidean distance is found between the ideal and the calculated cyclic cumulants; however, the distance is left squared, which speeds up computation in a practical environment. In the multichannel case, four signals [BPSK, QPSK, 8-PSK, and minimum shift keying (MSK)] are simultaneously transmitted within close proximity in frequency. Signal power ranged from 7 to 10 dB above the noise power. A total of 16,384 samples were used in the estimate, and all four modulations were correctly detected in more than 60% of the trials. If correct classification of the 8-PSK signal is excluded, the model boasts greater than 90% accuracy in the three-case test.

The authors of [19] enhance the computation of the cyclic cumulant by modifying the cumulant calculation to select a discriminating feature at the cycle frequency corresponding to the bitrate. This discriminating feature is parameterized so that the distance between features is maximized. The discriminating feature performs remarkably well, even at 0 dB and 500 symbols. The drawback is that the feature can only be optimized for the two-class case.

2.2.4 Hierarchical Method

One particular hierarchical structure is not used in all situations. The structure depends upon the modulation types that are to be classified and how much intermediary information would likely be gleaned from the operation, such as constellation family. Intermediate information is more robust than narrowing to a single constellation because related constellations have similar features, which creates unreliable classification in noisy environments or with limited sample sizes. Because of the similar features of constellation families, most hierarchal structures find the intermediate information as a byproduct. One of the disadvantages of the hierarchical structure is that, while the processing power of most machines makes the computations to move through the hierarchy tree trivial, adding a new constellation may necessitate large changes to the structure.

The goal of the hierarchical method is to separate the constellations into groups based on cumulants that have the greatest separation. Therefore, constellation families tend to make excellent branches as they are often grouped tightly together. Figure 2-1 illustrates how some common modulations are grouped when looking at different fourth-order cumulants ($|\tilde{C}_4|$). While $|\tilde{C}_{40}|$ can be used to identify any of the modulations, the use of $|\tilde{C}_{41}|$ at the initial split in the hierarchy would provide more distinction between QPSK and QAM from ASK when the SNR is low. The value of $|\tilde{C}_{40}|$ can then be used to separate QAM, QPSK, and 8-PSK with less chance of incorrectly classifying the modulation as ASK. A basic tree for this hierarchical structure is shown in Figure 2-2.

Hierarchies are naturally threshold based. The threshold should optimally be set based on the variance of the feature used as was done in [5]. Unlike the one-shot method of classification, no additional complexity is found when using the variance to set the decision thresholds. Branch comparisons can also be performed by using relative distances as with the one-shot method; however, this procedure will only work well if the variances of the groups are approximately equal and the groups do not have a wide spread.

Part of the hierarchy's strength is also a weakness. Because the structure flow is usually one way, if an incorrect decision is made at one point the error is compounded at each additional branch. This situation will occur if the branches are not separated enough or too few samples are used for the estimate. The one-shot distance method does not suffer from this problem because all features are considered simultaneously.

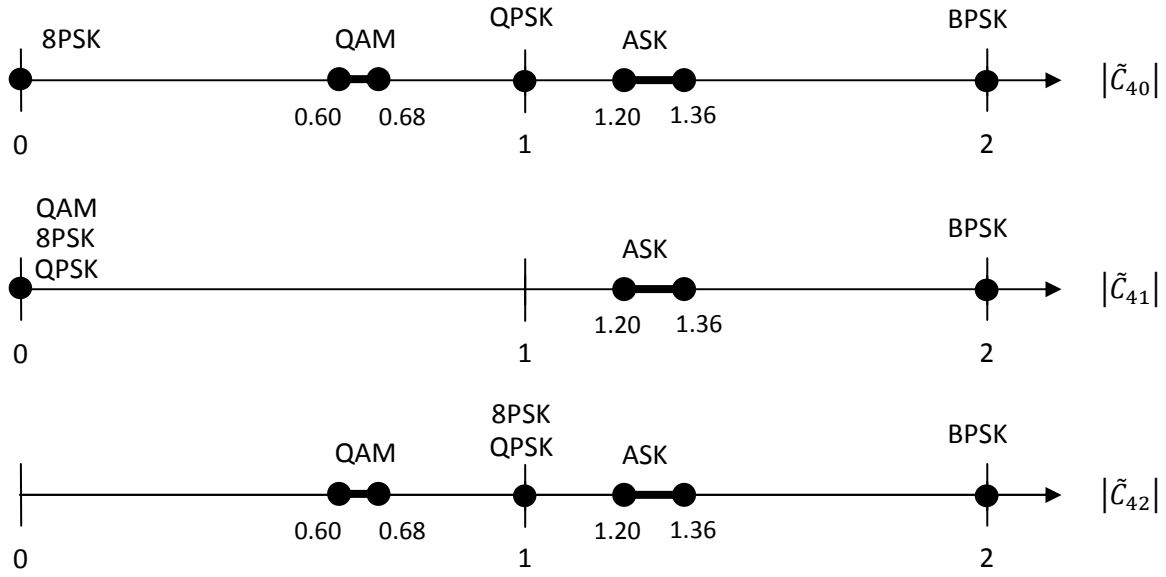

Figure 2-1. Grouping of constellations based on fourth order cumulant features.

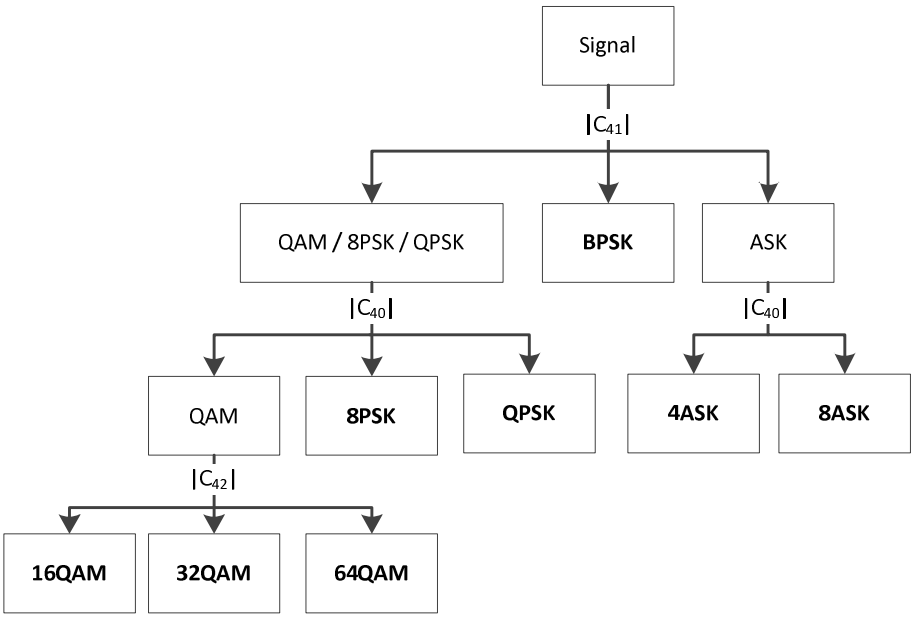

Figure 2-2. Hierarchical classification for several QAM, PSK, and ASK signals. Classifier outputs are marked in bold.

2.3 Real-Time Implementation

The theoretical works described in the preceding sections provide a rigorous and recreatable model for classification systems. Moving from the ideal environment formed by mathematics and simulation to practice involves accommodations for non-idealities due to the limitations of hardware and a live environment. Overcoming these issues is the focus of research implementations.

An early classifier is described in [25] that is capable of discriminating between several forms of AM and FM signals. The technique looks at the signal activity in terms of amplitude variation and zero crossing rate. The design uses mostly analog components, but the final decision is made digitally in a microprocessor. The system takes between 300 and 1500 ms to form a result.

A later design implementing classification techniques in real time is presented in [26], which moves more processing into the digital realm. The implementation is formed around a DSP chip that consists of a small 16-bit processor, a microcontroller, an 8-MHz

clock, and an A/D converter. The chip is used for both wideband sensing and signal identification. The identification is performed by comparing the spectral shape of the signal to a stored image. The modulation is not independently identified; however, the signal type is independently identified, assuming that only a limited number of types of signals are possibly present in the given spectrum. Because only the spectral shape is needed, issues such as frequency offset are not an issue, and the effect of additive noise is cancelled by averaging the spectrum over time.

A feature-based recognition classifier using the cyclostationary property of signals is described in [27]. This algorithm processes the baseband signal entirely in the digital domain on an FPGA. The FPGA is run-time reconfigurable, which allows for more efficient usage of the FPGA slices when performing the computationally complex algorithm. Another step that allows real-time classification is the addition of a mask to the cyclic spectrum so that only certain points are calculated, greatly reducing the complexity, but this requires prior knowledge of which modulations are likely. A slight frequency mismatch, such as 100 Hz when classifying a 4-MSPS signal, greatly diminishes the chance of correct classification. The algorithm does perform admirably well into negative SNRs; however, classification in that environment requires averaging the cyclic spectrum 10^4 or more times. While still considered real time, this averaging can add a significant delay (which is unspecified) between the appearance of the signal and the moment classification is complete. Also, the classifier was tested only with QPSK signals of known bandwidth, center frequency, and rolloff.

The authors of [28] describe the implementation of a likelihood-based algorithm on general purpose processors. The main advantage of this form of implementation is the flexibility and adaptability as the classifier enters new spectral environments with new modulation types. The testing parameters can be changed during run time to accommodate these new modulations. Carrier and phase recoveries are implemented in the system; however, certain known parameters, such as center frequency and pulse shape, are assumed to be known.

From these examples, the trend has been to push more of the classification process into DSP analysis. Of particular relevance to this project are the last two implementations discussed—one on an FPGA and the other using general purpose processors. The FPGA is successful in [27] in achieving high throughput on a complex classification algorithm. The versatility of general purpose processors is shown in [28]. The techniques described later in this paper attempt to utilize the benefits of both of these platforms in a single classification process while still achieving real-time results.

Chapter 3

Practical Classification Issues

3.1 Discussion of Issues

Cumulants have the advantage in real systems in that knowledge of the carrier phase is not needed for identification [20]. While immunity to phase rotation does not affect the cumulant result, improper estimation of the carrier frequency can cause severe degradation [22]. Another complication of a real system is the prolific use of pulse shaping, as is necessary in any bandwidth-limited system. Pulse shaping can dramatically affect the generated cumulant and therefore make a modulation type estimate unreliable. Because the parameters used for pulse shaping vary between applications, cumulant identification including pulse shaping information is impractical and in many cases impossible. Additional issues arise from the sampling limits of a real system. So that viable results can be achieved in a real-time environment, only a very limited length of time is available for samples to be collected.

Cumulant computation requires estimation of the expectation of signal characteristics, the variance of which is reduced (and therefore accuracy increases) with additional samples. The authors in [5] show that sometimes tens of thousands of samples are necessary to achieve reliable (>90%) classification, even in the noiseless case and between just two modulations. Practical systems also suffer from the inability to initially detect symbol timing, and, therefore, oversampling may be necessary. Other works [4] [10] have used anywhere from 2 to 50 samples per symbol when performing estimates. A real system may gain an estimate of an unknown signal based on the detected bandwidth; however, because of rolloff caused by pulse shaping and poor bandwidth estimations in low SNR environments, the bitrate estimation has limited accuracy.

3.2 Gaussian Noise

As mentioned in Section 2.2.1, higher order cumulants are robust against additive Gaussian noise because only the second moment of a Gaussian process is nonzero. However, because of a limited record length, Gaussian noise will still weaken the classifier as the SNR drops. In practical scenarios, many non-Gaussian noise sources exist. Thermal noise, which is often the limiting factor when receiving weak signals, is usually modeled as Gaussian. Additive Gaussian noise is also a standard used in characterizing a system's performance (Figure 3-1).

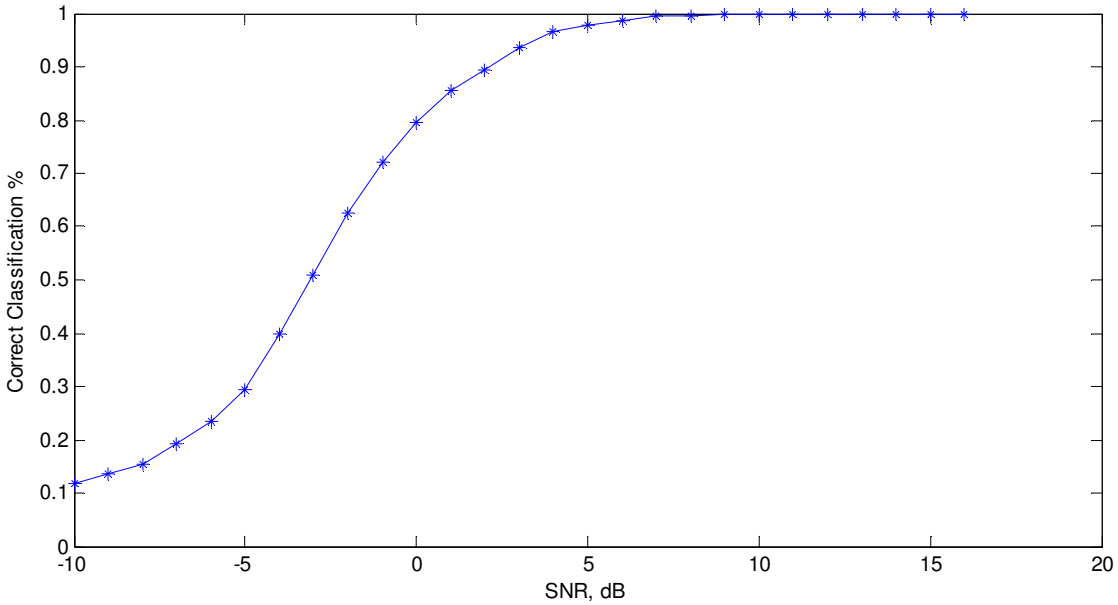

Figure 3-1. Correct modulation identification of nine modulations in Gaussian noise using the distance method.

Practical systems operating in a real-time environment are not able to compute enormous blocks of data to identify the modulation in use. Instead, the classifier must be able to operate reliably on a limited number of samples. This effect is studied in [**5**] for two-class problems with the minimum number of samples varying from the tens to the tens of thousands.

The current simulation was performed by using a number of samples varying from 250 to 5000 with 2 samples per symbol and square pulse shaping. All nine constellations are tested for by the classifier. The results are shown in Figure 3-2 for the QPSK case as an illustration.

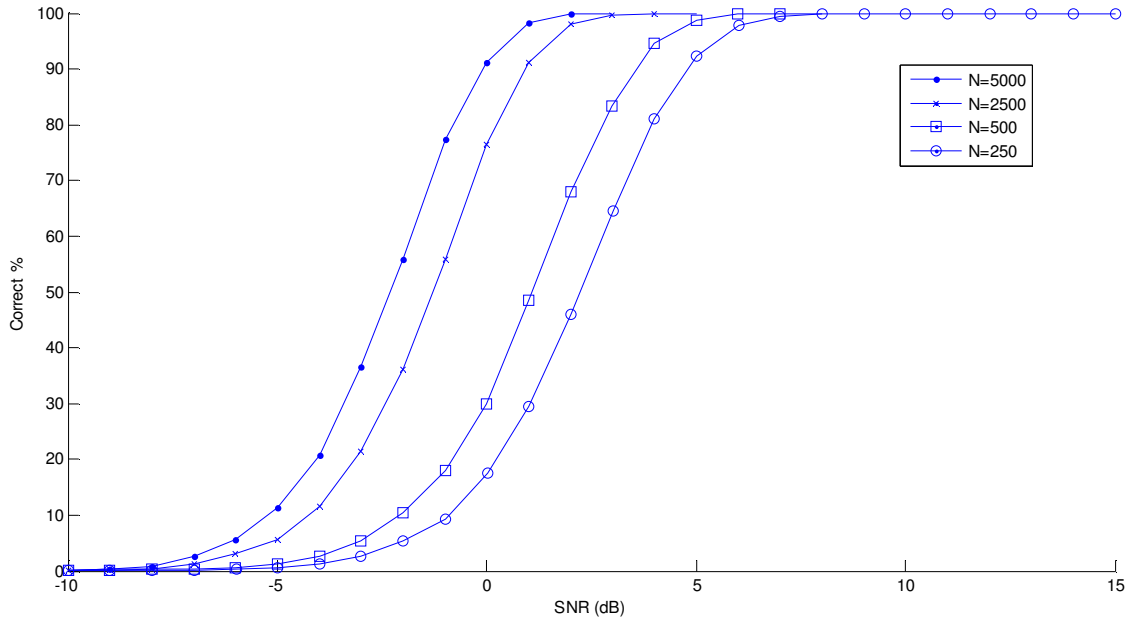

Figure 3-2. Probability of correct classification of QPSK with varying sample lengths.

3.3 Phase Rotation

In blind asynchronous classification, the phase of the received signal will not be known, which causes the constellation to rotate. Cumulants are known to be robust against this rotation, which will be demonstrated in the following simulation.

The simulation is performed similarly to the previous experiments, but with a constant phase offset added to each sample. The phase offset is tested at 20 evenly spaced points on $[0,2\pi]$ and tested with 1000 samples over 100 trials for each point. The resulting probability of correct identification is shown in Figure 3-3. That the phase rotation has no effect on the correct identification of a modulation can be easily seen.

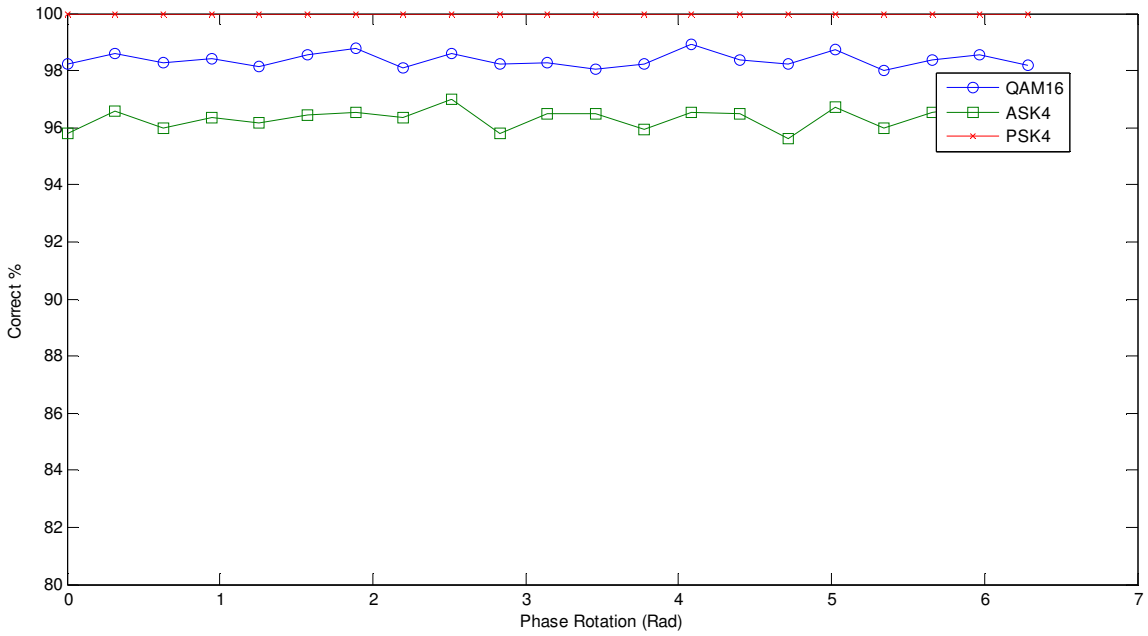

Figure 3-3. Effect of phase rotation on correct classification.

3.4 Frequency Offset

In a real system, even if a carrier frequency is known, this frequency may not be able to be recreated exactly at the receiver. Changes in humidity, temperature, or age can affect the resonant frequency of an oscillator and therefore can leave a baseband signal with a low-frequency sinusoidal component. The effort to match the carrier frequency is compounded in a modulation classification scheme because the signal's carrier frequency is unlikely to be known.

Normal receivers commonly use a form of a PLL to match the carrier frequency to a very high degree. A PLL constantly readjusts the frequency of the receiver's local oscillator until the phase difference between the local oscillator and the received carrier signal is minimal. Because the signal structure is unknown to the classifier, the phase change exhibited by most modulation schemes complicates the locking process. A common form of PLL is a Costas loop, which is able to lock onto and simultaneously

demodulate QPSK and BPSK signals. Higher order modulations require higher order loops. The restriction of a PLL locking onto only a select set of modulations is used in [1] as a modulation classifier itself. Because cumulants are already immune to phase rotation, a PLL is likely not the best method of locking onto the carrier frequency.

The authors in [20] describe two established methods of removing the effect of phase offset when computing cumulants. The first is to use differential moments (use samples separated in time) to cause the frequency offset to become a phase offset, which has no effect on the cumulant. The second method uses cyclic cumulants for classification. The differential method requires knowledge of the symbol timing.

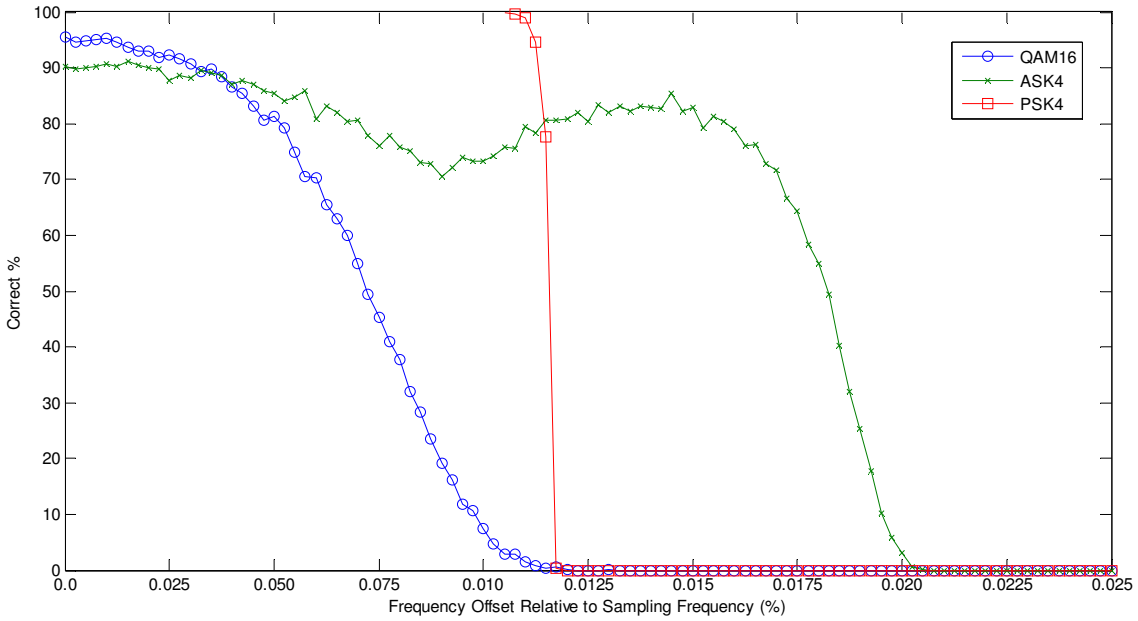

Figure 3-4. Frequency offset relative to sampling frequency.

3.5 Pulse Shaping

Real communication systems cannot send baseband data that has been directly up converted to the carrier frequency. Even when an attempt is made, natural components of the system would cause some band-limiting distortion of the signal. Practical systems

use pulse shaping to limit the signal to the desired bandwidth. A common pulse shape is the root-raised cosine (RRC), which is ideal for limiting inter-symbol interference. While a necessary component of modern communications, pulse shaping complicates the ability to use cumulants for modulation identification.

RRC pulse shaping extends the bandwidth of baseband signal by a rolloff factor of α. The rolloff factor can vary between 0 and 1 depending upon the desired bandwidth and complexity of the RRC filter. An estimate of the rolloff factor can be found by looking at the frequency spectrum of the received signal; however, an estimate often requires knowledge of the modulation as the spectrum appears different with different constellations.

Common cumulant classification techniques assume that the bits appear as rectangular pulses (or at least the sample is taken at the ideal moment). RRC pulses can be dealt with either by adjusting the ideal cumulant values to accommodate the rolloff or by using an equalizer as suggested in [5]. Adjustment of the ideal cumulant values is not feasible because the distribution of possible rolloff factors would allow for too many possible pairs of modulation and rolloff factor for a given cumulant value. Optimal equalization requires knowledge of the rolloff factor.

The following simulation was performed by pulse shaping the symbols of a QPSK signal and then computing the cumulant C_{40} by sampling at times other than the ideal instant. The results in Figure 3-5 show that the cumulant value is less than half that of the ideal case when the calculation is performed at midpoint between symbols. This emphasizes the need for accurate symbol rate synchronization and timing when performing classification using cumulants.

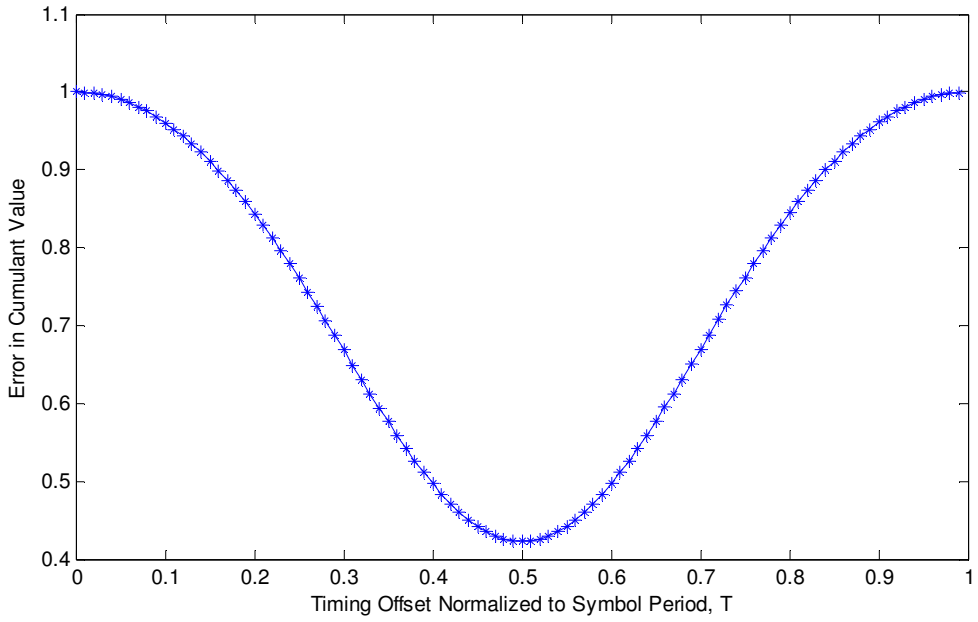

Figure 3-5. Simulation showing the effect of pulse shaping on cumulant accuracy.

Chapter 4

Design

4.1 Hardware Overview

The hardware employed by this radio consists of two main processing modules, an NI PXIe-7695 FPGA and a real-time processor (NI PXIe-8130 RT Module). The FPGA is directly connected to a D/A–A/D converter (NI 5781 Adapter Module). The converter is attached to an Ettus XCVR2450 transceiver board through a proprietary NI interposer board. The system is controlled through a personal computer, which provides graphical feedback for the user. A layout of the complete system (showing both transmitter and receiver) is shown in Figure 4-1.

The NI PXIe-8130 RT Module operates with dual 2.3-GHz cores, which are capable of dynamically splitting tasks between two threads. This is utilized by separating the wideband sensing and fine bandwidth estimation from the classification algorithm. The detector can then begin the fine bandwidth estimation of the next signal or return to performing the wideband spectrum sweep.

The FPGA is a Vertex-5 95T, with 14,720 slices and nearly 9 Mb of random access memory (RAM). The slices are used for almost every operation on the FPGA, while the RAM use was limited to storing data while averaging and providing temporary storage when restructuring the output of the definition of FFT (FFTs) from bit order to natural order. The FPGA resource usage can be found under Section 5.4.

The NI 5781 baseband transceiver serves as the A/D–D/A and is operated at 50 MHz, with a 14-bit input for the in-phase and quadrature components and a 16-bit output for the transmitter. The converter is attached to the interposer board through differential connections where the signals are formatted properly for the XCVR2540 RF front end.

The XCVR2450 performs the analog down conversion to baseband with a maximum bandwidth of 30 MHz. The bandwidth and receiver gain are controllable from

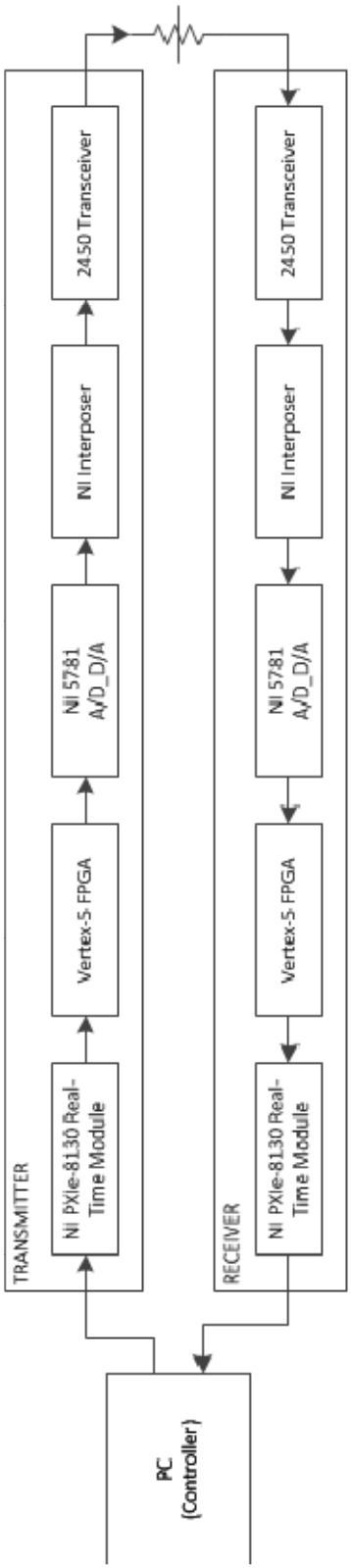

Figure 4-1. Hardware overview.

the NI PXIe-8130 RT Module . For testing purposes, the connection usually reserved for the antenna will be directly connected to a variable attenuator and then to the opposing XCVR2450. This setup reduces the effects of external signals; however, many signals are still detectable. These signals will be viewed as part of the interference which must be accepted in a realistic scenario and, therefore, useful to validate the real-world operation of the system.

4.2 Wideband Signal Detection

The first step in the classification process is to determine which signals are present in the band of interest. Because the radio frequency (RF) front end of this platform is the XCVR2450, the band of interest is chosen to be the 2.4-GHz industrial scientific and medical (ISM) band (2.4-2.5 GHz). The range of this band far exceeds the instantaneous bandwidth of the RF front end; therefore, the band is scanned in increments. This section explains the functionality of the wideband sensor.

4.2.1 Sensing Method

The spectrum sensor is implemented as an energy detector because the low complexity makes real-time implementation simple and viable for wideband scanning. All channel information is gathered from this algorithm. Two main types of energy detection are available—peak detection and average energy. While both detectors are based on a threshold, the peak detection is better able to detect narrowband signals but at the cost of being more susceptible to noise and requiring more memory to be dedicated to averaging. This radio prototype uses the average energy detector, which is less influenced by noise but conversely is less able to detect narrowband signals. This detriment may be overcome by reducing the channel bandwidth when averaging although, again, at the cost of the influence of additional noise.

4.2.2 Scanning Method

The wideband scanning of the ISM band requires adjusting the receiver center frequency many times. Data are sampled at 50 MHz. Because the received signal is down-converted to complex baseband, the outcome is an effective bandwidth of 50 MHz. The XCVR2450 is limited to a 30-MHz -6dB bandwidth, which means not all of the 50-MHz received bandwidth is viable for sensing. Instead, the bandwidth surrounding the center frequency is run-time selectable by the user/engine. The scanning process is complicated by the center frequency, which is impacted by the down-conversion process as seen in Figure 4-2.

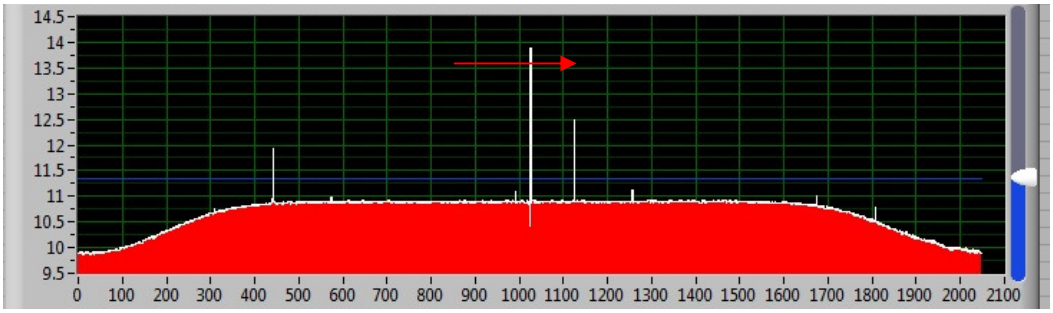

Figure 4-2. Scanning is complicated by additional very low frequency content added during the down-conversion process. Only the lowest channel(s) is (are) affected.

A complete description of the spectrum requires bandwidth jumps of less than 25 MHz. The alternative measurement pattern ignores the discontinuity at the receive center frequency and does not overlap. This method improves scanning speed, but fewer channels are available. Also, if the objective is to find all signals within the desired band, this method will miss narrowband signals that occur at the set receive frequency. When a channel is chosen, the channels that are affected by the receive discontinuity are listed as occupied when no overlap is performed. This procedure may affect the engine's choice of channel selection, especially if a wideband section (multiple contiguous channels) is desired for transmission.

The overlap method requires more scanning cycles to make a complete scan of the ISM band, the number of cycles depending on the amount of overlap used. If channels are set to be sufficiently small in bandwidth, the gain of using the overlap method may be small. In addition to the added scanning cycles, the overlap method also complicates recording channel selection criteria, which is stored over time. For example, a partial overlap will cause the Least Occupied counter in some channels to change faster than other channels. Also, the stored information for the Average Energy selection method will not necessarily represent the same time duration among different channels. This unequal recording method is compounded by using the same center frequencies for each full spectrum scan.

To partially alleviate the discrepancies introduced when overlapping the scanning cycles, the center frequencies for scans are offset after each full spectrum scan. Therefore, the discrepancies exist when only a few scanning cycles are viewed, but over many spectrum scans the information is smoothed. The main downside to this approach is that each full spectrum scan may not require the same number of scanning cycles, which affects the overall time required to scan the spectrum. This jitter is most likely negligible; however, jitter may cause problems for certain applications.

4.2.3 Signal Information

At the end of the scanning cycle, the relative occupancy data are used to determine if a signal is present and classifiable. A signal is defined as classifiable if it has been persistent and is active on the last scan. After this determination, the classifier goes through several steps to prepare for the cumulant measurement. These steps are required to obtain the best estimate of the cumulant through center frequency estimation and bandwidth estimation.

4.3 Signal Estimation

4.3.1 Rough Estimates

The first step uses the relative occupancy channel data to find a rough estimate of the signal bandwidth and center frequency. The occupancy information is normalized on [0,1]. Many signals will occupy more than one channel (depending on the signal's bandwidth). The grouping of channels into signals is done by passing the forward and reversed arrays through a threshold, where the index of each low-to-high crossing is marked. The indexes of the reversed array are then subtracted from the length of the array, effectively determining the index location from the original first index of the array. This reversed array is then re-reversed, and the original threshold indexes are subtracted from it. This process yields the bandwidth in terms of channels. These values are halved and then added to the forward array values after they were passed through the threshold. This yields the estimated center frequency in terms of channels.

4.3.2 Automatic Filter Adjustment

A critical aspect of the AMC platform is its ability to accommodate any likely signal in terms of both bandwidth and center frequency. The rough estimates from Section 4.3.1 are used to refine the identifiable characteristics of the signals before entering the classification step. This fine tuning requires a much higher frequency resolution than that for the wideband energy detector. For this purpose, the signal is filtered and decimated in several steps. Determination of the decimation factors is not a trivial step. Recommended practice stipulates that decimation when using FIR filters should not exceed 10, so the rates after each filter must be determined. The dissemination of decimation factors is performed by making each rate as large as possible without exceeding 10 and then iteratively adjusted a limited number of times. A larger bandwidth, corresponding to less decimation, is always preferred so that the edges of the signal are sure to be within range.

This decimated signal is then fed back into the energy detector previously used for the wideband estimate. The result is an accurate depiction of the spectrum near where the signal was detected with minimal additional resource use on the FPGA. The spectral data are then used to determine estimates of the 3-dB bandwidth and center frequency. The bandwidth is found by calculating the power 3 dB down from the maximum on either side of the maximum. The center frequency is expected to be midway between the 3-dB points (not necessarily corresponding to the maximum point).

These refined measurements are then used to further readjust the decimation rates and filter bandwidths in the same manner as described previously. However, instead of being fed from the decimation filters back into the energy detector, the data are forwarded to the classifier in the NI PXIe-8130 RT Module.

4.3.3 Persistent Signals (Signal Database)

The rough estimates of the signal parameters are used to determine if a signal has previously been detected by the scan. A signal is determined to be the same if neither the bandwidth nor center frequency change by a fractional amount. The change in center frequency is measured in channel bandwidth so that the threshold is constant throughout the band of interest.

Each signal stored from the previous scan is compared to the signals active during the current scan to determine persistence. The signals that are deemed persistent have their last active timer updated. The remaining signals are noted as new and are added to the signal list. The inactive signals are not discarded as they may resume during future scans. Keeping all signals allows for future additions, which track signal behavior, possibly by using a technique such as hidden Markov models. This type of addition would enhance the classification of signals, perhaps enabling identification of particular users. Because no signals are discarded, the search for persistent signals lengthens for each additional signal that appears in the spectrum. For this reason, it may be beneficial for future implementations to discard a signal if it does not appear in a given amount of time, as noted by the last active time stamp attached to each signal.

4.4 Classifier Structure

Once the fine estimate of the bandwidth and center frequency is found, the classifier is able to begin the process of estimating the cumulants. There are several processing steps that must take place for only good samples to be used in the estimates. The main steps involve resampling the signal at four times the symbol rate, differentially processing to remove any frequency offset, and then interpolating to estimate the ideal sampling value. These steps are presented in Figure 4-3. Another important aspect is the sub-block processing, which reduces the error of the symbol timing estimate. Once the cumulants have been calculated, the values are used to form a distance measurement that corresponds to a particular modulation.

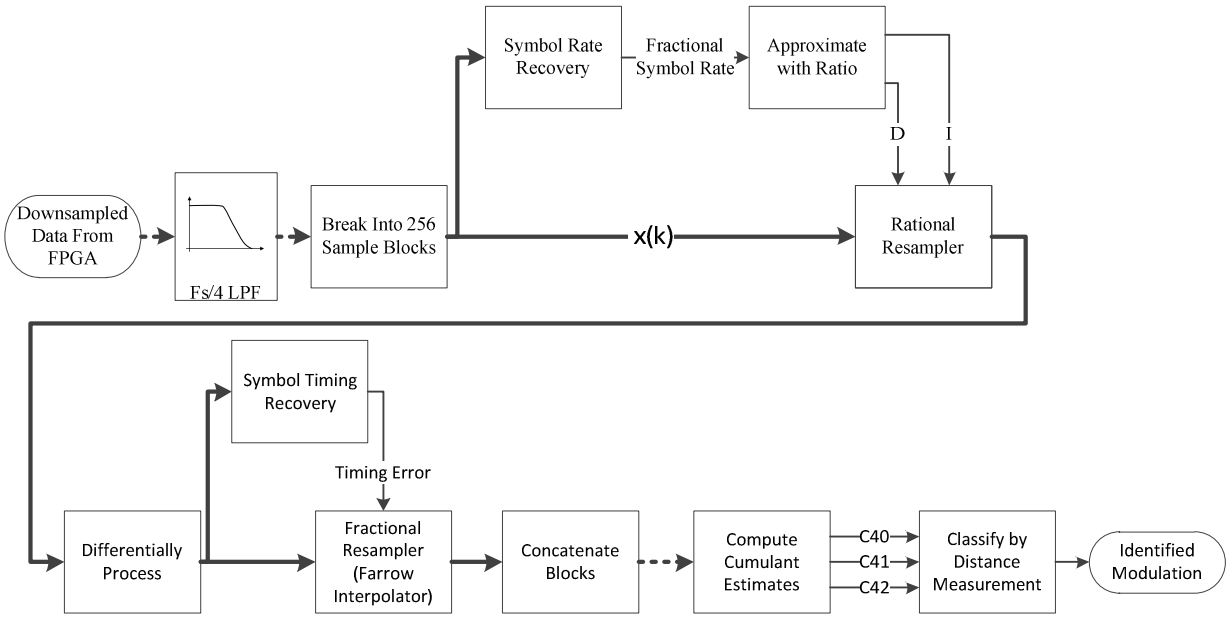

Figure 4-3. Classifier structure.

4.4.1 Symbol Rate Recovery

The data rate from the FPGA is likely to not be exactly 4 SPS, which is the required rate for some of the future processing steps. Therefore, the symbol rate must be determined so it can be correctly resampled at 4 SPS. The estimation algorithm is based on the square timing recovery algorithm, which is used in a later step. In this step, the symbol timing information is not important, just the rate.

The methodology for determining the input symbol rate is shown in Figure 4-4. The Fourier transform of the squared signal is found, which includes a tone at the symbol rate. This tone is found by searching for the maximum power of the squared spectrum in the range between approximately one-tenth and one-third the sample rate.

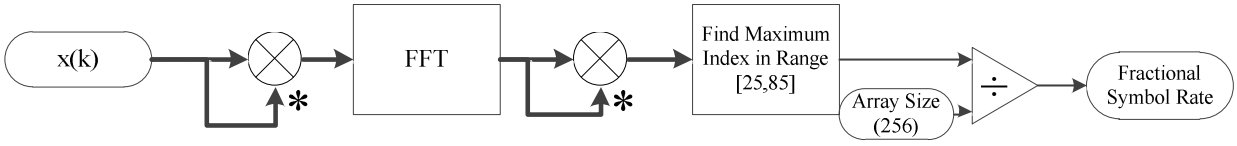

Figure 4-4. Symbol rate recovery.

4.4.2 Ratio Approximation

The rate found in the previous step is a fractional value that must be converted to values useful as inputs to a rational resampler. The resampler requires integer values for an interpolation factor, *I*, and a decimation factor, *D*. Ideally, these factors should be small to relax the restrictions on the aliasing filter that is part of the resampling process. Use of the ratio at the output of the symbol rate recovery block directly is not very useful because the factors are likely to be large (on the order of 100 or 1000). Truncation of these factors to a value in a reasonable range could cause severe distortion to the intended fraction. Factoring the ratio into primes and then simplifying are also not reasonable, both because computation of the prime factors of large numbers is time consuming and

because the resulting ratio could still involve large factors, approximated by a truncated version of the Euclidean algorithm called continued fraction decomposition.

The continued fraction decomposition method is an iterative process, which is depicted in Equation 4-1 [29]. The decimal ratio to be approximated is D_{ecimal}, while a_k are the intermediate coefficients and I and D are the numerator and denominator of the approximate ratio. A particular benefit of this method is that it can be stopped after any iteration based on either the number of iterations or the accuracy of the approximation.

$$D_{ecimal} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \cfrac{1}{\ddots}}}}} \approx \frac{I}{D} \qquad \text{(Equation 4-1)}$$

A block diagram for the continued fraction decomposition is shown in Figure 4-5. The process is formed out of two steps: first, the decimal fraction is decomposed into the coefficients of the continued fraction; second, the factors are reinserted into the fractional form to yield the numerator and denominator of the approximate fraction. A limit is placed on both the number of iterations and the expected accuracy of the decomposition. Inaccuracies affect the accuracy of the symbol timing recovery function for large blocks of data; however, this situation is accommodated through the sub-block processing of the data. Even if the determined ratio is exact, timing frequency offset and timing jitter still require sub-block processing.

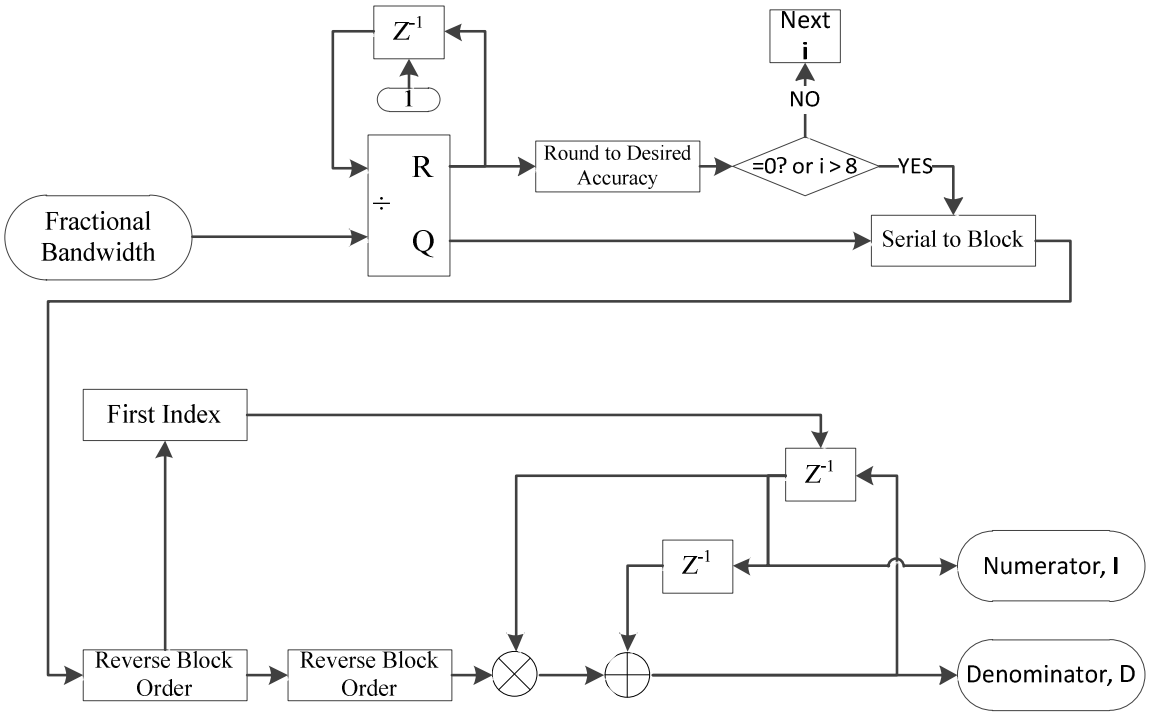

Figure 4-5. Overall block diagram for continued fraction decomposition.

4.4.3 Rational Resampling

Rational resampling is a critical step to adjust for any inaccuracies in the larger decimation process. After the necessary adjustment to the symbol rate is determined, the sub-block is fed into the rational resampler for the actual rate conversion. The outputs from the ratio approximation algorithm set the interpolation and decimation rates. The resampler itself is one of the software tools from the signal processing toolbox of LabVIEW.

The function performs adequately, and therefore no further development or modifications were found to be necessary. The principles of rational resampling can be found in most signal processing textbooks, such as [30], and therefore will not be explained here.

The resampler itself generates a finite impulse response (FIR) filter for the given pair of I and D. The filter is generated by using coefficients of a Kaiser window. An important note is that, after the resampling process, the number of samples used by further blocks is not known during the design process. This situation leads to an unequal number of samples in further steps from sub-block to sub-block.

4.4.4 Differential Processing

Most communications systems have the distinct advantage over AMC systems in that they are aware of the center frequency as well as the modulation type, which allows for simple frequency tracking using techniques such as a PLL. In the AMC environment, the center frequency can only be estimated, which will most likely contain a significant error, depending upon the method used, and is unable to lock on to the signal in the conventional manner due to the variety of signal types available. To counter this effect, the immunity of cumulants to phase rotation will be used to an advantage.

Differential processing is a technique where the signal is multiplied by a delayed and conjugated version of itself. This multiplication converts the frequency offset into a phase offset against which cumulants are robust. While this method is effective, it has the consequence of changing the constellation diagram of the original signal, often to one with less desirable characteristics, such as closer constellation points. A comparison of constellations before and after differential processing can be found in Section 5.1.

4.4.5 Symbol Timing

A necessary step that is often ignored or at least assumed to be ideal in most of the literature is accurate symbol timing recovery. While all digital receivers need some form of symbol timing function, the lack of information known a priori when performing modulation classification makes the problem non-trivial [4]. Accurate symbol measurement is critical when performing classification because ideal cumulant values (or any other statistical measure) are calculated with the assumption of ideal timing. Inaccurate measurement will add significant variance to the symbol estimates in the best

case and may form entirely new constellations (due to periodic error) in the worst case. This section describes the theory and implementation of a square timing recovery algorithm to perform blind symbol timing estimation.

Digital square timing recovery is a digital realization of an analog technique that effectively detects the phase of the spectral line corresponding to the symbol rate. This spectral line appears after the squaring operation on the input signal. In the analog case, the spectral line is extracted with a PLL; however, in the digital case, the line is found using a discrete Fourier Transform (DFT) [31]. After computation of the DFT, the phase angle at the symbol rate is found, which is then normalized by $-T/2\pi$ to determine the estimated symbol timing error, $\hat{\varepsilon}$. $\hat{\varepsilon}$ is then filtered. As noted in [31], there is the potential for the phase to enter unstable equilibrium, which occurs if the input samples are a half sample off of the ideal sampling time. To eliminate this problem, the authors added a Kalman filter before determining the phase angle. The simplicity in this implementation is maintained when a low-pass FIR filter is used instead. Because of the filtering operation before the phase angle is computed, the filtering of $\hat{\varepsilon}$ is omitted. A block diagram of the operation is shown in Figure 4-6.

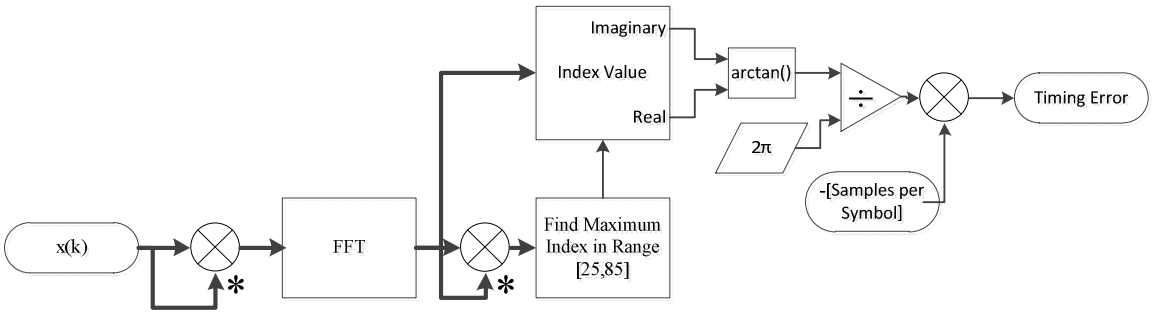

Figure 4-6. Square symbol timing recovery

The authors in [**32**] present a method where the squaring operation is replaced by multiplication of the signal with a delayed and conjugated version of itself. This method is identical with the differential processing technique described in Section 4.4.4. A single constraint is added whereby the delay must be an integer multiple of the symbol rate.

This constraint is easily met, which results in the processing algorithm depicted in Figure 4-7, where the differential processing block is combined with the symbol timing recovery block.

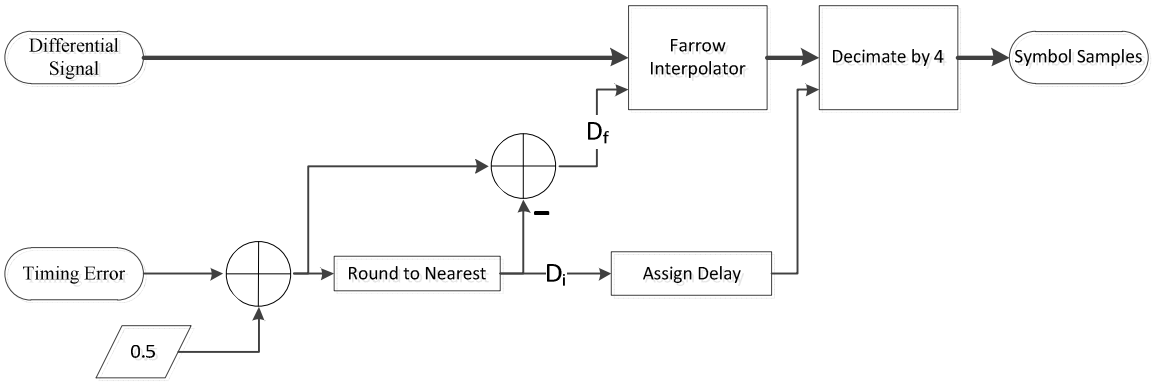

Figure 4-7. Fractional resampling.

4.4.6 Interpolation (Fractional Resampling)

The output from the symbol timing estimation block is a value expressing the delay between the midpoint time of the input samples and the ideal sampling time for the symbol. Because only four samples are taken per symbol, the ideal sampling time is likely not to be one of the four input samples. An interpolator is used to estimate the value of the symbol at such an intermediate point.

An important distinction is that this is not an interpolator in the standard communications sense of the word but instead is an interpolator in the mathematical sense. While an interpolator is often inferred to be a block that upsamples and then filters to provide a change in sample rate, this interpolator resamples the signal at different times while maintaining the same rate.

Interpolators have many uses in and out of engineering, particularly in graphics design and line fitting of data. Interpolators work by weighting the value of nearby samples on the basis of the distance to the desired sample and the interpolation method. Many functions are available from the very simple nearest-neighbor method to much more complex Gaussian methods [**33**]. In general, the complexity of an interpolator

increases as the number of samples used in the estimate increases. In the case of the nearest-neighbor method, the value of the closest sample is given the full weight of the function. A slightly more complex method is a linear interpolator, which estimates the value of the intermediate sample as if it were placed on a line drawn between the two nearest points. This process is performed by weighting the sample values inversely proportional to their distance from the desired point. Accuracy increases for continuous input as the order of the weighting scheme and number of nearby points increases. A survey of many interpolation methods can be found in [33].

Because interpolators function by summing the weighted values of samples, they are naturally implemented in the form of a FIR filter. If only a constant resampling interval is needed, a standard FIR filter is completely adequate, as the offset measurement can be pre-computed and combined with the weighting factor to produce a filter coefficient. However, in the case where a variable offset is desired such a simple structure is not applicable as the coefficients must change. One method of implementing a variable offset is to compute multiple interpolated values and then select the nearest one (effectively using nearest-neighbor interpolation among the interpolated values). A more effective structure for fractional interpolators is the Farrow filter as shown in Figure 4-8. The Farrow filter structure uses filter coefficients which have been decomposed into polynomial approximation functions using Horner's rule [33] [34]. The Horner form (Equation 4-2) reduces the number of multiplications of nth order polynomial from $2n$ to n [**35**].

$$h_n\Delta^n + h_{n-1}\Delta^{n-1} + \cdots + h_1\Delta + h_0 = \big((h_n\Delta^n + h_{n-1})\Delta + \cdots + h\big)\Delta + C_0 \qquad \text{(Equation 4-2)}$$

The polynomial coefficients, h, are then used as the coefficients of a group of filters with the output multiplied by the distance factor, Δ. For this implementation, the selected polynomial is a third-order spline using six data samples as described in [33] (Equation 4-3):

$$
{}^{\text{Cubic}}h_6(x) = \begin{cases} (6/5)|x_\Delta|^3 - (11/5)|x_\Delta|^2 + 1, & 0 \le |x_\Delta| < 1 \\ -(3/5)|x_\Delta|^3 + (16/5)|x_\Delta|^2 - (27/5)|x_\Delta| + 14/5, & 1 \le |x_\Delta| < 2 \\ (1/5)|x_\Delta|^3 - (8/5)|x_\Delta|^2 + (21/5)|x_\Delta| - 18/5, & 2 \le |x_\Delta| < 3 \\ 0, & \text{elsewhere} \end{cases}
$$

(Equation 4-3)

where x_Δ is the time duration between input sample point and the desired resampled instant. Many options are available for choosing a filter polynomial. The cubic family of filters performs approximately equally and has similar performance to other filters such as the Lagrange and Gaussian interpolation kernels. The difference is assumed to be negligible in this case because at high SNRs the interpolation filter is necessary to obtain correct classification and at low SNRs the differences between the polynomials are dominated by error caused by noise. Because the input data stream is continuous, the most accurate spline approximation will be made if the desired point is between the two center samples during the computation. The Δ's must be equal when calculating the interpolated sample for each input sample. Therefore, a substitution of variables is used for x_Δ for each of the six input samples as seen in Equation 4-3. After expanding the terms, the filter coefficients are visible as the coefficients in Table 4-1.

Table 4-1: Filter Coefficients for Farrow Interpolator after Variable Substitution

	Filter		
h_3	h_2	h_1	h_0
0.025	0.05	-0.1	-0.2
-0.125	-0.15	0.5	0.6
0.6	1.3	-0.4	-1.2
0.6	-1.3	-0.4	1.2
-0.125	0.15	0.5	-0.6
0.025	-0.05	-0.1	0.2

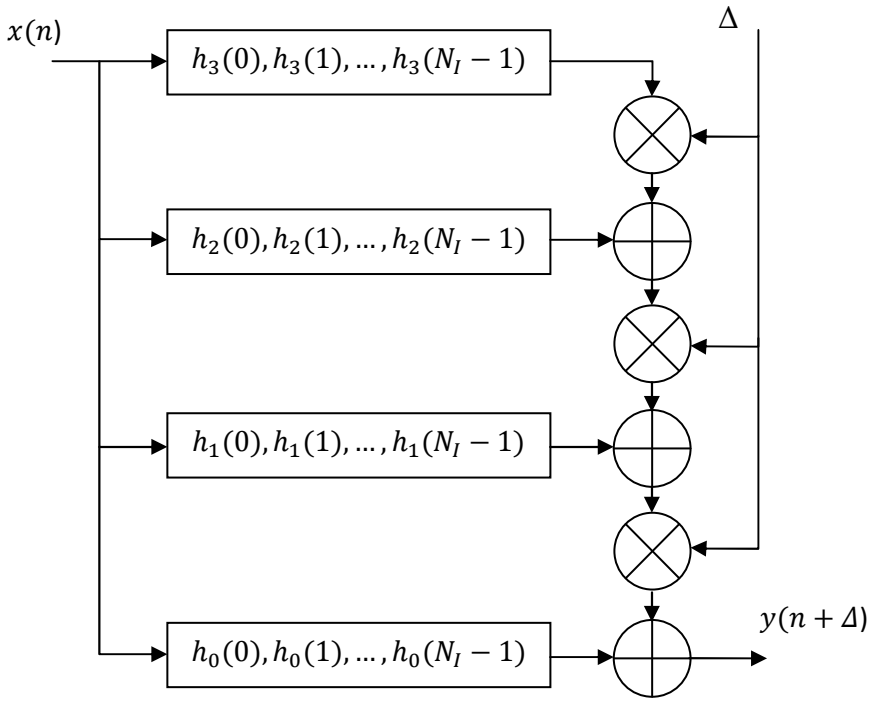

Figure 4-8. Farrow filter structure for variable resampling.

Because the signal is oversampled by a factor of 4, only one out of every four samples is taken from the output of the interpolator. The initial delay at the beginning of the sequence is calculated on the basis of the symbol timing delay calculated by the squared timing recovery step. Because the desired sample is always placed in the center of the interpolator, up to two of the original samples may be dropped if the desired sample is between the first two input samples. This dropping will cause a loss of at most one good sample point in the cumulant estimation, which is assumed to be negligible because several thousand samples are expected to be taken for each classification decision.

4.5 Transmitter Design

4.5.1 Design Criteria

A transmitter was developed to create various signals to be classified. An identical hardware platform was used. Testing such a variable system required a very flexible transmitter to accommodate many symbol rates, pulse shapes, transmit powers, and, of course, modulation types. In addition to the necessary flexibility, the radio must also run in real time.

4.5.2 Transmitter Structure

The transmitter design was developed by generating batches of symbols on the real-time processor. As a result, a random number generator could be used, which is not available on the FPGA module. Additional control of the modulation type, symbol rate, and transmit power is obtained through the NI PXIe-8130 RT Module .

The symbols are loaded onto the FPGA through a real-time first-in-first-out queue (FIFO). These symbols are then passed through a 101-tap pulse shaping filter. The large number of coefficients allows for creativity when developing the pulse shape. In particular, when narrowband signals are produced, the large set of taps keeps the effects of a limited filter length below -50 dB. The coefficients can be reloaded from the NI PXIe-8130 RT Module during run time without interrupting throughput. The shaped symbols are then interpolated through a variable-rate cascaded integrator-comb filter to make final bandwidth adjustments before the information is sent to the digital-to-analog converter. This process can be seen in Figure 4-9.

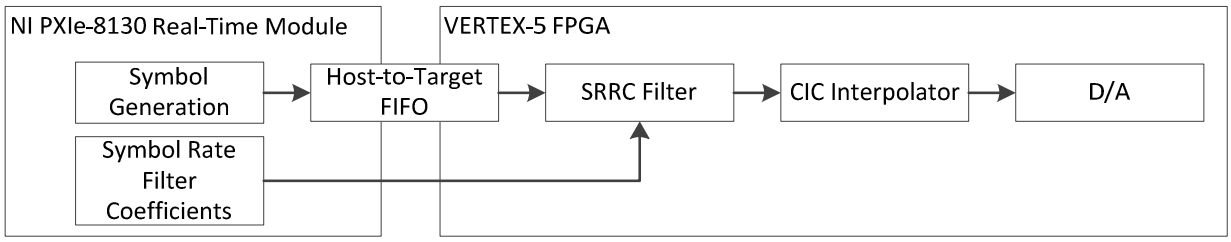

Figure 4.9. Transmitter structure.

4.6 Algorithm Distribution

The real-time functionality of the classifier is possible by performing high sample rate operations on the Vertex-5 FPGA and using the processors for more dynamic sections of the algorithm and data storage. Determinist FIFOs protect against data loss and ensure cycle accurate transfer between the FPGA and the NI PXIe-8130 RT Module. The division of processes between the platforms is displayed in Figure 4-10.

The FPGA processing consists of two paths, one which forms the energy detector and another which provides variable bandwidth down sampling. The energy detector path starts with a variable-length FFT, which can be changed during run time. The output of the FFT is summed by incrementing values stored in memory on board the FPGA. When the sum is completed, the values are concatenated to form channels and pushed through a FIFO queue to the NI PXIe-8130 RT Module.

The parallel path consists of cascaded pairs of FIR filters and decimators. The filters are 50 taps each, with reloadable coefficients to adjust bandwidth as necessary. The filter coefficients are computed during run time on the NI PXIe-8130 RT Module and passed onto the FPGA through a host-to-target (H-T) FIFO. Many different filter windows are run-time options; however, all testing was performed by using a rectangular window. Each filter is followed by a decimator, which allows for an integer rate change. This decimated output can be routed either directly to the NI PXIe-8130 RT Module or back into the energy detector. The rerouting is an efficient method of reusing the energy detector both for spectrum scanning and for bandwidth estimation once a signal is detected. The detector parameters, such as the number of averages, are usually changed between wideband sensing and bandwidth estimation; the testing operations used 100 averages when sensing and 50 averages when estimating the bandwidth and center frequency of the detected signals.

Processes on the NI PXIe-8130 RT Module are broken segregated between the two cores: one core functions primarily as the spectrum scanner and the other performs the classification. The processing core focused on scanning receives the channelized information from the FPGA and then passes the information through a threshold. The

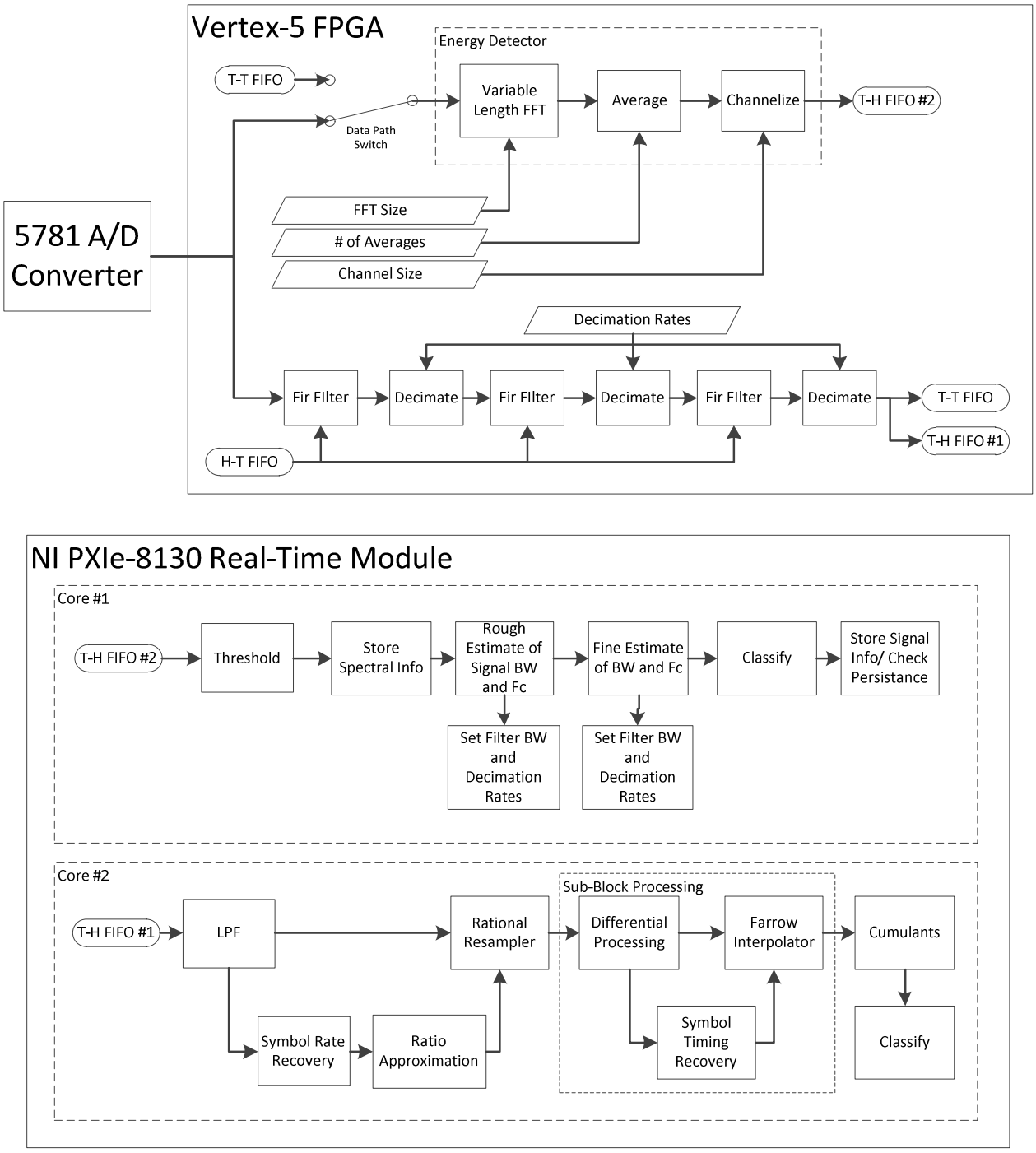

Figure 4-10. Division of processing across the Vertex-5 FPGA and NI PXIe-8130 RT Module .

threshold is set to achieve a particular rate of false alarm by the user. The false alarm rate is discussed in Section 5.3. This information, as well as the raw channelized energy, is stored in arrays to keep a record of recent spectral history. The channel information is also used to update other channel statistics, such as last occupied time and how often it has been occupied during the lifetime of operation. If the detector is set to perform classification, then the rough estimate of the bandwidth and center frequency is formed out of the threshold test. A fine estimate of the bandwidth is found by setting the input to the energy detector on the FPGA to the end of the down-sampling chain. After this estimate is found, the other processing core takes control of the FPGA settings and performs the classification. After the raw data are gathered for classification, the first core begins estimating the bandwidth of the next signal; therefore, both cores are operating simultaneously. After classification is complete, the signals are passed through a persistence check and the stored signal data are updated.

The second processing core is dedicated to the classification algorithm. The process has already been described earlier, in Section 4.4. The core processes data drawn from the FIFO containing down-sampled data. The filters and decimators are configured from the other processor; therefore, the values on this core do not have to be reset. The classifier output is stored in a memory location that can be accessed by both processors. Read/Write conflicts are avoided due to the delay between classification and the persistence check, which are the only times that memory location is accessed.

Chapter 5

Tests and Results

5.1 Modulations of Interest

A very large number of linear modulations have been implemented in communication systems to date. In addition to the standard BPSK, QPSK, OOK (2-ASK), and V32 (square) QAMs, there are V29 (star) QAMs, differential phase shift keying (DPSKs), and custom modulations that change by the implementation, perhaps even during runtime. For the purpose of testing this classifier, only a limited selection of modulations will be tested. These modulations are shown in Table 5-1, and the constellations (both normal and differential) are shown in Figure 5-1. For most of the modulations left out of this study, simply calculating the ideal (differential) cumulant values and adding them to the distance calculation are enough to expand the classifier's capability. For some modulations, however, additional steps may need to be taken. One such example is DQPSK, which has the same constellation as 8-PSK and, hence, the same expected cumulant values. A solution to this problem is proposed in [4] and would not significantly add to the computational complexity of the classifier.

Table 5-1: Modulations of Interest

Modulation	C_{40}	C_{41}	C_{42}	C_{D40}	C_{D41}	C_{D42}
BPSK	2	2	2	2	2	2
QPSK	1	0	1	1	0	1
8-PSK	0	0	1	0	0	1
4-ASK	1.36	1.36	1.36	0.3106	0.3106	0.3106
8-ASK	1.2381	1.2381	1.2381	0.1053	0.1053	0.1053
16-QAM	0.68	0	0.68	0.4624	0.2576	0
64-QAM	0.619	0	0.619	0.3832	0.093	0
256-QAM	0.604	0	0.604	0.3657	532	0
V29	0.5171	0	0.5823	0.2571	0.1	0

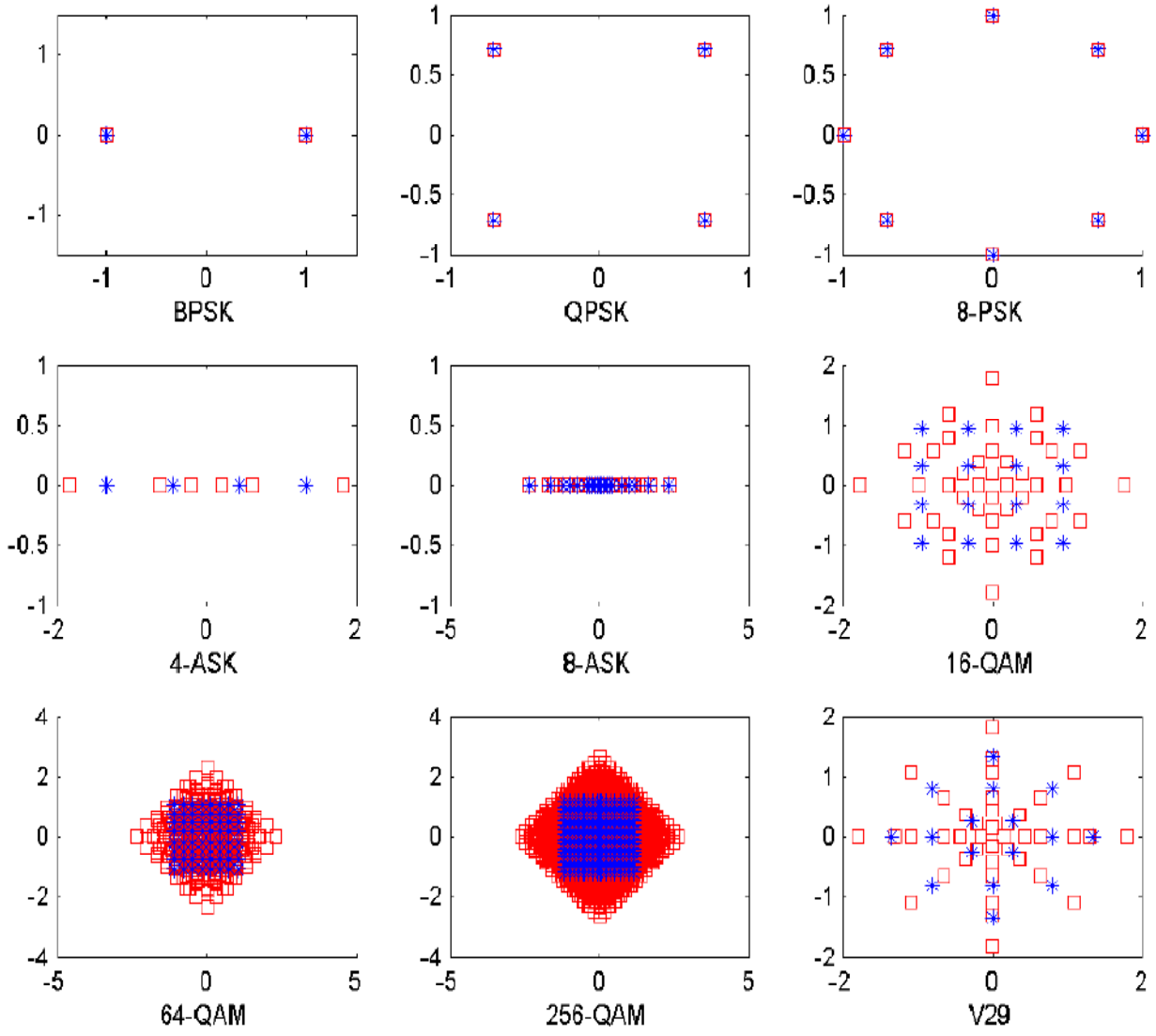

Figure 5-1. Normal (blue asterisk) and differential (red square) constellations.

Increasing the number of modulations in the classifier's database dramatically reduces the probability of correct classification. This result can be seen in much of the background work shown in Chapter 2, as well as in the results shown in this chapter. If an environment is known to only contain a few types of signals, limiting the classifier to only compare these known types will definitively increase the probability of correct

identification over a system with excess types. This situation may be of particular use in a cognitive network where the modulations change on the basis of environmental factors. A new node may wish to collaborate with the network but is unsure of what modulation is currently being used. If the possible modulations are drawn from a selective set (e.g., the PSK family), the new node has an advantage when classifying.

5.2 Classifier

5.2.1 Performance with All Modulations

The classifier is initially tested separately so that the functionality can be determined independently of the wideband scanner; therefore, the frequency, bandwidth, and noise power estimates preformed by the wideband scanner will not influence the classifier results. This situation will allow for objective comparison between the theoretical work presented earlier and this implementation.

Tests are performed after the center frequency and bandwidth have been established. The noise power is determined by averaging the power over 50 sets of input data with no signal present. Each set corresponds to one classification loop. This is a live test; the selected frequency (2.481 GHz) has rarely been seen in use. However, the ISM band is open to many communications technologies which may enter the band at any time. A particular concern is that of Bluetooth signals, which operate in the testing bandwidth and, due to their frequency hopping nature, are difficult to track during testing. No measures are taken to negate these effects, other than the spectrum is observed prior to testing and appears empty.

A set of confusion matrices are developed for SNRs of 0–20 dB in steps of 5 dB, as seen in Table 5-2. Each row corresponds to a particular modulation being transmitted. The columns show the signal classification. Each modulation was run for 1000 trials at each SNR; however, some points were removed if an interfering signal was present (noted by an increase in SNR by more than 2 dB). Therefore, at least 9500 trials were

used in the creation of each row. The signals transmit at a rate of 100 kSPS and are passed through a square root raised cosine pulse shaping filter with a rolloff of 0.25. The signal power is adjusted by a combination of an analog attenuator and adjustment of the transmitter gain.

Table 5-2. Confusion Matrices for 0-20-dB SNRs

a. 0-dB SNR

	BPSK	QPSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM	256-QAM	V29
BPSK	92.2%	0.4%	1.2%	0.0%	0.6%	0.0%	0.0%	0.3%	5.2%
QPSK	81.1%	1.2%	1.1%	0.0%	6.9%	0.0%	0.0%	1.7%	7.9%
8-PSK	62.7%	33.8%	2.7%	0.1%	0.1%	0.0%	0.0%	0.2%	0.3%
4-ASK	92.2%	0.3%	0.4%	0.0%	0.3%	0.0%	0.0%	0.2%	6.6%
8-ASK	78.8%	4.1%	16.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.7%
16-QAM	13.0%	7.1%	61.4%	17.1%	1.2%	0.2%	0.0%	0.0%	0.0%
64-QAM	99.6%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
256-QAM	89.5%	0.2%	0.5%	0.0%	4.6%	0.0%	0.0%	4.8%	0.4%
V29	88.8%	0.8%	0.5%	0.0%	0.6%	0.0%	0.0%	0.2%	9.0%

b. 5-dB SNR

	BPSK	QPSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM	256-QAM	V29
BPSK	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
QPSK	1.1%	98.5%	0.0%	0.0%	0.0%	0.1%	0.0%	0.2%	0.0%
8-PSK	1.6%	0.6%	84.6%	12.1%	0.1%	0.1%	0.0%	0.1%	0.8%
4-ASK	11.0%	29.9%	13.0%	41.2%	0.0%	4.7%	0.1%	0.0%	0.0%
8-ASK	7.9%	3.4%	68.0%	20.7%	0.0%	0.0%	0.0%	0.0%	0.0%
16-QAM	3.9%	48.0%	1.0%	7.5%	0.2%	16.3%	21.1%	1.8%	0.2%
64-QAM	0.0%	35.7%	8.4%	14.1%	0.0%	41.7%	0.1%	0.0%	0.0%
256-QAM	2.6%	21.1%	15.0%	29.3%	0.4%	30.8%	0.1%	0.1%	0.5%
V29	2.8%	33.8%	24.1%	29.7%	1.1%	8.4%	0.0%	0.0%	0.0%

c. 10-dB SNR

	BPSK	QPSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM	256-QAM	V29
BPSK	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
QPSK	1.7%	97.4%	0.0%	0.0%	0.2%	0.3%	0.0%	0.3%	0.1%
8-PSK	1.6%	0.3%	97.1%	0.0%	0.2%	0.0%	0.0%	0.1%	0.6%
4-ASK	0.0%	0.0%	0.0%	51.8%	1.1%	0.0%	8.5%	26.8%	11.9%
8-ASK	0.0%	0.0%	0.0%	66.9%	33.1%	0.0%	0.0%	0.0%	0.0%
16-QAM	0.0%	0.0%	0.0%	5.5%	1.8%	40.6%	32.5%	9.4%	10.2%
64-QAM	0.0%	0.0%	0.0%	1.3%	0.4%	3.9%	42.8%	41.7%	9.9%
256-QAM	1.3%	0.2%	0.0%	0.4%	3.8%	15.2%	64.8%	14.2%	0.0%
V29	0.0%	0.0%	0.0%	16.7%	42.1%	1.1%	6.7%	1.2%	32.2%

d. 15-dB SNR

	BPSK	QPSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM	256-QAM	V29
BPSK	99.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
QPSK	0.6%	99.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.2%	0.1%
8-PSK	0.7%	0.1%	98.8%	0.0%	0.1%	0.0%	0.0%	0.0%	0.3%
4-ASK	0.0%	0.0%	0.0%	99.9%	0.0%	0.0%	0.0%	0.0%	0.0%
8-ASK	0.0%	0.0%	0.0%	1.6%	98.4%	0.0%	0.0%	0.0%	0.0%
16-QAM	0.0%	0.0%	0.0%	6.5%	0.7%	70.2%	22.1%	0.0%	0.4%
64-QAM	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	58.3%	36.4%	5.2%
256-QAM	0.6%	0.1%	0.0%	0.2%	0.2%	0.2%	37.8%	60.8%	0.0%
V29	0.0%	0.0%	0.0%	0.0%	6.7%	0.0%	0.0%	3.2%	90.0%

e. 20-dB SNR

	BPSK	QPSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM	256-QAM	V29
BPSK	99.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%
QPSK	0.5%	99.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%
8-PSK	0.6%	0.1%	98.9%	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%
4-ASK	3.2%	4.2%	0.2%	89.4%	1.1%	1.0%	0.2%	0.0%	0.6%
8-ASK	0.0%	0.0%	0.0%	1.0%	99.0%	0.0%	0.0%	0.0%	0.0%
16-QAM	0.0%	0.0%	0.0%	3.1%	0.2%	89.3%	7.4%	0.0%	0.0%
64-QAM	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	89.8%	9.2%	1.0%
256-QAM	0.5%	0.1%	0.0%	0.2%	0.1%	0.2%	23.2%	75.8%	0.0%
V29	0.0%	0.0%	0.0%	0.0%	3.5%	0.0%	0.0%	5.3%	91.2%

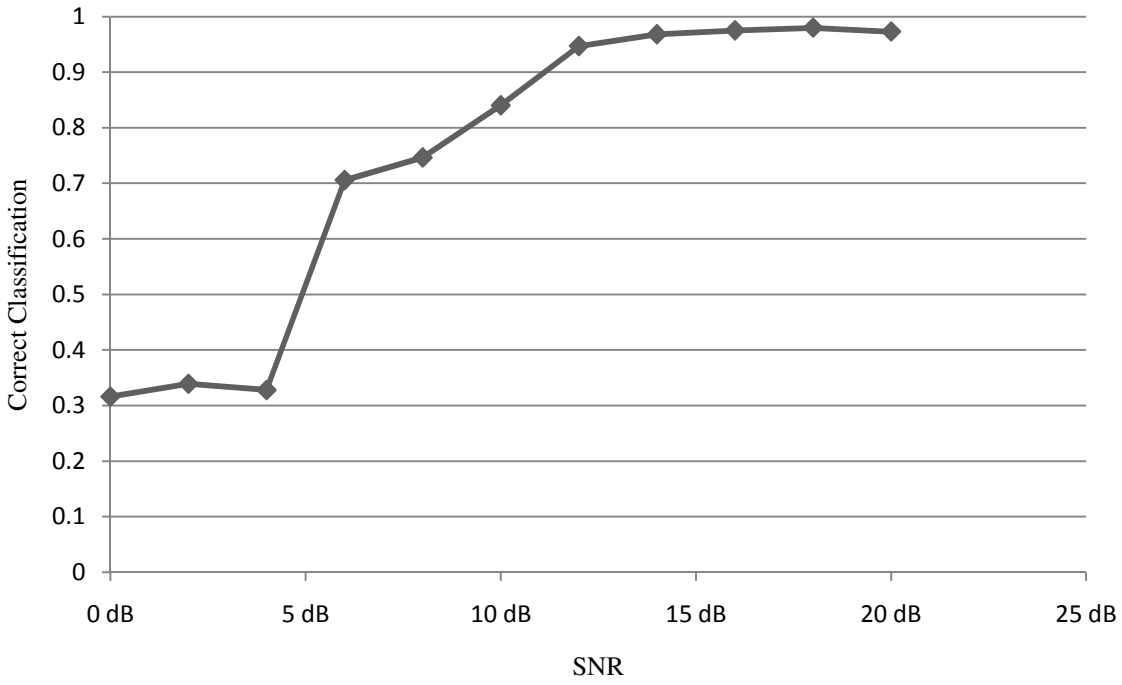

Figure 5-2. Overall ability to correctly classify the nine presented modulations.

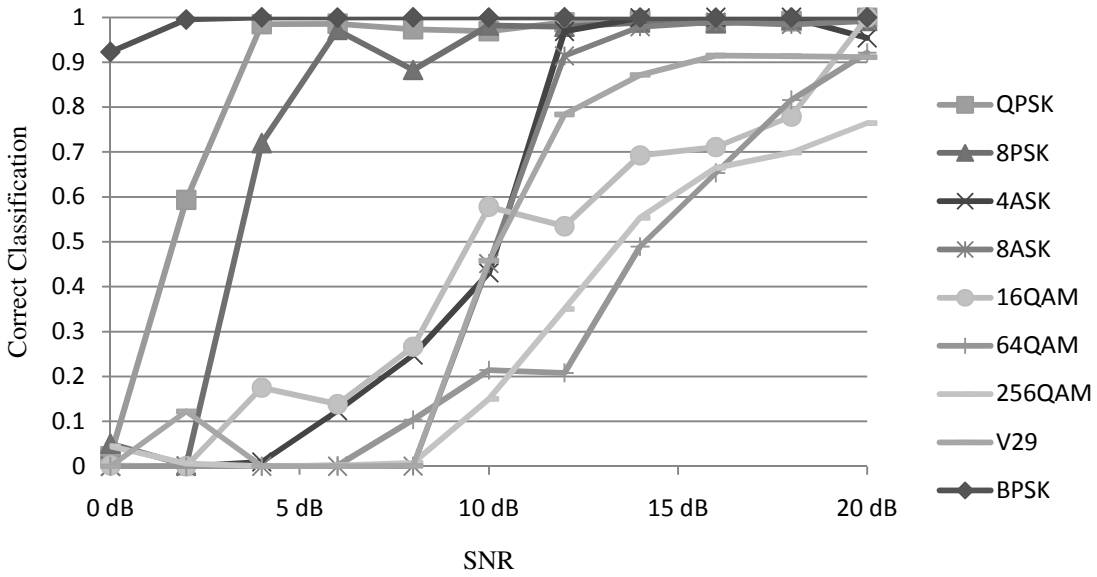

Figure 5-3. Classifier performance for individual modulations.

The initial bandwidth and center frequency are estimated by the wideband scanner. Once the receiver is centered at the correct frequency and the down sampling rates are set, the signal is turned off and a noise measurement is taken. This noise measurement is used to compute both the SNR while the signal is being classified and the noise power used to normalize the calculated cumulants. In this way, the operation is different than a normal wideband search: the noise estimate is much more accurate and reflects the theoretical approaches and therefore is better able to create a comparison to such work.

As can be seen in Figures 5-2 and 5-3, the classifier performs well at high SNRs, with the exception of 256-QAM, which never performs with greater than 80% accuracy. This outcome is attributed to the limited length of the data record. Cumulants require a good estimate of the entire constellation, and for the case of 256-QAM that is a large number of points which increase after differential processing. The greatest number of symbols that can be used in any classification attempt is 2048, which is only under the condition that the down sampling performed on the FPGA yields exactly 4 times the symbol rate.

Performance of the higher order modulations degrades dramatically below 10 dB. This degradation is due to the noise power estimate used in the normalization of the cumulant. As can be seen in Equation 2-2, as the signal power is reduced, the denominator approaches zero. Wild fluctuations in the estimate of the cumulant occur if the actual noise power drops during classification. As the denominator shrinks, the cumulants tend toward larger values, which favor the BPSK modulation as it has the largest cumulant values. This result shows that, in a live environment, noise characterization is extremely important for correct signal classification. Overall, the classifier performed the best for BPSK, with highly accurate classification at 5 dB. The other members of the PSK family also perform well at low SNRs.

Because noise has such an effect on the cumulant calculation, the probability density function of the noise is estimated and displayed in Figure 5-4. The distribution was then parameterized by assuming an alpha-stable distribution due to its bell-shape.

Using the techniques presented in [36], the parameters are found to be $\delta = 0.1018$, $\alpha = 1.88$, and $\gamma = 4.8$ which represent the location parameter, characteristic exponent, and dispersion, respectively. Because the estimated value of α is greater than 1.5, the estimated value has limited reliability. The estimation is assumed to be sufficient for the purposes of stating that the noise encountered by the classifier is not necessarily Gaussian, which has an α value of 2. This difference between the expected noise distributions is likely to have an effect on the cumulant estimate; however, the extent of the effect is a topic for further research.

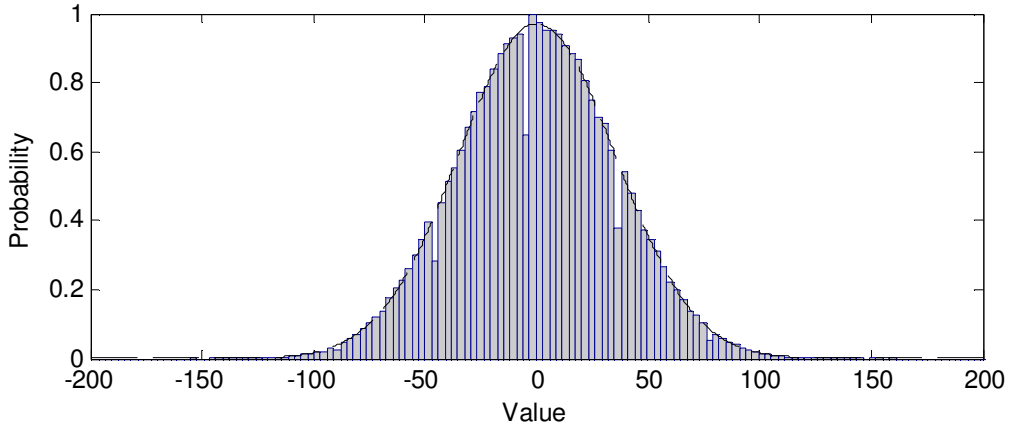

Figure 5-4. Noise distribution encountered by the classifier. Roughly 20,000 samples were used in the estimate. The estimated parameters were used to form the dashed line, which outlines the distribution.

An important characteristic from the end user's point of view is the reliability of the classification. Figure 5-5 depicts the probability of correct identification given that a particular modulation was detected. With the exception of the higher order QAM modulations, the reliability is very high. This discrepancy is a direct result of the limited length of the data set as mentioned earlier and can be remedied by grouping 64-QAM and all higher order QAMs into a single category. Also explained earlier is the tendency for classification to tend toward BPSK and other PSK modulations at low SNRs, which results in the low reliability of the M-PSK family at these low levels. However, both the M-PSK and V29 classification results become very reliable as the SNR reaches 5 dB.

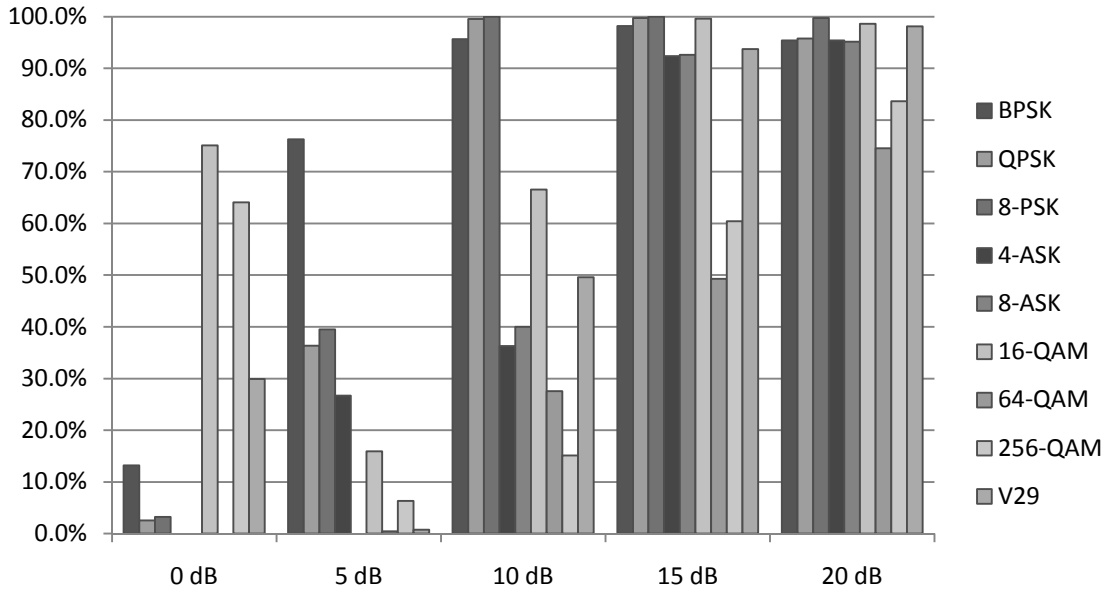

Figure 5-5. Probability of correct identification for a particular modulation.

5.2.2 Inter-Family Case

Classification needs not always correctly identify the specific constellation to provide valuable information. Another objective of classifiers is to determine to which family of modulations the signal belongs [4] [5] [37]. Figure 5-6 re-represents the data presented previously in terms of correctly identifying the class of modulations.

The classifier performs well for all modulation families at 15 dB; however, it also performs admirably at 5dB if the stand-alone case of V29 is ignored.

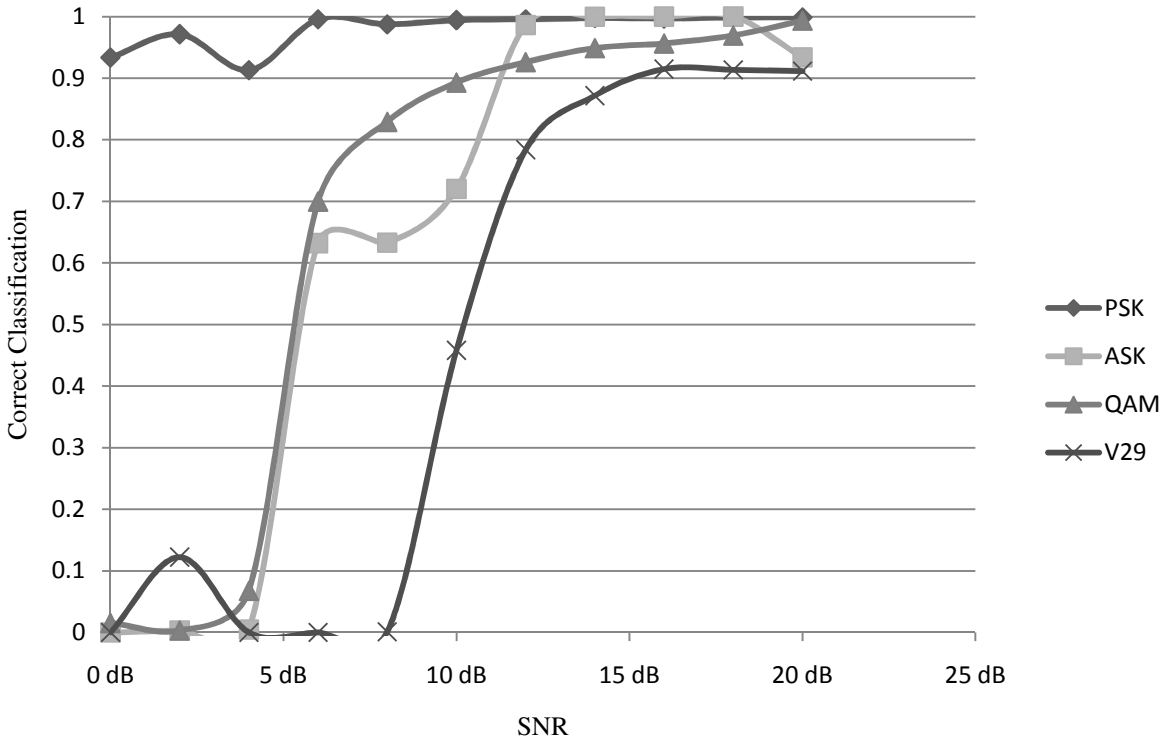

Figure 5-6. Family classification.

5.3 Energy Detector

The energy detector serves both as a wideband signal detector and a narrowband bandwidth estimator. In the former case, the detector must be capable of distinguishing signals from noise. It is desirable for the detector to function at low SNRs for wideband detection because, once the signal is identified, the detector bandwidth is re-tuned to approximate the signal bandwidth and the SNR may increase for classification.

A common method of characterizing a detector is to compare the probability of detection, P_d, and the probability of false alarm, P_{fa}. Because of the flexible nature of the detector, a complete characterization is not practical for this paper. Instead, a particular case using settings found useful in practice will be studied. The detector is set to compute an 8192-point FFT over a 50-MHz bandwidth. Each point is averaged over 100

samples and then combined into channels of two frequency bins each. This yields a channel bandwidth of approximately 12.2 kHz. The false alarm rate is found by recording the energy in a single bin and comparing the distribution to a threshold. This noise energy distribution is shown in Figure 5-7, which includes 30,000 data samples. The distribution is reminiscent of a Gamma distribution; however, attempts to find its parameters did not result in any respectable approximation. The distortion could be caused by the non-Gaussian nature of the underlying noise process.

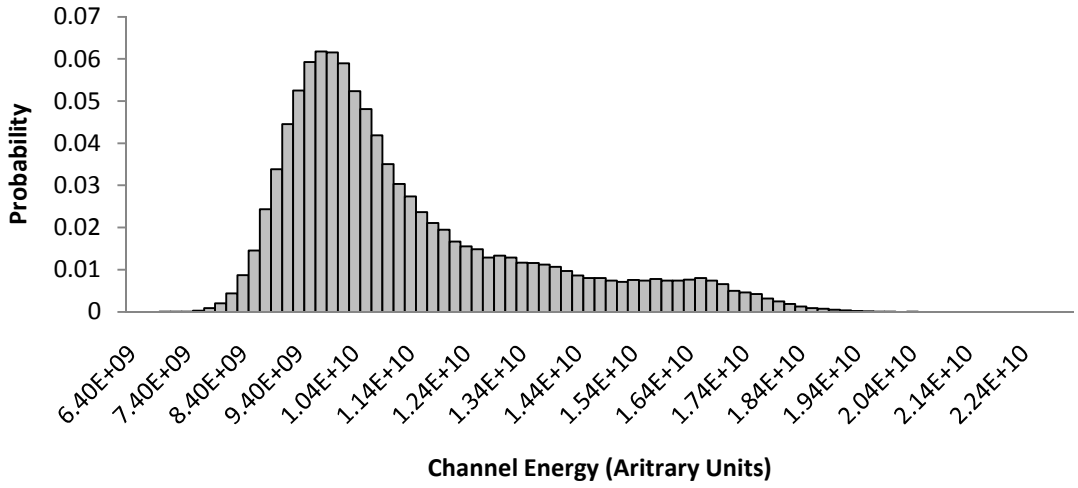

Figure 5-7. Channel energy distribution due to noise.

Once a set of false alarm rates are selected, the probability of detection is found by applying a sinusoidal signal in the observed frequency range. The signal is adjusted in power in 2-dB increments with 10,000 samples taken for each measurement. The results are shown in Figure 5-8: a signal at an SNR of -1.5 dB can be detected even with a low rate of false alarms at these practical settings.

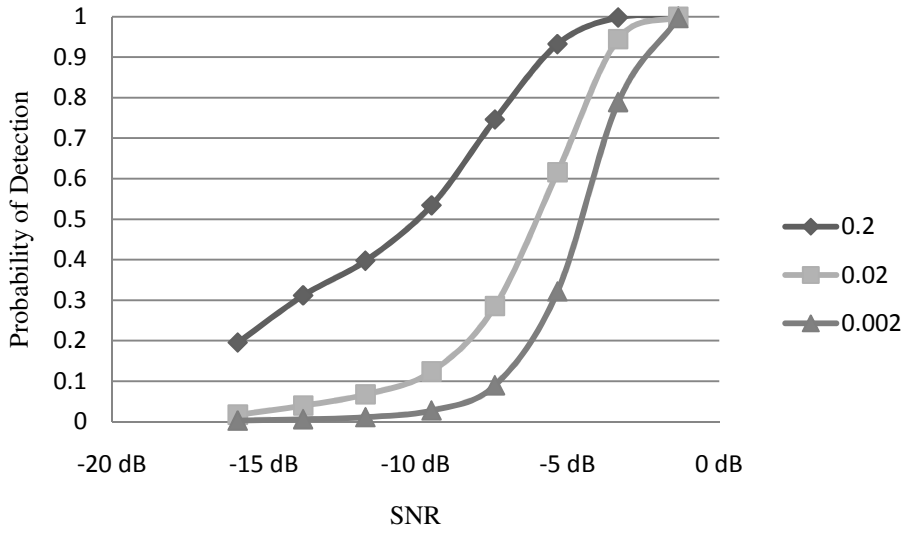

Figure 5-8. Probability of detection vs. SNR at various false alarm rates.

Chapter 6
Conclusions

6.1 Discussion of Results

The radio receiver described in this paper is capable of detecting signals and then automatically classifying the modulation if it is one of several common types—all done in real time. With adaptive modulation becoming popular as radios grow smarter, rapid classification becomes a necessity. These computations were performed in real time due to the combination of an FPGA and real-time processor with deterministic FIFOs providing a link between the two platforms.

While much research has been done in classification techniques, even with cumulants as a classification feature, this work shows how these theories interact in a practical application. Much of the research done previously using cumulants ignores problems such as frequency offset; however, this problem was shown to be a significant issue. The use of differential processing negated the effect of frequency mismatch but simultaneously changed all of the modulation constellations and, therefore, the cumulants of each. Therefore, the work in [5] [20] and [22] could not be directly implemented, at least if the precise center frequency is not known (as is the case in the completely blind classification presented here).

Other assumptions, such as accurate knowledge and the Gaussian distribution of noise power, are also not found to be practical. In a fluid spectral band, energy from spectrally close signals or spatially distant ones can cause changes as users enter and leave the band. These changes have a negative effect on classification, especially at low SNRs.

In a real-time application, the time spent classifying a particular signal should be minimized. This is a problem for large constellations, which require a large number of

samples just to estimate all of the possible constellation points, the size of which usually grows after differential processing. This performance issue is visible in the probability of classification for the M-QAM family, particularly 256-QAM. The PSK family is not influenced by the differential processing and normally is limited to a smaller constellation.

The energy detector is very useful both for wideband scanning of the spectrum and for estimating the bandwidth of any signals detected. This double functionality reduces the overall size of the algorithm without a decrease in speed.

The resulting signal classification system would be an asset to a software radio operating in an unlicensed band or as part of a radio network with adaptive modulation schemes. As adaptive radios become prevalent and spectrum use grows to be more flexible, the need for blind classification techniques will also increase.

6.2 Applications

The detection and classification system functions well over a large frequency band; however, this system is only able to classify signals with positive SNRs. This limitation leads toward implementation in a DSA radio, which would give the radio the ability to distinguish between primary and secondary users as well as to initiate contact with other radios. Passive blind classification would allow for communication to start without a control channel, even when a protocol that uses adaptive modulations is used.

The relatively high SNR required for classification compared to some other methods may be scoffed at by those in the military intelligence community; however, the real-time feedback should not be overlooked. Many algorithms that are capable of classification at very low SNRs, such as -20 dB, are also very computationally complex and thus are reserved for post-processing. This classification system gives real-time spectral feedback across a very large bandwidth.

6.3 Future Work

The classifier may be expanded by adding the capability to demodulate the signals. Once the modulation and symbol rate are known, a PLL can be used to maintain phase and frequency lock on the signal. The output bits are likely to be incorrect due to the phase uncertainty of the PLL; however, this can be accounted for in post-processing if the data protocol can be determined.

Hierarchal methods are often used with cumulant classification techniques. Restructuring the decision metric may increase the probability of correct classification, especially among modulation families. This method is best used in an environment where all possible modulations are known because changing the hierarchal tree is more complicated than the distance method used in this paper.

The most beneficial extension of this work is to move more of the classifier onto the FPGA. The record length for a single classification estimation is limited by the amount of data that can be sent from the FPGA to the NI PXIe-8130 RT Module. All processes with the exception of the rational resampler can easily migrate to the FPGA. The variable nature of the resampler complicates the filter specifications to prevent aliasing. A likely resampler design would be more restricted in terms of input factors, but this could be accommodated by further limiting the ratio approximation algorithm. The resulting variable record length for classification would greatly enhance the versatility of the system; fewer measurements could be taken in high-SNR environments or when only a few modulations are under test while large records could be used when signal strength is low or it is desirable to differentiate between high-order QAMs.

A final suggestion is to improve the noise estimation process by using median or myriad filters. These filters are useful in the event of impulse α-stable noise [**38**]. As demonstrated in Chapter 5, the cumulant estimates suffer at low positive values of SNR, even though simulation shows that classification at negative SNRs is possible. Part of the issue is the Gaussian assumption of the non-Gaussian noise as seen in Figure 5-6. Use of a more realistic estimation of the noise distribution may lead to higher quality classification.

References

[1] R. Morelos-Zargoza, R. Kohno K. Umebayashi, "Modulation Identification and Carrier Recovery System for Adaptive Modulation in Satellite Communications,".

[2] C. Cole, R. Krumland, M. Miller C. Weaver, "The Automatic Classification of Modulation Types By Pattern Recognition," Standford Unversity of California Stanford Electronics Labs, Technical Report AD0691069, 1969.

[3] L. Gadbois, P. Yansouni Y. Chan, "Identification of the Modulation Type of a Signal," in *IEEE International Conference on Accusic, Speech, and Signal Processing*, 1985, pp. 838-841.

[4] Jiadong Xu Luokun Liu, "A Novel Modualtion Classification Method Based on High Order Cumulants," in *International Conference on Wireless Communications, Networking and Mobile Computing*, 2006, pp. 1-5.

[5] B. Sadler A. Swami, "Hierarchal Digital Modulation Classification Using Cumulants," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 416-429, March 2000.

[6] H. Wang, Qiao Cai Xiaoqian Chen, "Performance Analysis and Optimization of Novel High-Order Statistic Features in Modulation Classification," *IEEE*, pp. 1-4, 2008.

[7] Kisbon Kim Andreas Polydoros, "On the Detection and Classification of Quadrature Digital Modulations in Broadband Noise," *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1199-1211, August 1990.

[8] Samir Soliman S. Hsue, "Automatic Modulation Classifciation Using Zero Crossing," *IEEE Radar and Signal Processing*, vol. 137, no. 6, pp. 459-464, December 1990.

[9] C. Le Martret D. Boiteau, "A General Maximum Likelihood Framework for Modulation Classification," in *IEEE International Conference on Acustics, Speech and Signal Processing*, Seattle, WA, 1998, pp. 2195-2168.

[10] A. Polydoros K. Kim, "Digital Modulatio Classification: The BPSK versus QPSK case," in *Military Communications Conference*, San Diego, CA, 1988, pp. 431-436.

[11] K. Chugg, A. Polydoros C. Long, "Further Results in Liklihood Classification of QAM Signals," in *IEEE Military Communications Conference*, Fort Monmouth, NJ, October 1994, pp. 57-61.

[12] J. Mendel Wen Wei, "Maximum-likelihood Classification for Digital Amplitude-Phase Modulations," *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 189-193, February 2000.

[13] Y. Chen, A. Czylwik L. Haring, "Automatic Modulation Classification Methods for Wireless OFDM Systems in TDD Mode," *IEEE Transactions on Communciations*, vol. 58, no. 9, pp. 2480-2485, September 2010.

[14] J. Sills, "Maximum-likelihood Modulation Classificaiton for PSK/QAM," in *IEEE Military Communications Conference*, Atlantic City, NJ, 1999, pp. 217-220.

[15] F. Liedtke, "Computer Simulation of an Automatic Classification Procedure for Digitally Modulated Communication Signals with Unknown Parameters," *Signal Processing*, vol. 6, no. 4, pp. 311-323, August 1984.

[16] Nhi Ta, "A Wavelet Packet Approach to Radio Signal Modulation Classification," in *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, Philadelphia, PA, 1994, pp. 508-511.

[17] Jun-Ho Choi, Sun-Phil Nah, Wong Jang, Dae Young Kim Cheol-Sun Park, "Automatic Modulation Recogition of Digital Signals using Wavelet Features and SVM," in *International Conference on Advanced Communcation Technology*, Gangwon-Do, 2008, pp. 387-390.

[18] William A. Gardner, "The Cumulant Theory of Cyclostationary Time-Series, Part I: Foundation," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3387-3408, December 1994.

[19] C. Le Matret, J. Lacoume P. Marchand, "Classification of Linear Modulations By a Combination of Different Orders of Cyclic Cumulants," in *IEEE Signal Processing Workshop on Higher-Order Statistics*, 1997, pp. 47-51.

[20] S. Li, C. Song, F. Chen Lei Shen, "Automatic Modulation Classification of MPSK signals Using High Order Cumulants," in *ICSP*, 2006, pp. 1-4.

[21] V. Reddy V. Chaithanya, "Blind Modulation Classification in the Presence of Carrier Frequency Offset," in *International Conference on Signal Processing and Communications*, Bangalore, 2010, pp. 1-5.

[22] Yi Gong, Yong Guan Qinghua Shi, "Asynchronous Classification of High-Order QAMs," in *IEEE Wireless Communications and Networking Conference*, Las Vegas, NV, 2008, pp. 1188-1193.

[23] Chad Spooner, "Classification of Co-channel Communciation Signals using Cyclic Cumulants," in *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 1995, pp. 531-536.

[24] Y. Bar-Ness, W. Su O. Dobre, "Higher-Order Cyclic Cumulants for High Order Modulation Classification," *IEEE*, pp. 112-117, 2003.

[25] Z. Krsmanovic, N. Remenski P. Petrov, "An Automatic VHF Signal Classifier," Elektronska Industriia R&D Institute, Belgrad, Yugoslavia, White Paper.

[26] B. Pavic, V. Tadic V. Matic, "The Implementation of Digital Signal Processing for Automatic Recognition of Radio Emission and Spectrum Occupancy Analysis," in *IEEE International Caracas Conference on Dvices, Circuits and Systems*, Cancun, 2000, p. T55/1.

[27] A. Cabric, R. Brodersen A. Tkachenko, "Cyclostationary Feature Detector Experiments Using Reconfigurable BEE2," in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, Dublin, 2007, pp. 216-219.

[28] Wei Su, MengChu Zhu J. Xu, "Software-Defined Radio Equipped with Rapid Modulation Recognition," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1659-1667, May 2010.

[29] William Hammond, "Continued Fractions and the Euclidean Algorithm," University at Albany, Albany, Lecture 1997.

[30] John Proakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 2nd ed., John Griffin, Ed. New York, United States: Macmillian Publishing Company, 1988.

[31] Martin Oerder and H. Meyr, "Digital Filter and Square Timing Recovery," *IEEE Transatctions on Communications*, pp. 605-612, May 1988.

[32] Joachim Speidel Jianxin Wang, "16QAM Symbol Timing Recovery in the Upstream transmission of DOCSIS Standard," *IEEE Transactions on Broadcasting*, vol. 49, no. 2, pp. 211-216, June 2003.

[33] Claudia Gonner, Klaus Spitzer Thomas M. Lehmann, "Survey: Interpolation Methods in Medical Image Processing," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049-1074, November 1999.

[34] Fred Harris, "Performance and Design of Farrow Filter Used For Arbitrary Resampling," Communication Systems and Signal Processing Institute, San Diego, Paper.

[35] N. Anane and M. Anane H. Bessalah, "Multifunction Generator using Horner Scheme and Small Tables," *ICM*, vol. 9, no. 11, pp. 224-227, Dec 2003.

[36] Chysostomos Nikias, George Tsihrintzis, "Fast Estimation of the Parameters of Alpha Stable Inpulsive Interference," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1492-1503, June 1996.

[37] Shue-Zen Hsue Samir Soliman, "Signal Classification using Statistical Moments," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 908-916, May 1992.

[38] G. Arce J. Gonzalez, "Optimality of the Myriad Filter in Practical Impulsive-Noise Environments," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 438-441, February 2001.