Adaptive Classification of Interfering Signals in a Shared Radio Frequency Environment

by

Ganesh Nachiappa Ramaswamy

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science

and

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February, 1992

Signature of Author .....
Department of Electrical Engineering and Computer Science
December 3, 1991

Certified by .....
Pierre A. Humblet
Professor of Electrical Engineering
Thesis Supervisor (MIT)

Certified by .....
David F. Bantz
Manager, Portable Systems
Thesis Supervisor (IBM Research)

Accepted by .....
Campbell L. Searle
Chairman, Departmental Committee on Graduate Students

Adaptive Classification of Interfering Signals in a Shared Radio Frequency Environment

by

Ganesh Nachiappa Ramaswamy

Submitted to the Department of Electrical Engineering and Computer Science on December 3, 1991, in partial fulfillment of the requirements for the degrees of Bachelor of Science and Master of Science

Abstract

In a shared radio frequency environment, where several intentional and unintentional transmitters are expected to co-exist, the ability to identify the source of an arbitrary interfering signal, using an automated diagnostic tool, is desirable. Treating the subject as a pattern recognition problem in a radio frequency environment, a viable scheme for the classification of the interfering signals, with adaptive learning capability, has been developed. The scheme incorporates an architecture for signal acquisition, a strategy for feature extraction, and algorithms for signal classification and learning. To lay the background for current and future work in the subject, a categorization of interfering signals consisting of six categories has been proposed and mathematical models for representative examples from the six categories were constructed. Performance of the proposed scheme with respect to hardware and software system parameters was evaluated through Monte Carlo simulations.

Thesis Supervisor (MIT): Pierre A. Humblet
Title: Professor of Electrical Engineering

Thesis Supervisor (IBM Research): David F. Bantz
Title: Manager, Portable Systems

In Loving Memory
of Both of My Grandfathers

ACKNOWLEDGEMENTS

My time with the Portable Systems Group at the IBM Thomas J. Watson Research Center has been a very rewarding experience. I am particularly indebted to David Bantz, my manager for all three of my internship assignments and one of the two supervisors of my thesis, for all the valuable lessons that I had learnt from him. Without exaggeration, I can say that he has made an engineer out of me.

I extend my sincere gratitude to Pierre Humblet, my MIT thesis supervisor, for his understanding and continuous support throughout the six months. I will especially remember the friendly smile with which he always greeted me, the friendly smile that is otherwise hard to find at MIT.

Since the project undertaken was primarily theoretical, it was necessary for me to get periodic feedback on my work. Conversations with Chia-Chi Huang, another member of the Portable Systems Group, were very helpful in this respect.

One of the key aspects of the project was the modelling of some of the known interfering signals. Since no empirical measurements were made for the project, actual measured data obtained from variety of sources played an important role in the construction of the models. I would like to acknowledge Jonathan Cheah (Hughes Network Systems), Hap Patterson (Sensormatic Electronics Corporation), Ken Blackard (Virginia Tech), Ted Rappaport (Virgina Tech), and Bob Cato (IBM Raleigh) for sharing their knowledge concerning some of the known interfering signals with me.

I am grateful to Rick Kaufman (Centralized Scientific Services, IBM Research) for his input on commercial radio frequency components, and to Anand Narasimhan (RPI) for his suggestions concerning time-frequency analysis during the early stages of the project.

I am also grateful to all staff of Project Agora at IBM for their cooperation during

critical moments of the project and for tolerating my obnoxious use of CPU time, and to all staff of the IBM Research Library for their continuous support in helping me obtain the literature needed for the project.

Throughout my stay at MIT and IBM, I was backed by the support and encouragement of my family in Malaysia, and my friends at MIT and IBM. The support was essential in my remaining sane during the difficult four-and-a-half years.

To all of you - *Terima kasih, berribu-ribuan terima kasih*[1] !!

[1]Thank you, thank you thousands of times!! (Malay)

Contents

List of Figures

List of Tables

Introduction

Upon reading the title of the thesis, several questions may come to the mind of the reader. *What* is a shared radio frequency environment? *Why* are we interested in adaptive classification of interfering signals? *How* can we adaptively classify the interfering signals?

The first two questions will be answered in this introductory chapter. The answer to the third question will be the subject of the thesis.

Description of the Problem

A shared radio frequency environment is a public-use frequency band where several intentional and unintentional transmitters are expected to co-exist. Unlike a conventional frequency band with one permitted user, where the interference is usually treated as Gaussian noise, the problem of interference is much more complicated in a shared frequency band because of the presence of several permitted users, transmitting a wide variety of signals, including *transient* or otherwise time-varying signals.

Examples of shared frequency bands are the 902-928 MHz, 2400-2,483.5 MHz, and 5,725-5,875 MHz bands[2] in the United States [61]. Allocations for spectrum use in these bands have been made by the Federal Communications Commission (FCC), under the provisions for Radio Frequency Devices (Part 15), Industrial, Scientific, and Medical Equipment (Part 18) and Amateur Radio Stations. Transmitters operating under Part 15 require no license, but are neither afforded interference protection from

[2]Hereinafter these three frequency bands will be referred to as the bands centered at 915 MHz, 2.44 GHz, and 5.8 GHz, respectively.

existing and future licensed operations nor from any other Part 15 devices.

In addition to the indoor radio networks employing spread spectrum signalling, authorization for which was recently provided under Part 15 of the FCC rules [55], a variety of intentional and unintentional transmitters use all or part of the three bands. Examples of such users are amateur radio stations, electronic article surveillance systems, microwave ovens, photocopiers, elevator switches, garage door openers, toy walkie-talkies, RF welding equipment, diathermy, and other communications and non-communications equipment [6], [14], [41], [50], [57].

Interference from other sources using the same frequency band is considered to be one of the major impairments to the successful operation of indoor radio systems [34]. However, our concern in this thesis is not with methods to counter such interference, since this subject is discussed in many recent textbooks on spread spectrum systems, including [74].

Our focus in this project will be to develop a scheme to identify the source of an arbitrary interfering signal[3]. Such a scheme can then be incorporated into an *automated diagnostic tool*, capable of performing interference diagnosis. The diagnostic tool can be used in several circumstances, for example to survey a given environment prior to installing a radio network (perhaps to find a suitable location for the base station of the network), or to identify the source of interference when a radio link fails due to interference (such that a remedy could be found), or to monitor a shared radio frequency environment on a regular basis and avoid potential radio link failures. The diagnostic tool will provide advice automatically, relieving the pressure to perform the diagnosis manually, and eliminating the need for an expert human engineer.

We have motivated the need for interfering signal classification. But why are we interested in making the classification process *adaptive*? If the classification process is specific to only a given set of interfering signals in a given environment, the attractiveness of the diagnostic tool will disappear when new sources of interference

[3]In addition to interference diagnosis for shared radio frequency environments, the developed scheme could also be adapted to identify illegal users in radio frequency environments with one user, or to perform interference diagnosis for other problems involving transient and random signals, like interference diagnosis for power-lines.

are discovered. Such discovery of new interference, or a new behavior of a known interference is not at all unlikely since the spectrum in use is a shared spectrum for which no license is required, and users may appear and disappear unpredictably. Hence, adaptive learning capability of the tool would allow the user to update the tool whenever necessary, with minimal input from the user.

Therefore, we set our objectives as to develop a scheme for interfering signal classification, with adaptive learning capability. Such a scheme will incorporate a viable architecture for the diagnostic tool and the necessary algorithms to perform the learning and diagnosis. In order to motivate the future hardware implementation of developed architecture and algorithms, the performance of the proposed scheme and the significance of some of the system parameters will be characterized analytically and through simulations.

Contributions of the Thesis

The problem to be addressed in the thesis could be described as a pattern classification problem, in a radio frequency environment, involving a variety of signals exhibiting transient and random behavior. Unfortunately, no published prior work has been found for such a problem. The only classification problem that appears to address the radio frequency environment is the problem of *modulation recognition.* Modulation recognition, which is in fact a subset of the more complex problem of interfering signal classification, does not involve difficulties such as the transient nature of many interfering signals, frequency hopping phenomenon, difference in bandwidth between the interfering signals, and general random behavior of most interfering signals. So, the techniques used in modulation recognition, details of which will appear later in the report, are not directly applicable to our problem.

The contributions of the thesis consist of both *engineering contributions* and *academic contributions.* The engineering contributions of the thesis are:

- The development of an appropriate high-level architecture for the diagnostic tool, that can be supported by the current state of technology.

- The investigation of the sensitivity of system performance with respect to the system parameters.

The academic contributions of the thesis are:

- Categorization of the interfering signals and the modelling of several known interfering signals, which provides the background for current and future work in the subject.
- Development of a strategy for feature extraction, to allow the adaptive classification of a wide variety of signals
- The development of the learning and classification stages, and the associated identification of a viable decision rule among known decision rules for pattern classification, such that a simple algorithm for the adaptive learning process is possible.

The outcome of the project consists of a viable scheme for interfering signal classification, whose performance has been verified through simulations, and a set of initial system parameters to provide the background for the future hardware implementation of the proposed scheme.

Organization of the Report

The remainder of this report is divided into two parts. Part I reports the development of the architecture and algorithms for interfering signal classification, and Part II illustrates the performance evaluation of the proposed system that was performed by simulating the interfering signals and the learning and classification stages.

Part I consists of four chapters. In Chapter 1, the development of the architecture for adaptive signal classification is reported, with particular emphasis on the signal acquisition techniques. In Chapter 2, the term *interfering signal* is defined, a categorization of the interfering signals consisting of six categories is proposed, and mathematical models are constructed for examples drawn from each of the categories.

Chapter 3 addresses the topic of a suitable strategy for feature extraction, using the background provided by the models of Chapter 2. The selection of the six critical features is described and several other features are proposed for future extensions. Chapter 4 concludes Part I by illustrating the process of identifying a suitable decision rule, and the development of the learning and classification stages.

Part II also consists of four chapters. Chapter 5 describes the system implementation through simulation, which primarily involves the discussion on the software packages that were written for the Monte Carlo simulation of interfering signals, and for the implementation of the learning and classification stages. In Chapter 6, a model for the simulation process is discussed, and the results of initial experiments performed to validate the scheme, and to understand the significance of the features extracted, are reported. Chapter 7 continues with the experiments by evaluating the system performance with respect to software and hardware system parameters. Chapter 8 concludes the report by summarizing the accomplishments of the thesis and providing directions for future work.

There are two appendices to the report. Appendix A contains a description on an optional hardware module that would improve the performance of the system when the encountered interfering signal is a narrowband signal. Appendix B contains the actual documented software packages written for the performance evaluation.

Part I:

Architecture and Algorithms

Chapter 1

Architecture for Adaptive Signal Classification

We begin this chapter by discussing the basic functional specifications for the diagnostic tool proposed in the introduction to this report, and we derive a scheme for interfering signal classification that would satisfy the requirements of the tool. Then we illustrate the challenge involved in the signal acquisition, and motivate the use of the quadrature sampling of bandpass signals. We proceed to develop a viable hardware architecture for adaptive signal classification, identifying the significant system parameters that will be later addressed in this report. Appendix A of this report concerns an optional hardware module that can be used to improve the performance of the system if several types of narrowband interfering signals exist.

1.1 The Concept of a Diagnostic Tool

In this section we discuss the proposed concept of a diagnostic tool for interference diagnosis, which is the primary application of the signal classification architecture and algorithms to be developed in this project. Understanding the requirements of the tool would facilitate the development of the necessary system design theory.

We would like the tool to have the capability of correctly classifying the observed signal if the signal has been previously encountered by the tool, and the capability of

adaptive learning if the signal is new. We therefore conclude that the diagnostic tool should have two operating modes - the *Learning* mode and the *Diagnostic* mode.

The Learning mode will be used during the installation of the tool in a new environment, and whenever a new source is found. The Learning mode should require minimal supervision from the user. In this mode, the tool will collect data from the source in consideration, compute the necessary parameters to identify the source in the future and store these parameters in a system library. The internal processing involved will not be transparent to the user.

During the Diagnostic mode, the tool captures the signal encountered, derives a parametric representation of the data, compares it with the models stored in the system library, and finally provides the user with the classification of the signal. The tool will have the capability of estimating the likelihood of the diagnosis being correct, and if the probability happens to be lower than a predetermined threshold, it will declare a *no diagnosis* state, which may correspond to the discovery of a new source, or equivalently, the discovery of a new behavior of a known source. If the user desires, he may switch to the learning mode at this point to update the system library. The no-diagnosis state may also correspond to several other situations, for example excessive background noise, or the presence of more than one interfering signal[1].

The diagnostic tool may take one of several forms. It could be a *stand-alone diagnostic tool*, in which case it will have its own built in memory and processing power. It could be a *diagnostic sub-system* that can be interfaced to a personal computer, in which case the sub-system will have the necessary hardware for signal acquisition and storage, but will use the processing facilities of the host system. The diagnostic sub-system can also be incorporated into a host system in the form of a communications test set (for an example of a test set, see [31]), which would otherwise not have the hardware or the software needed to perform interference diagnosis.

Further development of functional specifications for the tool is beyond the scope of this project. Based on the discussion in the preceding paragraphs, we propose the

[1] In this project we will be concerned with classification when only one interfering signal is present. The extension of the work to include classification when multiple sources are present will be discussed briefly in Chapter 8, as a suggested topic for future work.

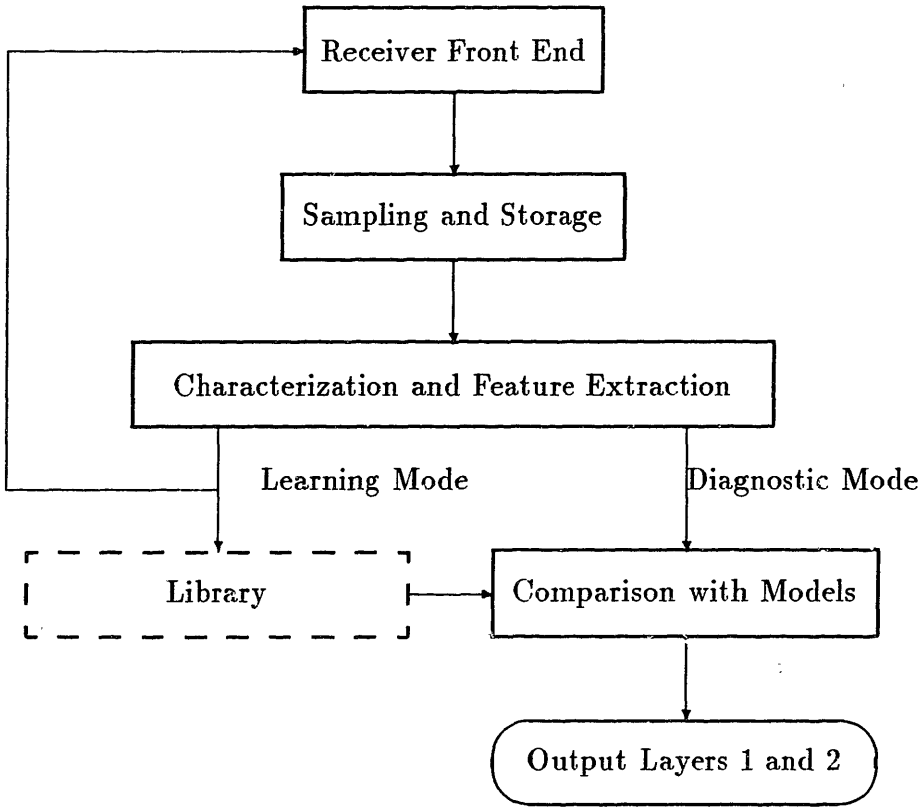

Figure 1-1: Proposed Scheme for Interference Diagnosis

scheme illustrated in Figure 1-1 for the interference diagnosis.

We recognize that the functions of the tool fall into two broad categories: *signal acquisition* and *signal analysis.* Signal acquisition is a hardware problem, and in Figure 1-1, the modules marked *Receiver Front End* and *Sampling and Storage* fall in this category. The Receiver Front End is responsible for capturing the signals in the band of interest and submitting the signal to the Sampling and Storage module. The Sampling and Storage module, as the name implies, is responsible for sampling and storage of the acquired signals.

Signal analysis is an algorithmic problem. The two modules in Figure 1-1, marked as *Characterization and Feature Extraction* and *Comparison with Models* will be employed for the purpose of signal analysis. Although the two modules could be implemented strictly in hardware, we prefer to implement them in software. The

Characterization and Feature Extraction module is responsible for transforming the acquired data (which may consist of several thousand samples), for the purpose of dimension reduction, into a feature vector consisting of features extracted from the acquired data. The Comparison with Models module is responsible for comparing the received feature vector with models stored in the system library and provide an output. The output will have two layers. *Output Layer 1* will consist of the identification of the most likely source among the models currently in the system library, or the no-diagnosis state if the likelihood of the most likely interfering signal is lower than a predetermined threshold. *Output Layer 2* provides the actual values of the features computed (which will otherwise not be transparent to the user), for independent evaluation of the diagnosis by the curious user.

As can be seen from Figure 1-1, all four of the modules are involved in the Diagnostic mode. In the Learning mode, only the first three of the modules are used. The *loop* in the Learning mode indicates that repeated measurements will be made in order to characterize the distribution of the feature vectors obtained. Therefore, there is an intermediate processing of the acquired feature vectors prior to storing the information in the Library, which we refer to as the *adaptive learning* process. This processing stage is not explicitly shown in Figure 1-1.

The system design issues that are algorithmic in nature will be discussed in Chapters 3, and 4. In the following section, we address the challenge of an appropriate sampling technique which can then be used to develop a suitable hardware architecture for diagnostic tool.

1.2 Strategy for Sampling

1.2.1 The Need for a Sampling Strategy

It is common to use a spectrum analyzer with a broadband antenna to perform radio frequency measurements [4], [22], [69]. Unfortunately, such an approach to signal acquisition is insufficient for our purpose. Most commercially available spectrum analyzers are *scanning* analyzers (also known as *non-real-time* analyzers) that is not

tuned to the entire spectrum in consideration at once, but only to a single frequency at one time. The analyzer scans through the spectrum, and since it must wait to tune to a frequency, the phenomenon under test must be repetitive or it may not be detected [30]. For example, consider a unmodulated carrier that is *hopping* in frequency, thereby virtually occupying a wider bandwidth. In order to accurately capture this signal, the spectrum analyzer's sweep rates have to be synchronized with respect to the hopping rate of the signal. There are two problems with this. First, the spectrum analyzer may not have a high enough sweep rate to accommodate high hopping rates of the signal. Second, *prior* knowledge of the hopping pattern of the signal is necessary to establish sweep rates that are synchronized, and since our purpose is to classify an unknown signal, such knowledge of time-varying behavior of interfering signals will not be available prior to the classification.

The purpose of the above discussion is primarily intended to motivate the fact that in order to accurately capture an unknown signal, which can virtually be anywhere in a given band, the *entire* band has to be captured simultaneously. We recall that the bands that are of interest to us are the three bands, centered at 915 MHz, 2.44 GHz, and 5.8 GHz. Since the higher band of 5.8 GHz is currently limited by the cost of technology for consumer products [14], we focus our attention on the two lower bands. Also, since the the 2.44 GHz band has a bandwidth of 83.5 MHz, higher than the bandwidth of the 915 MHz band, we realize that by setting our target for the 2.44 GHz band, the obtained solution can be easily adapted for the 915 MHz band.

We state here the famous sampling theorem due to Shannon (1949):

Theorem 1.1 *If a function f(t) contains no frequencies higher than W cps it is completely determined by giving its ordinates at a series of points spaced (1/2W) s apart.*

A proof of the theorem, and its extension to the case of random signals, could be found in [36]. Therefore, by direct application of Theorem 1.1 for the 2.44 GHz band, which has frequency components up to 2.4845 GHz, a sampling rate of almost 5 GHz would be required. Such high sampling rates are unfortunately beyond the scope of the current technology.

Fortunately, since the signal is bandlimited to the permitted frequencies, either because the signal is naturally bandlimited, or because the signal has been bandpass filtered to remove the spectral components that are not in the band of interest, we found solutions to the sampling problem by means of bandpass sampling theorems.

1.2.2 Direct Sampling of Bandpass Signals

Even though a bandpass signal is bandlimited, *directly* sampling a bandpass signal is more complicated than a bandlimited lowpass signal, because two spectral bands are involved in the case of bandpass signal, one centered at the positive center frequency of f_o and another centered at $-f_o$. Since sampling produces replicas of the original spectrum [56], appropriate choice of sampling frequency is necessary to avoid aliasing.

There are several theorems that have been discussed in theory that can be used in the selection of the appropriate sampling frequency. One such theorem is the *first-order sampling theorem* for bandpass signals, stated below, where the signal is directly sampled at a lower rate than that predicted by Theorem 1.1.

Theorem 1.2 *For a bandpass signal y(t) having spectral components (in Hz) only in the range $f_o - W \leq |f| \leq f_o + W$, where $f_o \geq W$, minimum required sampling rate (in Hz) to determine the signal for all values of time, by direct sampling of y(t), is given by*

$$f_{s(min)} = \frac{2(f_o + W)}{k + 1} \tag{1.1}$$

where k is the largest nonnegative integer satisfying

$$k \leq \frac{f_o - W}{2W} \tag{1.2}$$

For a proof and a detailed treatment of the theorem, see [11]. Discussions on direct sampling of bandpass signals could be found in many recent textbooks, including [12], [56], [60], [67].

We notice that $f_{s(min)}$ in equation (1.1) takes values in the range $[4W, 8W)$, with the minimum value of $4W$ achieved when (1.2) is satisfied with equality. The value of $4W$ is precisely twice the bandwidth of the signal, which is the same as the re-

quired minimum sampling frequency for a lowpass signal of equivalent bandwidth. Generalization of the first-order sampling of bandpass signals to second and higher order sampling, where two or more interleaved sequences of equispaced sampling is performed [36], [60], results in a more efficient sampling rate with the minimum rate of twice the bandwidth applicable to any value of W.

Theorem 1.2 provides, in theory, a method for sampling a bandlimited high frequency signal at a much lower rate than that predicted by Theorem 1.1, and this is particularly useful when f_o is much larger than W. The signal directly reconstructed from the samples of the direct sampling process will correspond a lowpass signal, with a bandwidth of $\frac{1}{2}f_{s(min)}$, but with the knowledge of f_o, the original bandpass signal can be determined by means of frequency shifting [60].

However, many practical issues arise in implementing this method in hardware. We shall assume, for the convenience of discussion, that (1.2) is satisfied with equality and the sampling rate of $4W$ is applicable. By employing the direct sampling technique, we will essentially be making an analog-to-digital converter (ADC) running at a rate of $4W$, thereby *expecting* to see a signal that varies in the order of $2W$, to digitize a bandpass signal which has frequency components up to $f_o + W$.

First, we realize that any jitter in the clock input to the ADC will result in an increased error in quantization because the signal is varying much faster than the sampling rate. Second, there will be a further increase in the sampling error due to the *aperture time* constraints of real ADCs . For example, the AD9028[2], a high speed flash 8-bit ADC capable of sampling rates up to 300 MSPS, has an aperture delay of 1.4 ns, and an aperture uncertainty of 3 ps (rms) [2].

The aperture time is the interval between the application of the *hold* command and the actual opening of the switch within the ADC, and consists of a delay and an uncertainty. While there are methods to compensate for the aperture delay, by means of advancing the hold command by the known value of aperture delay, the aperture uncertainty poses the ultimate limitation. The maximum frequency f_{max} which can be handled with less than one least significant bit (LSB) error, is related

[2]This is the fastest 8-bit converter listed in [2]

to the number of bits per sample, n, and the aperture uncertainty, τ_{au}, by [2]

$$f_{max} \leq \frac{2^{-n}}{\pi \tau_{au}} \tag{1.3}$$

For the AD9028, the maximum frequency which can be handled with less than 1 LSB error is computed to be approximately 400 MHz, which is much less than frequencies up to 2.4835 GHz that may be encountered in the 2.44 GHz band.

Direct sampling of bandpass signals is not impossible to implement, but implementation using current hardware technology may lead to errors in sampling that are higher than the error that would be encountered in the case of a lowpass signal with equivalent bandwidth. We realize that this error is due to the presence of high frequency components in the bandpass signal, and if we could *downconvert* the bandpass signal to a lower frequency band, then the error rate would be reduced. This option, leading to another sampling theorem for bandpass signals, will be discussed in the next section.

1.2.3 Quadrature Sampling of Bandpass Signals

Now, we consider the option of preprocessing the bandpass signal prior to sampling. The preprocessing takes the form of downconverting the bandpass signal to two equivalent baseband signals, then sampling these two signals at the rate prescribed by Theorem 1.1.

We state the *quadrature sampling theorem for bandpass signals* as:

Theorem 1.3 *A bandpass signal $y(t)$ having spectral components (in Hz) only in the range $f_o - W \leq |f| \leq f_o + W$, where $f_o \geq W$, can be determined from samples of its two equivalent quadrature baseband components, each sampled uniformly at $2W$.*

Proof: Any bandpass signal $y(t)$ may be written as [12], [64]

$$y(t) = x(t)\cos[2\pi f_o t] + \phi(t)] \tag{1.4}$$

With $x_I(t) = x(t)\cos[\phi(t)]$ and $x_Q(t) = \sin[\phi(t)]$,

$$y(t) = x_I(t)\cos[2\pi f_o t] - x_Q(t)\sin[2\pi f_o t] \tag{1.5}$$

Since $\cos[2\pi f_o t]$ and $\sin[2\pi f_o t]$ are real functions, $x_I(t)$ and $x_Q(t)$ will be real provided that $y(t)$ is real. Further, since $y(t)$ is has a bandwidth of $2W$ centered around f_o, $x_I(t)$ and $x_Q(t)$ will be lowpass signals bandlimited to $[-W, W]$. By Theorem 1.1, $x_I(t)$ and $x_Q(t)$ can each be uniquely determined by sampling each of them at $2W$. With the knowledge of f_o, $y(t)$ can be uniquely determined from the samples of $x_I(t)$ and $x_Q(t)$, each sampled at a rate $2W$. □

For an alternate derivation of the theorem, see [10]. Since the applicability of Theorem 1.1 has been shown for the case of random signals [36] and since we have only used Theorem 1.1 to prove Theorem 1.3, we realize that Theorem 1.3 can be applied to the case of random bandpass signals.

We will refer to $x_I(t)$ and $x_Q(t)$ as the *in phase* (I) and *quadrature* (Q) components of $y(t)$. By using conventional techniques of downconversion [42], where the bandpass signal $y(t)$ is multiplied with the signals $2\cos(2\pi f_o t)$ and $-2\sin(2\pi f_o t)$ from a local oscillator and then lowpass filtered to retain only the frequency components in the range [-W, W], we can obtain $x_I(t)$ and $x_Q(t)$, respectively.

We notice that, as opposed to direct sampling of bandpass signals at a rate of $4W$ (or higher), now we are able to sample the *two* quadrature components at $2W$ *each*. For the 2.44 GHz band, the bandwidth of $2W$ corresponds to a value of 83.5 MHz, which can be achieved by many of the high speed ADCs currently available in the market, including the AD9028 encountered before. By downconverting the bandpass signal into two baseband quadrature components, we have solved the problem induced by the presence of high frequency components. However, we have also introduced additional hardware into the system by the option, and hence additional system parameters.

Having found a suitable sampling strategy, we proceed to discuss a hardware architecture for the diagnostic tool in the next section.

1.3 Proposed Hardware Architecture

The requirements introduced by the need to perform adaptive signal classification introduces no additional hardware complexity. The signal acquisition stage is identical in both the Learning mode and the Diagnostic mode, with the only hardware difference being the repeated measurements needed to characterize the distribution of feature vectors in the Learning mode, which can be achieved by generating an appropriate control signal in software. Therefore, the *same* hardware can be used for signal acquisition in both of the modes.

There is an added advantage in using the same hardware for both modes, besides being cost efficient. Since the signal acquired in the Learning mode will be subject to the same system parameters (say, for example, the number of bits per sample), as the signal acquired in the Diagnostic mode, the characterization of the distribution of feature vectors in the Learning mode will be a more accurate description of the feature vectors likely to be encountered by a given diagnostic tool in the Diagnostic mode.

Although detailed hardware design for the diagnostic tool is beyond the scope of the paper, we would like to at least outline a high level architecture for the tool. We recall that there were four modules in the scheme proposed in Figure 1-1. We would like to reduce these four modules into three hardware stages, with the two software modules of Characterization and Feature Extraction, and Comparison with Models, combined into a single hardware stage of *Processing Unit.* In addition to containing the two software modules, the Processing Unit will also be responsible for part of the *control* of the signal acquisition process on the one end, and the *user interface* on the other end. The three hardware stages are discussed in the following subsections.

1.3.1 The Receiver Front End

The Receiver Front End takes two inputs, a BAND SELECT control signal from the Processing Unit to indicate the frequency band that should be captured (either the 915 MHz, 2.44 GHz, or the 5.8 GHz band), and an ATTENUATOR INPUT control signal

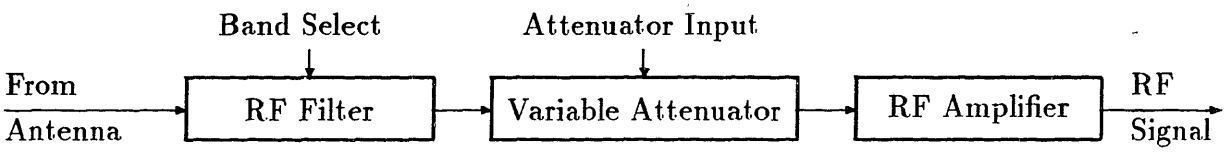

Figure 1-2: The Receiver Front End

from the Sampling and Storage module, to set the appropriate attenuation level for the encountered signal. It has one output, which is the received Radio Frequency (RF) signal, submitted to the Sampling and Storage module.

A possible implementation of the Receiver Front End is shown in Figure 1-2. The reader should note that the receiver continuously receives RF signals, with possible changes in the attenuation level as prescribed by the ATTENUATOR INPUT control signal, provided that the tool has been turned on, and the band has been selected. Therefore the Receiver Front End need not be aware of the mode of operation.

The band-selection could be achieved by having three bandpass filters with the 3dB cut-off frequencies set at the edges of the three bands we discussed before[3] and by employing the BAND SELECT control signal to choose the appropriate filter.

The attenuator is present to adjust the relative strength of the received signal such that the full dynamic range of the analog-to-digital converter in the Sampling and Storage module can be utilized. The attenuator also ensures that the RF amplifier is not driven into saturation. Since different interfering signals may require different attenuation levels, the attenuator should be a variable attenuator. A programmable attenuator may be used for this purpose, provided that its settling time is short compared to the duration of occurrence of the interfering signals that are of interest. If such an attenuator is not available, then several different attenuators, set at different attenuation levels, may be used in parallel, with the selection of the appropriate attenuator made by the ATTENUATOR INPUT control signal.

[3]Here we are assuming that it is sufficient to observe only the activities within the band, and we will not be concerned with *out of band emissions* from neighboring bands.

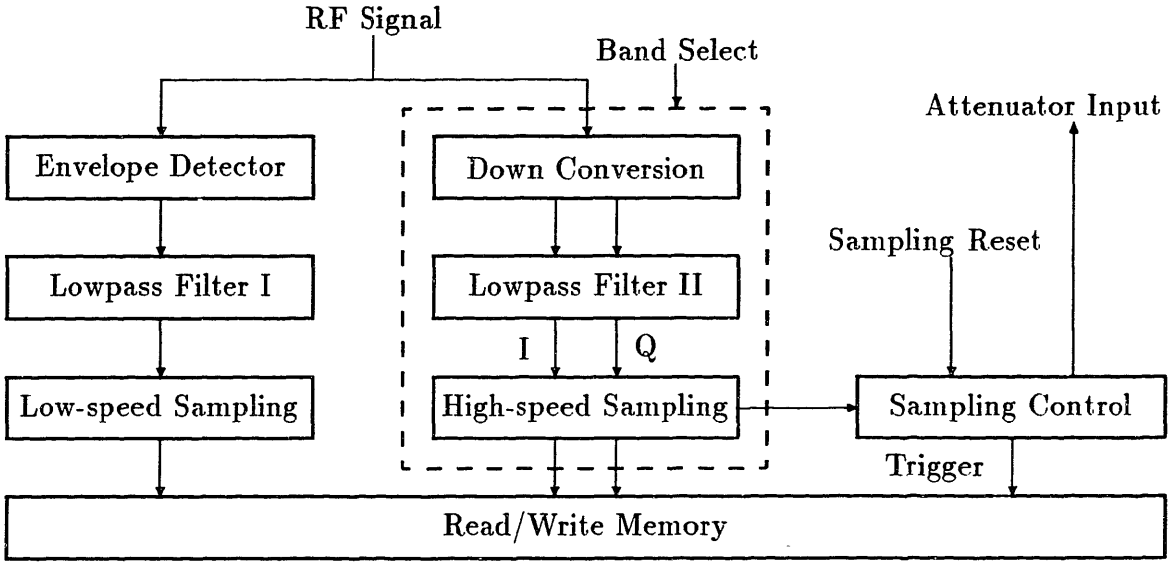

Figure 1-3: Sampling and Storage

Further discussion on the components of a typical radio frequency receiver could be found in handbooks and textbooks on radio frequency surveying, including [69].

1.3.2 Sampling and Storage

The Sampling and Storage module takes the RF signal from the Receiver Front End, and the control signals BAND SELECT and SAMPLING RESET from the Processing Unit as its inputs, and generates the ATTENUATOR INPUT control signal for the attenuator in the Receiver Front End, and the SAMPLING COMPLETE interrupt signal for the Processing Unit, as its outputs. Like the Receiver Front End, the Sampling and Storage module will also be not aware of the mode of operation, and will faithfully sample and store the signals received whenever the SAMPLING RESET is set by the Processing Unit. Such continuous sampling allows flexible triggering.

A possible implementation of the module is shown in Figure 1-3. The generation of SAMPLING COMPLETE signal is not shown in the figure. Also not shown in the figure is the additional control circuitry to control the transfer of data between the

Processing Unit and the Sampling and Storage Module that may become necessary.

In Section 1.2, we saw how a lowpass signal bandlimited to $[-W, W]$ needs a sampling rate of only $2W$. However, this is only in the ideal case, and in practice usually a higher rate would be required. *Oversampling* by a factor of 2, thereby employing a sampling rate of $4W$ as opposed to the ideal sampling rate of $2W$, is typically recommended [44]. While a higher sampling rate leads to a greater accuracy in digitization, it also makes the acquisition system operate at a higher speed, requiring more memory for the same duration of observation, thereby making the system more expensive. Since further verification of the need for oversampling requires hardware experiments, which is beyond the scope of this project, we shall assume that the factor 2 of oversampling is applicable.

The *wideband*[4] *IQ signal acquisition* shown in the dotted box in Figure 1-3 is a direct implementation of Theorem 1.3. The BAND SELECT control signal from the Processing Unit is used to select the appropriate downconversion frequency, lowpass filter bandwidth and the sampling rate, since these three values will be different for different frequency bands. A sinusoid at f_o, the center frequency of the band of interest, from a local oscillator can be used as an input to a 90^o power splitter, which will generate $\cos 2\pi f_o t$, and $\sin 2\pi f_o t$. These two signals can be multiplied with the incoming RF signal in a frequency mixer, and lowpass filtered (using Lowpass Filter II in Figure 1-3) to retain only the frequency components in the range $[-W, W]$ where $2W$ is the bandwidth of the band being captured. The resulting I and Q signal, as defined in Section 1.2.3 of this chapter, can be digitized using a high speed ADC as discussed before. Taking into consideration the factor 2 of oversampling discussed above, a sampling rate of 167 MHz is suitable for the 2.44 GHz band.

The reader may be surprised to see the blocks labelled *Envelope Detector*, *Lowpass Filter I* and *Low-speed Sampling*, in Figure 1-3. These three blocks will be used to acquire what we shall refer to as the *slow-varying envelope* of the encountered interfering signal. As we shall see in Chapter 2, there are devices such as Microwave

[4]Appendix A describes an optional hardware module that can be added to the system to improve performance in the special case where the signal encountered is a *narrowband* signal. So, we use the word *wideband* here to distinguish between the two hardware modules

Ovens that emit RF energy only during one half of the 60 Hz cycle of the power supply. Hence, we are interested in detecting the presence or absence of a "square wave" at 60 Hz, the slow-varying envelope, which is a piece of information which we suspect would be useful in identifying the signal. Since the 60 Hz frequency corresponds to a period of 16.7 ms, in order to compute the slow-varying envelope in software from the wideband I and Q samples that are sampled at a high frequency, a huge amount of sampling memory (in the order of Megabytes) will be needed. So, we have elected to compute the slow-varying envelope in hardware using an envelope detector, a lowpass filter, and an ADC sampling at a much slower rate. The envelope detector is a rectifier circuit, commonly described in many textbooks, including [42]. Ideally, we would like to the slow-varying envelope to be a square wave at 60 Hz for signals from devices exhibiting the 60 Hz behavior, and a pure DC value for other signals. However, a large bandwidth will be required to accurately characterize a square wave [42], and this will unfortunately permit high frequency components to corrupt the slow-varying envelope. But since we are only interested in an *approximate* shape of the envelope, we realize that capturing up to, say, the fifth harmonic of the square wave should be sufficient. Therefore, the time constant of the envelope detector should be sufficiently large to remove all high frequency components, and to further ensure that only the frequency components below the fifth harmonic of the square wave remains, we include a lowpass filter with a 3 dB bandwidth of 300 Hz. The slow envelope can be sampled at 1200 Hz, corresponding to a factor 2 of oversampling, and the duration of observation needs to be only 16.7 ms, which is the period of a 60Hz signal, thereby requiring only 20 samples per observation.

The block labelled *Sampling Control* is responsible for generating the ATTENUATOR INPUT control signal to set the attenuation level in the Receiver Front End, and the TRIGGER signal to start storing the sampled signal. After the band has been selected and the SAMPLING RESET has been set, the module will be continuously sampling the received signal. Initially the attenuation level should set to a low level, such that any signal above the mean thermal noise may be detected. By observing a small number of samples, the amplitude of the received signal will be compared to

a threshold value, and if the threshold is exceeded then an interfering signal will be assumed to be present. Then, based on the first few samples observed, the Sampling Control will make a *guess* as to what the appropriate level of attenuation should be, and after a delay corresponding to the settling time of the attenuator, the TRIGGER signal will be generated to start the storage of the samples. Upon completion of the storage, the SAMPLING COMPLETE interrupt signal will be sent to the Processing Unit.

There are several system parameters in the Sampling and Storage module that may have a significant influence on the performance of the diagnostic tool. With respect to the slow envelope acquisition, we have essentially resolved, in theory, the issue of the lowpass filter bandwidth, and the sampling rate. Further verification of the method and the analysis related to the system parameters have to be performed through actual hardware experiments. Likewise, in the case of the wideband I and Q signal acquisition, again we have resolved the issue of the lowpass filter bandwidth and the sampling rate, leaving further verification (involving issues like the possible non-linearity of the mixer, and the non-ideal behavior of the filters) up to future hardware experiments. In the case of the Sampling Control, we only outlined a general scheme that can be used to control the sampling process, and we realize that a significant amount of design and verification has to be performed. In particular, the appropriate threshold for comparison of the signal amplitude, the exact number of samples to be used in generating the control signals, and the criteria for generating the control signals are expected to important.

There are three other parameters of the system that we would be able to analyze in theory and through simulations. These parameters are:

- The dynamic range of the sampling system
- The duration of observation
- The sensitivity of the system to frequency and phase jitters from the local oscillator circuitry used in the downconversion process

The dynamic range issue affects both the slow envelope acquisition and the wide-

band I and Q signal acquisition. The remaining two issues are related to only the wideband I and Q signal acquisition, since we have resolved the issue of duration of observation (i.e. 20 samples at 1200 Hz) for the slow varying envelope, and there is no downconversion involved.

The dynamic range issue will be addressed from the viewpoint of the number of bits per sample required for acceptable performance, keeping in mind that the attenuator can be used to utilize the maximum dynamic range of the ADC. The duration of observation is related to both the sampling rate and the record length (the number of samples per observation), but since we have set the sampling rate (recall the factor 2 of oversampling), only the record length is a variable in our investigation. Since the local oscillator circuitry used in the IQ downconversion process may exhibit frequency and phase jitters, we would like to model these jitters as stochastic processes to gain an understanding of how sensitive the system performance will be to the non-ideal behavior of the system. These topics will be addressed in Part II of this report.

1.3.3 The Processing Unit

As we saw in Section 1.1 of this chapter, the diagnostic tool can take one of several forms, and the required hardware for the Processing Unit will be different for each implementation. In any case, the Processing Unit should have the memory needed to contain the system library, but since the feature vector computed is expected to be much smaller in dimension than the actual data, only a small storage space will be needed. The memory allocated for the system library should allow both READ and WRITE operations, since data will be written during the Learning mode and read during the Diagnostic mode.

In addition to the primary responsibility of performing the necessary signal analysis during the two modes of operation, the Processing Unit will also be responsible for the control of the tool (with the exception of the ATTENUATOR INPUT control signal generated by the Sampling and Storage Module), and for the interface with the user.

After the entry of the desired inputs from the user, the Processing Unit will generate the BAND SELECT and SAMPLING RESET signals previously discussed. Upon

receiving the SAMPLING COMPLETE interrupt signal from the Sampling and Storage module, the Processing Unit will read in the data acquired, and perform the necessary signal analysis. The Processing Unit will have the knowledge of N, the number of independent measurements needed for the Learning Mode, and is solely responsible for generating the required number of SAMPLING RESET signals and for keeping track of the number of measurements completed at any given time during the Learning Mode. Another responsibility of the unit is to update the system library after the adaptive learning process. The Processing Unit is also responsible for providing the two output layers (the results of the diagnosis) through the selected used-interface.

Hardware parameters of the Processing Unit relate to the processing speed of the of the system, which is a *customer satisfaction* issue, and therefore will not be addressed in this project. However, there are several software parameters that affect the performance of the system, including the required number of independent measurements to be made during the Learning mode, the specific features that should be included in the feature vector, the algorithm for adaptive learning and classification of the interfering signals, and the value of p_{th}, the threshold value for the likelihood of the most likely signal, below which the diagnostic tool should declare the no-diagnosis state.

Since the Processing Unit is the only module that is aware of the mode of operation, the requirement of making the signal classification adaptive introduces severe constraints in the form of appropriate feature selection methodology and classification algorithms. Theoretical issues related to the selection of these parameters will be discussed in detail in Chapters 3 and 4, and the related performance evaluation will be discussed in Part II of this report.

1.4 Chapter Summary

This chapter began Part I of this report, by discussing several issues related to the architecture for adaptive signal classification. We introduced the reader to details concerning the proposed application of the architecture and algorithms developed in

this project, in the form of a diagnostic tool for interference diagnosis. We recognized the difficulties in signal acquisition, and we explored strategies for sampling. We then discussed the details of the Receiver Front End, and the Sampling and Storage module. We also discussed briefly the hardware aspects of the Processing Unit which incorporates the two software modules of Characterization and Feature Extraction, and Comparison with Models. Since the design of these two modules are primarily algorithmic, the necessary algorithms will be developed in Chapters 3 and 4. In Chapter 2, we will proceed to develop models of interfering signals, keeping in mind that the signal acquisition architecture will affect the form of the received signal. The system hardware system parameters identified in this chapter will be further addressed in our performance evaluation process, which is reported in Part II. Appendix A of this report contains a brief description of an optional hardware module that could be added to improve the performance when the encountered signal occupies a bandwidth much narrower than the bandwidth of the captured band.

Chapter 2

Interfering Signals: Definition, Examples, and Mathematical Models

The objective of this chapter is to gain an understanding of how a variety of received interfering signals, obtained through a signal acquisition architecture employing Theorem 1.3 of Chapter 1, would look like in its equivalent baseband representation. We begin this chapter by defining the term *interfering signal* and proposing six categories of interfering signals. We then develop a general baseband representation of the received interfering signals, and proceed to model examples drawn from the six categories of interfering signals in the desired baseband representation. These models will be both used in motivating appropriate feature selection methodology in Chapter 3, and for performance evaluation in Part II.

2.1 Definition and Categorization

2.1.1 What is an Interfering Signal?

In this project, we will not be concerned with out-of-band emissions of transmitters form neighboring bands, although these emissions exist as a major electromagnetic

compatibility problem [68]. We will also not concern ourselves with natural radio noise sources such as atmospheric, solar, and cosmic noise sources. Therefore our focus will be entirely on permitted users of a given shared frequency band.

In a shared radio frequency environment, where there are more than one legal users, *interfering signals* can only be defined with respect to a given user. However, it should be noted that interference is a problem only to *intentional transmitters* who emit radio frequency energy for the purpose of transmitting information, and therefore depend on the safe reception of the transmitted signal, such as indoor radio local area networks (radio LANs), amateur radio, garage door openers, and electronic article surveillance devices. The *unintentional transmitters* of the shared spectrum, that are either functionally dependent upon the radiation power (like radio-frequency stabilized arc welders), or happen to radiate electric energy because it is less expensive for the manufacturer of the equipment to accept its presence than to suppress it (for example, photocopiers and elevator switches), typically do not face the problem of interference.

So, we define *interfering signals* from the standpoint of the destination of a given transmission.

Definition 2.1 *"Interfering Signals" in a shared radio frequency environment are all components of permitted signals that are present in a given frequency band, having sufficient power above the mean thermal noise to be detected by the receiver of a given user, with the exception of the intended signal to be received by the given user.*

For radio LAN operations in the 915 MHz and 2.44 GHz bands interfering signals would include signals from sources like amateur radio, electronic article surveillance devices, microwave ovens, photocopiers, elevator switches, garage door openers, toy walkie-talkies, diathermy, radio-frequency stabilized arc welders, and other non-communication equipment [6], [14], [41], [50], [57]. Although most of the work in this project will focus on interfering signal classification for wireless LANs, the algorithms and architecture developed could easily be adapted for other users, as we will see in Chapter 8.

2.1.2 The Six Categories of Interfering Signals

To perform an exhaustive search of all possible interfering signals and their characteristics is not within the scope of our project. Therefore, we will divide the interfering signals into several categories, and choose a representative example from each category for further analysis.

With respect to categorizing the interfering signals, we found the work of Middleton to be particularly inspiring. Middleton has divided impulsive electromagnetic interference arising from non-Gaussian random processes into three classes, and the work has been reported in several publications, including [53], [68], [73]. The three classes of interference are Class A, which consists of noise that is "typically narrower spectrally than the receiver in question, and as such generates ignorable transients in the receiver's front end when a source emission terminates"; Class B, where, "the bandwidth of the incoming noise is larger than that of the receiver's front-end-stages, so that transient effects, both in the build-up and decay occur, with the latter predominating"; and Class C, which "is the sum of Class A and Class B interference," [53]. Middleton further developed statistical models for the three classes of interference, assuming that, " the locations of the various possible emitting sources are Poisson-distributed," and " the emission times of the possible sources are similarly Poisson-distributed in time," [53].

We realize, however, that such Poisson-distributed impulsive noise is only a subset of all possible interfering signals in a shared radio frequency environment. Therefore, we developed a more comprehensive categorization for the interfering signals, implicitly using Middleton's idea of categorizing the signals according to their bandwidth.

In comparing the bandwidth of the interfering signal, we will use the official allocations made by the FCC for radio frequency devices (under Part 15) mentioned in the introduction to this report[1]. Since the receiver bandwidth, as we decided in

[1]The bandwidth allocated for ISM equipment (Part 18) appears to be somewhat different than that allocated for radio frequency devices (Part 15) in the 2.44 GHz band. ISM equipment are allowed to operate at 2450 MHz, with a tolerance of ± 50 MHz, whereas the band allocated for radio frequency devices is 2400-2483.5 MHz [61]. So, one can expect emissions from Part 18 equipment that may appear to be "out-of-band" from the viewpoint of Part 15 devices.

Chapter 1, is equal to the official bandwidth allocations made by the FCC, one could also interpret the comparison of bandwidths as being between the signal bandwidth and the receiver bandwidth. We first introduce three broad categories: signals with bandwidth that is much less than the bandwidth allocated for the band it occupies, signals with bandwidth that is in the order of the bandwidth allocated for the band it occupies, and signals that have a much larger bandwidth than the bandwidth allocated to the band in consideration[2]. We shall refer to these three categories as Type A, Type B, and Type C, respectively.

Type A interfering signals can be further categorized into signals that are pure tone signals, and signals that have a larger bandwidth than a pure tone signal (but much smaller than the bandwidth of the band it occupies). We will refer to these to subcategories as Type A1, and Type A2, respectively. One could think of the Type A2 signals as modulated Type A1 signals. An example of a Type A1 signal would be signals transmitted by Electronic Article Surveillance Devices, and an example of Type A2 signal would be narrowband transmissions from amateur radio stations.

Type B signals, which occupy a bandwidth that is in the order of the allocated bandwidth, can also be further categorized into two sub-categories. Interfering signals may occupy a large bandwidth, either because they are narrowband signals that exhibit *frequency hopping*, or because they may naturally have a large bandwidth at all times. We will refer to these two subcategories as Type B1 and B2, respectively. Type B1 signals have a narrow local (or short-term) bandwidth, but a wide global (or long-term) bandwidth. Both the local and the global bandwidths of Type B2 signals are wide. An example of Type B1 signal would be frequency hopped spread spectrum transmissions from amateur radio stations, and an example of Type B2 signal would be emissions from a microwave oven.

Type C signals, which occupy a bandwidth that is much larger than the bandwidth of the band in consideration, can also be further categorized into two subcategories. These signals could either be impulsive with pulse durations sufficiently narrow in time

[2]Signals that have a bandwidth that is greater than the width of a given band are probably occupying more than one of the permitted bands, and as such we use the phrase "band in consideration" to compare that bandwidth of these signals, as opposed to "band it occupies"

that the emissions extend through several of the permitted bands (with suppressed emissions in the *forbidden* bands that occur in between the permitted bands), or these signals could non-impulsive signals (for example, the classical wideband Gaussian noise). These two subcategories would be referred to as Type C1, and Type C2, respectively. An example of Type C1 signal would be emissions from a photocopier. Although thermal noise is generally Gaussian [18], but by Definition 2.1, thermal noise cannot be considered as an interfering signal, and we have not found a suitable alternate example for Type C2 signals. However, we will still include a theoretical model for wideband Gaussian noise, since it appears to be a good approximation for many naturally occurring phenomena [20].

We formally state the six categories of interfering signals in Definition 2.2.

Definition 2.2 *The six categories of interfering signals in a shared radio frequency environment are:*

- *Type A1: Single tone signals.*
- *Type A2: Signals with bandwidth larger than single tone signals, but much smaller than the bandwidth of the band occupied; typically a modulated Type A1 signal.*
- *Type B1: Frequency hopping signals that have a narrow local bandwidth but a wide global bandwidth, thereby occupying most or all of the bandwidth allocated.*
- *Type B2: Signals that occupy most or all of the bandwidth allocated at all times when they are present; typically a modulated signal with a constant carrier frequency.*
- *Type C1: Signals that occupy a bandwidth much greater than the bandwidth of the band in consideration, due to a highly impulsive random process.*
- *Type C2: Signals that occupy a bandwidth much greater than the bandwidth of the band in consideration, other than Type C1 signals.*

We will select one representative example from each category and develop mathematical models for them, which will be used in our analysis in subsequent chapters.

2.2 Background for Model Construction

There are three reasons why we are interested in constructing models for the interfering signals. First, construction of such models would provide a deeper understanding of the physical and statistical characteristics of the interfering signals, which we will need in order to derive the appropriate feature extraction methodology for signal classification. Second, these models can be used to generate test cases for performance evaluation, which will be addressed in Part II of this report. Third, these models would further contribute to the understanding of the behavior of the interfering signals, which can be used in the successful deployment of indoor radio local area networks.

We realize that the models constructed should represent the signals likely to be received by the signal acquisition architecture described in Chapter 1. In order to reduce the complexity of the models, we will assume that the channel frequency response, $H(f)$, is represented by $H(f) = 1$ for all frequencies of interest, and therefore the channel exhibits no multipath effects. The signal-to-noise ratio (due to the channel and receiver noise) will first be assumed to be very large to facilitate the development of the models. However, in implementing the models for performance evaluation, finite signal-to-noise ratios will be introduced in Chapter 6.

Since we have assumed the application of Theorem 1.3 in deriving the architecture for signal acquisition, the received signal will consist of in-phase and quadrature components, which we will generally refer to as the *equivalent complex baseband* form of the actual bandpass signal, in this chapter and subsequent chapters. The in-phase and quadrature components will be the real and imaginary parts, respectively, of the equivalent complex baseband representation.

We make the following proposition for the form of the received signal.

Proposition 2.1 *The received signal from an interfering source, obtained through an architecture employing Theorem 1.3, can be written in the equivalent complex baseband form $R(t)$ given by*

$$R(t) = s(t)r(t)e^{i[2\pi\Delta f(t)t+\Delta\theta(t)]} \tag{2.1}$$

where $s(t)$ and $r(t)$ represent the real-valued slow-varying and rapid-varying envelopes, and $\Delta f(t)$ and $\Delta\theta(t)$ represent offsets between the carrier frequency (if any) and phase of the interfering signal and the local oscillator.

Proof: Let us rewrite equation (1.4),

$$y(t) = x(t)\cos[2\pi f_o(t)] + \phi(t)] \tag{2.2}$$

Here, $y(t)$ is the (bandpass) interfering signal, bandlimited either because it is naturally bandlimited, or because it has been bandpass filtered in the receiver. We know form the proof of Theorem 1.3, that the in-phase and quadrature components, $x_I(t)$ and $x_Q(t)$, of the equivalent baseband representation of the bandpass signal are given by $x_I(t) = x(t)\cos[\phi(t)]$, and $x_Q(t) = x(t)\sin[\phi(t)]$, respectively. Since the real part of $R(t)$ is the in-phase component, and the imaginary part of $R(t)$ is the quadrature component, we write

$$R(t) = x(t)\cos[\phi(t)] + ix(t)\sin[\phi(t)] = x(t)e^{i\phi(t)} \tag{2.3}$$

We could always write the envelope $x(t)$ as a product of a slow-varying envelope $s(t)$, and a rapid-varying envelope $r(t)$. The motivation for writing $x(t)$ as a product, and a formal definition of $s(t)$ will both appear in Definition 2.3. In the case where the interfering signal is modulated, its carrier frequency need not be the same as the center frequency f_o of the band, which is the frequency we use for downconversion. Hence there may be an offset $\Delta f(t)$ in the two frequencies[3], which may be a function of time because the carrier frequency may be a function of time (for example, in the case of frequency hopping signals). Likewise there will be an offset $\Delta\theta(t)$ in the phase, due to the difference in the phase of the carrier and the phase of the local oscillator, which again could be a function of time, since the phase of the interfering signal could be a function of time (for example, in the case of frequency or phase modulated signals). So, we conclude that $\phi(t)$ in equation (2.3) could be written as

[3] $\Delta f(t)$ will be zero when there is no modulation, like in the case of Type C signals.

$\phi(t) = 2\pi\Delta f(t)t + \Delta\theta(t)$. The result follows. □

In Chapter 1 we illustrated a method through which the slow-varying envelope $s(t)$ could be estimated in hardware. Now we would like to propose a formal definition for $s(t)$. We realize that a radio frequency device that has been powered on need not continuously transmit radio frequency energy. A classic example would be the emissions from microwave ovens, which occur only during one half of the power supply cycle of 60 Hz, and a violation to this rule has *has never been observed* [15]. So, there is a *deterministic* component to the general stochastic behavior of radio frequency emissions. Hence, we make the following definition for $s(t)$.

Definition 2.3 *The slow varying envelope of an interfering signal, from a source that has been powered on, is the deterministic component of the envelope of the emission, indicated by a level 0 when emissions are known to to be absent, and by a level 1 otherwise.*

For sources other than microwave ovens, such deterministic behavior is usually not known, and therefore we expect to see $s(t) = 1$ in most cases. But estimating $s(t)$ in practice is a very difficult problem, and the method proposed in Chapter 1 may not be optimal. We leave the issue of developing methods for a more accurate acquisition of $s(t)$ to future hardware designers.

2.3 Examples and Mathematical Models

We have identified the four parameters, $s(t), r(t), \Delta f(t)$, and $\Delta\theta(t)$ as being critical in constructing models for interfering signals. Hence, we are in search of statistical characterizations of these four parameters for each of the six categories of interfering signals.

In performing such an investigation, we will assume that the band of interest is a hypothetical shared band with a center frequency of f_o and a bandpass bandwidth of $2W$, to make our work independent of any actual band and to facilitate extending our findings to applications other than interference diagnosis for radio LANs. However,

we will draw examples from the 915 MHz and the 2.44 GHz bands, when useful data concerning a given source is available.

2.3.1 Type A1 Interfering Signal

The example of a Type A1 interfering signal that we will discuss in detail is the signal from an Electronic Article Surveillance Device (EASD), in the form of a radio detection system, that is often used for anti-theft purposes in places like shopping malls and libraries. The description in the following paragraph was obtained from [59].

The radio detection system operates in the 915 MHz band. In one form of its implementation, a packaged diode is attached to the articles in the store. The shoplifter, who leaves the store with the packaged diode still attached to the article, passes between two antennas, one of which is radiating at a given frequency in the 915 MHz band and, at the same time, under a metallic plate radiating at a much lower frequency, say around 100 kHz. The diode mixes the two frequencies by rectification and the resultant sum or difference frequency is the telltale sign giving away the presence of the diode on the article.

For the purpose of constructing a model for the EASD in our hypothetical band, we will assume that a theft never occurs during the duration of observation. Since the lower frequency from the metallic plate is not within the band of interest, we are concerned with only the pure tone signal from the antennas. We assume that this signal can be modelled as a single tone signal.

Therefore, for the EASD, $s_{EASD}(t) = r_{EASD}(t) = 1$, and both $\Delta f_{EASD}(t)$ and $\Delta\theta_{EASD}(t)$ will not be a function of time. Since it is quite likely that we will not know the operating frequency of the device, we will assume that the operating frequency will be uniformly distributed in $[f_o - W, f_o + W]$, and hence Δf_{EASD} will be uniformly distributed in $[-W, W]$. Similarly $\Delta\theta_{EASD}$ will be assumed to be uniformly distributed in $[0, 2\pi]$.

2.3.2 Type A2 Interfering Signal

The example of Type A2 interfering signal that we will model is a frequency modulated (FM) signal from an amateur radio station. First, let us write an expression for a typical FM signal [67]:

$$y_{FM}(t) = A\cos[2\pi f_c t + k_f \int_0^t g(t)dt] \tag{2.4}$$

The carrier frequency is f_c, which need not be the same as the center frequency f_o of our hypothetical band, and the maximum frequency deviation is $k_f|g(t)|_{max}$. We will model $g(t)$ as a Gaussian noise bandlimited to audio frequencies (approximately 20 kHz).

Once again, we find that $s_{FM}(t) = r_{FM}(t) = 1$. We assume that such an amateur radio station uses a constant carrier frequency during a given duration of transmission. However, the carrier frequency need not be the same for different transmission periods, and need not be the same for different amateur radio stations, and therefore could take any value in $[f_o - W, f_o + W]$. Hence, we model $\Delta f_{FM}(t)$ as an unknown constant, uniformly distributed in $[-W, W]$ for a given observation. The phase offset $\Delta\theta_{FM}(t)$ will be a function of time, given by

$$\Delta\theta_{FM}(t) = k_f \int_0^t g(t)dt + \theta_o \tag{2.5}$$

where θ_o is uniformly distributed in $[0, 2\pi]$. Appropriate values for k_f will be chosen for our model in Chapter 5, keeping in mind that k_f should only be a small fraction of $2W$, in order for the model to belong to Type A2 interfering signal.

2.3.3 Type B1 Interfering Signal

Type B2 interfering signals are typically frequency hopped spread spectrum signals. Authorization for both direct sequence and frequency hopped spread spectrum transmission has been given to amateur radio stations [55], and it appears that radio amateurs were among the commercial users of spread spectrum [65].

Now, we would like to model the example of a Type B2 interfering signal based on amateur radio transmissions employing Single Sideband, where the carrier frequency is changed periodically, and we will refer to this model as Frequency Hopped Single Sideband (SSB/FH) transmission.

Since we cannot expect to see any deterministic absence of the interfering signal, we conclude that $s_{SSB/FH}(t) = 1$, and we assume that the carrier is modulated with $g(t)$, Gaussian noise bandlimited to audio frequencies. The carrier frequency will be a function of time due to the frequency hopping phenomenon, and we will assume that the hop duration is uniformly distributed in $[t_1, t_2]$. Suitable values for t_1 and t_2 will be chosen in Chapter 5, for repeated generation of this interfering signal. The hop duration, once chosen, does not vary for a given observation.

We write equation (2.1) in a slightly different form for this model:

$$R_{SSB/FH}(t) = r(t)e^{i\theta(t)}e^{i[\Delta f(t)+\Delta\theta_o]} \tag{2.6}$$

where $r(t)e^{i\theta(t)}$ is an *analytic* signal[4] associated with $g(t)$, the message to be transmitted. We could further write

$$r(t)e^{i\theta(t)} = g(t) + i\mathcal{H}[g(t)] \tag{2.7}$$

where the notation $\mathcal{H}$ indicates *Hilbert Transform.* So $r_{SSB/FH}(t)$ is the magnitude of the right hand side of equation (2.7), and $\Delta\theta_{SSB/FH}(t) = \theta_{SSB/FH}(t) + \Delta\theta_o$, where $\theta_{SSB/FH}(t)$ is the argument of the right hand side of equation (2.7), and $\Delta\theta_o$ is a constant phase offset that is uniformly distributed in $[0, 2\pi]$. $\Delta f_{SSB/FH}(t)$ will be a function of time, periodically changing from one value that is uniformly distributed in $[-W, W]$, to another value similarly distributed, with the period given by the hop duration, which is yet another random variable (because, in general we will not know the hop duration chosen by the radio amateur), as mentioned in the previous paragraph. Further details will appear in Chapter 5.

[4]For a definition and discussion on analytic signals, and the associated *Hilbert Transforms*, see [56]

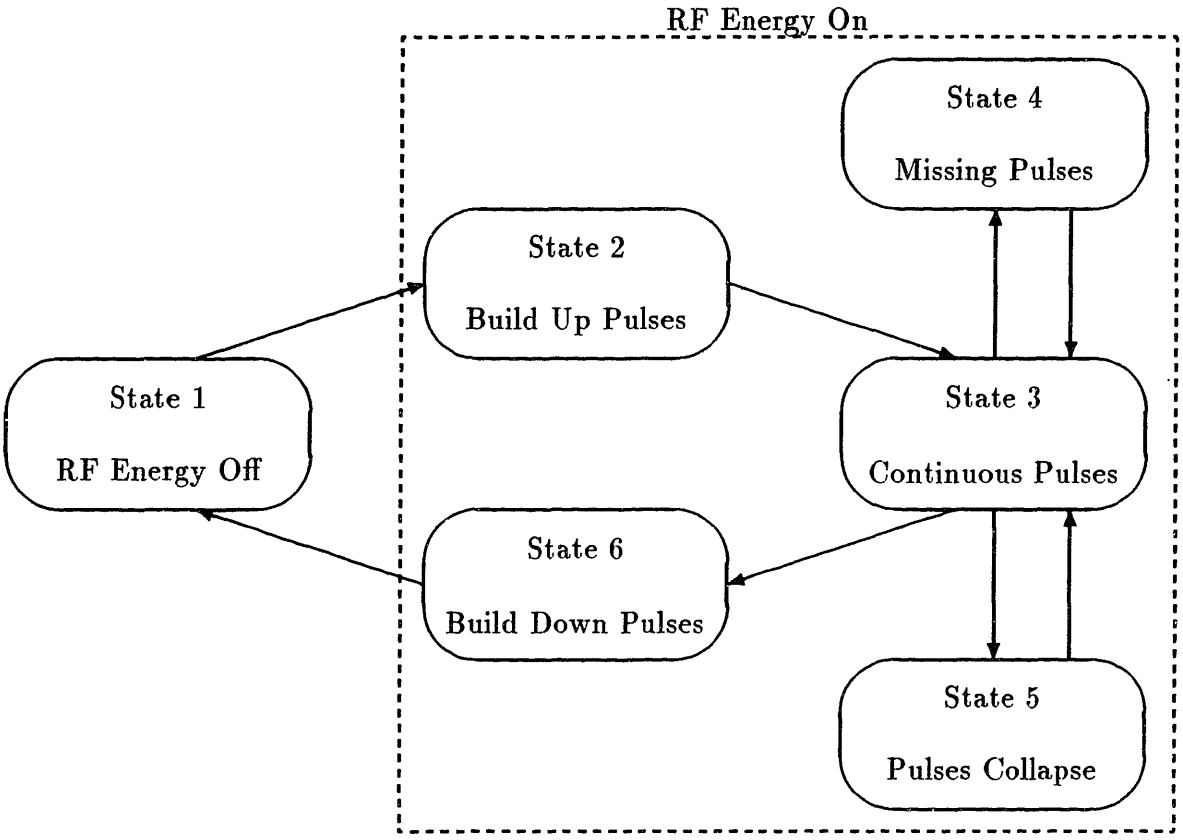

Figure 2-1: Proposed State Model for Microwave Oven Interference

2.3.4 Type B2 Interfering Signal

Although direct sequence spread spectrum signals from amateur radio stations or other sources could be considered as examples of Type B2 interfering signals, we have chosen radio frequency interference from microwave ovens as the representative because of some unique properties of the interference, and because of the ubiquity of the use of microwave ovens that coincides with the indoor communications operations domain.

Following are some observations we made using the data from [13], [14], [15] for microwave oven interference characteristics:

- The radiation has a random and contiguous frequency content over approximately 100 MHz bandwidth, with a carrier frequency of 2.45 GHz and specific amplitude envelope distribution.

- The interference radiation emission is synchronous to the 60 Hz power supply frequency, and the interference energy exists for only half of the period.

- During the one half period of the power supply when interference energy exists, the amplitude of the emission takes a complicated distribution. During the first $n\Delta t$ seconds, n pulses are emitted, spaced Δt apart, with the peak amplitudes uniformly rising from a relative value of 0 to 1, where n is between 1 and 10, and Δt was observed to be approximately $3\mu s$. Then the envelope takes the form of continuous pulse train, approximately equal in amplitudes, also spaced Δt apart. When this continuous pulses train occurs, sometime pulses may be missing, or several pulses may collapse together to form contiguous radiation. At then end of the one half period, again n pulses are emitted spaced Δt apart, with the amplitudes decreasing.

- Rise times of the pulses are about 5-10 ns, and that missing pulses occurred roughly about 5-10% of the time.

- Amplitude distribution of for the pulses is always higher than the adjacent continuous carrier by about 10 dB.

Based on the above information, we developed a six-state model[5], shown in Figure 2.1, for the microwave oven interference. State 1 corresponds to the one half period of the power supply when there is no interference energy present. States 2 and 6 correspond to the build-up and build-down of the pulse train, during the start and

[5]In [14], an ideal interference model in the form of a mathematical expression, which does not account for the missing pulse phenomenon, and the pulse collapsing phenomenon, and the non-zero pulse durations, was proposed. The six state model proposed here could be considered as an extension of this model, although we had to make several assumptions concerning the transition probabilities between States 3, 4, and 5, duration of stay in State 5, and the distribution for pulse durations. Future work in extending this model should include the verification of the assumptions we had made.

the end of the radiation for every cycle. There will be n pulses in each state, and we will assume that the integer n is uniformly distributed in $[1, 10]$. The duration the process stays in these two states are $3n$ μs per cycle, corresponding to the value of $\Delta t = 3\mu$s. State 3 corresponds to the continuous emission of pulses spaced Δt apart, during which the process may move to States 4 or 5 randomly, and return to State 3 randomly. We assign the value of .075 (the mean of 5% and 10%), for the probability of the process going to State 4 from State 3 at any given time, and a value of 1 for the probability that it will return to State 3 from State 4. Probabilities of transition to and from State 5 are not known, and we will make appropriate assumptions when we perform the Monte Carlo simulations in Chapter 5.

Now, we will return the the issue of constructing a model in our hypothetical band which we will refer to as MWO, based on the microwave oven interference. Since 2.45 GHz is 8.25 MHz away from the center frequency of the 2400-2483.5 MHz band, and it appears to be a constant, we find that $\Delta f_{MWO}(t) = 8.25$ MHZ for all t. Further, $\Delta\theta_{MWO}(t)$ will also be a constant, randomly distributed in $[0, 2\pi]$. And we realize that $s_{MWO}(t)$ will now be a square wave at 60 Hz, oscillating between levels 0 and 1, and $r_{MWO}(t)$ will take a complicated distribution, dictated by States 2 through 6 of Figure 2.1. Further details will appear in Chapter 5.

2.3.5 Type C1 Interfering Signals

Reader may find a strong similarity between our Type C1 signals and Middleton's Class B noise that was discussed in Section 2.1.2 of this chapter. Type C1 interfering signals are impulsive noise signals, that have an extremely wide bandwidth, occupying several of the permitted bands, with emissions probably suppressed in the forbidden bands that occur in between the permitted bands.

The example of Type C1 interfering signals that we will explore is radio frequency emissions from photocopiers. Measurements of impulsive noise due to photocopiers have been studied, and some *limited* information is available. The following are some observations we made from [6], [7]:

- Impulsive noise inside buildings is very wide band, and pulse durations observed

are directly a function of the receiver bandwidth.

- The mean amplitude of radiation, above mean thermal noise power, was 11.5 dB, 12.7 dB, and 10.6 dB, in the three bands centered at 918 MHz, 2.44 GHz, and 4.0 GHz, respectively, for photocopier emissions.

- The mean and standard deviation of the pulse duration observed in the 2.44 GHz band, using a 30 MHz bandwidth, were 143 ns and 155 ns, respectively, for photocopier emissions.

- Spacing between consecutive bursts were similar in the three bands measured, and specifically the mean and the standard deviation of pulse spacing in the 2.44 GHz band, using a 30 MHz bandwidth, were 221 ns and 220 ns, respectively, for photocopier emissions.

We would like to construct a model in our hypothetical band centered at f_o, which we will refer to as PC, based on interference characteristics of the photocopier in the 2.44 GHz band. Once again we find $s_{PC}(t) = 1$, since we cannot expect to see any deterministic absence of the interference. Unfortunately, characterization of $r_{PC}(t), \Delta f_{PC}(t)$ and $\Delta\theta_{PC}(t)$ does not follow directly from the observations above (although we have some information about the amplitude distributions and pulse interarrival times, we have no information about the phase).

Although the exact process within the operation of a photocopier that is responsible for the emissions is not clear, it appears that *corona discharges* are often present in electrophotographic reproduction [35]. Further, "lightning pulses" that are very narrow in time have been found due to corona discharges [29], and the general statistical properties of corona pulses have been studied [1] [40], [49]. So, assuming that the photocopier emissions are due to corona discharges, or a process very similar to corona discharges, we state the following postulate[6] on which the model we are about to develop will be based on.

[6]The model proposed here, although implicitly uses the data observed in [6], [7], was developed primarily based on the postulate we proposed. Therefore, the authors of [6], [7] should not be held responsible for any errors that may have resulted in the modelling, and future researchers should verify all aspects of the model prior to extending the model.

- The pulses occur randomly, according to a Poisson distribution, and the pulse durations are sufficiently narrow to allow the interference energy to extend to frequencies much larger than 4 GHz. Further, there is no modulation present, and hence in its natural form, the interference does not exhibit any oscillatory behavior.

- The pulse shape, for a pulse occurring at time T_k could be characterized by $A_k e^{-\lambda(t-T_k)}u(t-T_k)$, where A_k has a Gaussian distribution, and λ is a random variable with mean value much larger than 4 GHz.

Specific assumptions for the distribution of λ, as observed *after* a bandpass filtering process in the receiver, will be made in Chapter 5, for the purpose of simulating the interference. Now, based on the postulate above, we can develop the appropriate distribution for $r_{PC}(t)$. Since the pulses arrive according to a Poisson distribution, the interarrival time between pulses will be exponentially distributed. For an exponential distribution, the expected value should be equal to the standard deviation [23] and the values of 221 ns and 220 ns for the mean and standard deviation of the pulse spacing in the 2.44 GHz band, appear to support our postulate (the receive filter appeared only to affect the pulse duration, and not the pulse spacing). Hence, we conclude that $r_{PC}(t)$ should consist of randomly spaced pulses, with exponential interarrival times, and with the peak amplitude Gaussian distributed. Further assumption about the pulse duration will be made in Chapter 5.

We suspect that $\Delta f_{PC}(t)$ should be zero since there is no modulation. However, it seems unclear how the $\Delta\theta_{PC}(t)$ should look like. In order to help decide the distribution for the phase offset, we state and prove the following lemma, which will consequently be used to prove Proposition 2.2.

Lemma 2.1 *For a pulse $c(t) = e^{-\lambda(t)}u(t)$, the Fourier Transform[7] of $c(t)$, is approximately constant over $[f_o - W, f_o + W]$ and $[-f_o - W, -f_o + W]$, provided that $f_o \ll \lambda$ and that $W \ll f_o$.*

[7]We define the Fourier Transform of $x(t)$ as $X(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft}dt$ and the Inverse Fourier Transform of $X(f)$ as $x(t) = \int_{-\infty}^{\infty} X(f)e^{i2\pi ft}df$.

Proof: The Fourier Transform of $c(t)$, at f_o is given by[67]

$$C(f_o) = \frac{1}{\lambda + i2\pi f_o} = \frac{\lambda}{\lambda^2 - (2\pi f_o)^2} - i\frac{2\pi f_o}{\lambda^2 - (2\pi f_o)^2} \tag{2.8}$$

Hence, when $\lambda \gg f_o$, we could approximate $C(f_o)$ as

$$C(f_o) \approx \frac{1}{\lambda} \tag{2.9}$$

With $W \ll f_o$, the result follows. □

The assumption $f_o \ll \lambda$ is not unreasonable since the interference energy appears to extend to frequencies much larger than 4 GHz, and the center frequencies of interest are only 915 MHz and 2.44 GHz. We use the above lemma to prove the following proposition.

Proposition 2.2 *For a pulse occurring at T_k, given by $c_1(t) = c(t - T_k)$, where $c(t)$ is as defined in Lemma 2.1, the output $c_2(t)$ of the bandpass filtered version of $c_1(t)$, using an ideal bandpass filter with center frequency f_o and bandwidth $2W$, is given by*

$$c_2(t) \approx 2C(f_o)\hat{c}(t)\cos[2\pi f_o(t - T_k)] \tag{2.10}$$

where $\hat{c}(t)$ is the lowpass filtered version of $\delta(t - T_k)$ with an ideal lowpass filter having a cutoff frequency of W, provided that the conditions of Lemma 2.1 hold.

Proof: First we note that $C_1(f) = C(f)e^{-i2\pi f T_k}$. By using Lemma 2.1, we write

$$c_2(t) = C(f_o)\left(\int_{-f_o-W}^{-f_o+W} e^{i2\pi f(t-T_k)}df + \int_{f_o-W}^{f_o+W} e^{i2\pi f(t-T_k)}df\right) \tag{2.11}$$

Upon integration and substitution of limits,

$$c_2(t) = 2C(f_o)\frac{\sin[2\pi W(t - T_k)]}{\pi(t - T_k)}\cos[2\pi f_o(t - T_k)] \tag{2.12}$$

We recognize that $\frac{\sin[2\pi(t-T_k)]}{\pi(t-T_k)}$ is the response of an ideal lowpass filter with cutoff W to a delayed ideal impulse $\delta(t - T_k)$. The result follows. □

With the aid of Proposition 2.1 we conclude that $\Delta f_{PC}(t) = 0$. Now we could write the equivalent baseband representation $R_{PC}(t)$ as

$$R_{PC}(t) = \sum_{k=1}^{q} A_k \hat{c}(t - T_k) e^{-i(2\pi f_o T_k + \theta_o)} \tag{2.13}$$

where θ_o is the constant phase offset due to the downconversion process which we assume is uniformly distributed in $[0, 2\pi]$, and the pulses occur at times T_k, for $k = 1, ...q$. The interarrival times $T_k - T_{k-1}$ are exponentially distributed and the peak amplitude A_k is Gaussian distributed. Equation (2.13) characterizes both r_{PC} and $\Delta\theta_{PC}$. We realize that $\hat{c}(t)$ appears to be the response of an ideal lowpass filter to an ideal impulse, because of the assumptions made through Lemma 2.1. Hence, in Chapter 5, we will reconsider the issue of appropriate duration and shape for the filtered pulse. We could further write equation (2.13) in a manner consistent with Proposition 2.1, with explicit expressions for $r_{PC}(t)$ and $\Delta\theta_{PC}(t)$, but we realize that such a step is unnecessary since all the information we need is contained in equation (2.13).

2.3.6 Type C2 Interfering Signal

We will model the example of Type C2 interfering signal, called GN, based on the theoretical *bandlimited white Gaussian noise* (the noise is bandlimited because of the receive filter).

We write the bandpass noise as

$$y_{GN}(t) = x_1(t)\cos(2\pi f_o) + x_2(t)\sin(2\pi f_o t) \tag{2.14}$$

Here $x_1(t)$ and $x_2(t)$ are *independent* and each of them are white Gaussian noise processes, bandlimited to $[-W, W]$. We note that $x_1(t)$ and $x_1(t + k\tau)$, where k is a non-zero integer and $\tau = (1/2W)$, are independent, and similarly for $x_2(t)$ [64].

Noting that $s_{GN}(t) = 1$ and $\Delta f_{GN}(t) = 0$, we could write the equivalent baseband

Model	Type	Model Based On
EASD	A1	Electronic Article Surveillance Devices
FM	A2	Narrowband Amateur Radio Frequency Modulated Signals
SSB/FH	B1	Frequency Hopped Single Sideband Amateur Radio Signals
MWO	B2	Microwave Oven Emissions
PC	C1	Photocopier Emissions
GN	C2	Theoretical bandlimited white Gaussian noise

Table 2-1: Summary of the Six Models of Interfering Signals

representation $R_{GN}(t)$ as

$$R_{GN}(t) = r_{GN}(t)e^{i\theta_{GN}(t)} \tag{2.15}$$

where the envelope $r_{GN}(t)$ has a Rayleigh distribution and the phase $\theta_{GN}(t)$ has a uniform distribution, since it is well know that the envelope of a Gaussian process is Rayleigh distributed, and the phase is uniformly distributed [20], [54]. More details will appear in Chapter 5.

2.4 Chapter Summary

In this chapter, we presented a definition for interfering signals, and introduced the reader to the six classes of interfering signals that we developed. Then we proposed a representation for the received form of an interfering signal, which identifies the slow-varying and rapid-varying envelopes, the frequency offset and the phase offset as critical parameters in describing an interfering signal. We also provided a formal definition for the slow-varying envelope. We then selected a representative example from each of the six categories of interfering signals and constructed models for them in our hypothetical band. A summary of the six models appear in Table 2.1. These models will be further developed in Chapter 5, to be used in the performance evaluation stage. These models were drawn from different categories, thus spanning a wide variety of interfering signals, and we will use them to motivate the appropriate feature selection methodology, which will be the topic of Chapter 3.

Chapter 3

Strategy for Feature Extraction

This chapter concerns itself with the design of the feature extraction stage whose purpose is to extract representative features from the observed data for the purpose of adaptive learning and classification. We first discuss the constraints of the feature extraction stage, and we set our objectives to develop a strategy for feature extraction. In the process of developing the strategy, we will identify several features that will be useful in classifying the six examples of interfering signals that we discussed in Chapter 2, and we also identify several other features that would be useful when more interfering signals are involved.

3.1 Constraints and Objectives

In Chapter 1 we divided the problem of system design for adaptive signal classification into two independent problems of signal acquisition and signal analysis, and we solved the problem of signal acquisition, illustrating the benefits of using the same signal acquisition architecture for both the Learning and Diagnostic modes. Similarly, we would like to design the feature extraction stage such that the same features are extracted in both of the modes, with repeated measurements required for the Learning mode. The topic of adaptive learning and signal classification will be discussed in Chapter 4, and in this chapter we will focus on identifying suitable features to extract from the acquired signal.

The primary purpose of the feature extraction stage to is reduce the the observed data to a smaller feature vector, consisting of *sufficient features.* Specifically, this corresponds to extracting as few features as possible, but at the same time ensuring that the features extracted contain sufficient information for distinguishing among the different interfering signals.

The Learning mode should require minimal supervision from the user, with the user having to only ensure that the interfering signal is present for the duration of the measurements. The architecture we proposed in Chapter 1 is capable of making the required number of measurements during the Learning mode. If the feature extraction stage is designed carefully, there will be no necessity for modifying the feature extraction stage when new interfering sources are found, thus making the adaptive learning requirement satisfied with a simple design.

If we could identify a set of optimal features such that any interfering signal could be recognized using this set, our problem will be greatly simplified. However, we realize that to search for a feature vector that has such *universal* applicability would be an impossible task, especially since an exhaustive search for all characteristics of interfering signals has not been done. So we set our target as to first find a suitable approach, or a suitable *strategy* for feature extraction and identify a viable set of features to recognize the limited number of interfering signals that were discussed in Chapter 2, and in the process of doing so, we hope to discover several other features that would be useful when a larger set of interfering signals is encountered.

As stated in the introduction to this report there appears to be no prior work related to the complex problem of interfering signal classification, and thus there seems to be no suitable background on which we could base our search. In pattern classification problems, feature extraction stage is considered to be much more problem dependent than the learning and classification stages [21]. Hence, an approach that has been used for related classification problems, which can at best be regarded as a subset of the complex problem of interfering signal classification, is not expected to be directly applicable[1]. So, we will use the understanding of the fundamental char-

[1]For example, one subset of interfering signals is modulated signals, and for the purpose of

acteristics of the interfering signals that were developed in Chapter 2 to provide the starting point for the feature extraction problem.

3.2 Constructing the Feature Vector

The four parameters consisting of the slow-varying and rapid-varying envelopes, the phase and frequency offsets, were identified in Chapter 2 as being critical in describing an interfering signal. Now we would like to further explore methods to recover those parameters implicitly present in the received signal. We realize therefore, there is a need for an intermediate stage, which we call *waveform characterization*, where we reconstruct the representative waveforms in the form of the slow-varying and rapid-varying envelopes, phase and instantaneous frequency functions, and spectral frequency estimates. From these intermediate waveforms, we will then extract a set of sufficient features to construct the feature vector.

There is one issue that we need to resolve prior to launching into the feature extraction process. The actual amplitude of the signals, and hence the *power* received from signals is not a useful measure in classifying the signal because the received power is often a function of several variables including the power supplied to the source of interfering signal, and the distance of the source form our receiver. Hence, we realize that the envelope computations should incorporate a normalization step, such that the actual power of the received signal does not bias our decision. So, we normalize an envelope $e(t)$ (which may either be the slow-varying or rapid-varying envelope), such that the resulting normalized envelope $e_n(t)$ has the property:

$$E[e_n^2(t)] = 1 \tag{3.1}$$

modulation recognition features like modulation index were used[52]. Another subset would be impulsive noise sources, where features like pulse duration and pulse spacing have been suggested for radio noise surveys [68], and Gaussian factor and overlap index have been used for recursive identification of impulsive noise channels[73]. Clearly these features can only be used if we are willing to tolerate a more complex learning and classification process, where we use different features for different types of interfering signals.

By ensuring that the second moment of the envelope has a nominal value of 1, we will be comparing normalized received signals that will appear to have the same total energy or power[2], regardless of the origin of the signal, and hence we could consider the comparison to be "fair". The problem of normalization will not be present in feature extraction from the characterizations of phase, instantaneous frequency and spectral frequency estimates.

3.2.1 The Slow-Varying Envelope and Feature v_1

In Chapter 1, we illustrated one method of obtaining $s(t)$, the slow-varying envelope, in hardware. If this method is used, then there will be no additional processing involved in computing $s(t)$, with the exception of normalizing the envelope using the criterion of equation (3.1). The problem remaining is the determination of a suitable feature to be extracted from $s(t)$.

We recall from Chapter 2 that there were only two kinds of $s(t)$. The first kind corresponds to most of the interfering signals, where there was no known duration of deterministic absence of the interference and $s(t) = 1$ for all t. The second kind was observed in the case of microwave oven emissions, where $s(t)$ was a square wave at 60 Hz. Further, we note that the duration of observation was fixed to be equal to one period of the 60 Hz square wave.

In order to distinguish between a constant-valued envelope and a square wave, we could use the *variance* of the envelope. The variance of an ideal constant-valued envelope will be 0. By writing the variance $\sigma^2_{s(t)} = E[s^2(t)] - E^2[s(t)]$, and by noting that $E[s^2(t)] = 1$ due to our normalization, it can be easily shown that for an ideal square wave envelope, where the duration of observation is exactly one period of the square wave, the variance would be $\frac{1}{2}$. Since the variance is continuous valued, it can also be used to classify envelopes other than the two kinds we saw above.

Hence, we decide to include the variance of the slow-varying envelope in the feature vector. This feature will be referred to as v_1.

[2]For a definition of energy and power of a signal, see[45]

3.2.2 The Rapid-Varying Envelope and Feature v_2

There are two steps involved in obtaining $r(t)$, the rapid-varying envelope. First, since the acquired complex baseband signal $R(t)$ will consist of the in-phase and quadrature components $x_I(t)$ and $x_Q(t)$, $r(t)$ is given by $r(t) = \sqrt{x_I^2 + x_Q^2}$, and this absolute value has to be computed. Then the computed envelope should be normalized using the criterion of equation (3.1).

Reasoning along the same lines as with the previous case of $s(t)$, we see that the variance of $r(t)$ contains useful information. Due to the normalization, the variance will take values between 0 and 1. The variance will be close to 0 for signals with a constant envelope (like FM signals and signals from EASD), and the variance will be close to 1 for highly impulsive signals (like emissions from photocopiers).

We do not consider extracting the mean of the envelope as an additional feature because the variance $\sigma^2_{r(t)}$ can be written as $\sigma^2_{r(t)} = 1 - E^2[r(t)]$, after the normalization, and therefore computing the mean would be redundant. However, higher order central moments in the form of skewness or kurtosis[3] of the envelope would be useful in classifying impulsive signals that exhibit varying levels of impulsiveness, and also in classifying non-impulsive signals that exhibit a variety of distributions. For information on the different values of skewness and kurtosis for a variety of impulsive and non-impulsive signals, see [33].

But in light of our decision to extract only the minimum required features for classifying the six signals in consideration, we restrict ourselves to including only the variance of the rapid envelope in the feature vector. This feature will be referred to as v_2.

3.2.3 Instantaneous Frequency and Features v_3 and v_4

With respect to the general interfering signal model of Proposition 2.1, we have reconstructed $s(t)$ and $r(t)$ and decided on suitable features to extract. Now the remaining problem concerns the phase and carrier frequency offsets, $\Delta\theta(t)$ and $\Delta f(t)$.

[3]Definition and discussion on skewness and kurtosis could be found in most textbooks on statistics, including [23], [43].

Unfortunately, the phase $\phi(t)$ of the complex baseband signal $R(t)$ consists of *both* of the offsets (recall the expression $\phi(t) = 2\pi\Delta f(t)t + \Delta\theta(t)$ from the proof of Proposition 2.1).

Computing $\phi(t)$ from the observed $R(t)$ is straightforward, since $\phi(t)$ is simply the argument (phase) of $R(t)$. However, it should be noted that this method only generates the *principal* value of $\phi(t)$, in the range $[-\pi, \pi]$ (or equivalently, in the range $[0, 2\pi]$), although the actual phase may very well take a value outside this range.

We realize that it is not possible to exactly recover $\Delta f(t)$ from $\phi(t)$ in general, even in the absence of any noise. However, we would like to obtain as much information as possible about $\Delta f(t)$ from $\phi(t)$. One answer to this problem is in the form of *instantaneous frequency.*

Instantaneous frequency has become popular for analysis of time-varying signals, and has found applications in a variety of fields including detection of harmonically related signals [8], time-varying filtering [9], analysis of cyclostationary signals [75], speech pattern analysis [5], and modulation recognition [38], [46], [71]. Since our interfering signals in general will exhibit a time-varying behavior, we expect the instantaneous frequency to be useful in the classification process.

We define the instantaneous frequency of a complex valued signal as the derivative of the phase[4], and this definition is similar to the classical definition by Mandel [47]. Hence, for our complex signal $R(t)$, whose phase is given by $\phi(t)$, the instantaneous frequency will be $\frac{d\phi(t)}{dt}$. Noting that $\phi(t) = 2\pi\Delta f(t)t + \Delta\theta(t)$, and writing $\Delta\theta(t) = \theta(t) + \Delta\theta_o$, we see that the instantaneous frequency is given by

$$\frac{d\phi(t)}{dt} = 2\pi\frac{d\Delta f(t)}{dt}t + 2\pi\Delta f(t) + \frac{d\theta(t)}{dt} \tag{3.2}$$

Since equation (3.2) defines the instantaneous frequency for a continuous-time

[4]Definition of the instantaneous frequency is necessary because, as noted by Cohen [16], the term *instantaneous frequency* is subject to definition. For example, in [27], in addition to a definition that is similar to (3.2) a variety of "instantaneous" frequencies have been illustrated, including Mathematical Frequency, Zero Crossing Frequency, and the Running Fourier Frequency, and they were all instantaneous in the sense that they were all a function of the present time (which the Fourier frequency, in the ordinary sense, is not).

signal, we need develop a method for computing its value for a discrete-time signal. However, to avoid the theoretically required digital differentiators, we will develop an approximate method based on the difference in phase between adjacent samples. We write the (radian) instantaneous frequency $w_i(t) = \frac{d\phi(t)}{dt}$ as

$$w_i(t) = \lim_{\delta t \to 0} \frac{[(\phi(t+\delta t) - \phi(t-\delta t)] \bmod .2\pi}{2\delta t} \tag{3.3}$$

where the notation mod.2π represents a modulo 2π operation to account for $\phi(t)$ being defined on $[-\pi, \pi]$. Setting $\Delta t_s = 2\ \delta t$ for a discrete-time signal, where Δt_s is the sampling interval, and by using the discrete-time index n, instead of t, and by delaying $w_i[n]$ by half a sample, we see that

$$w_i[n] = \frac{(\phi[n] - \phi[n-1]) \bmod .2\pi}{\Delta t_s} \tag{3.4}$$

for a discrete-time signal. We would like to express w_i as a fraction of the sampling frequency. Noting that $w_i[n]$ takes values on $[-\frac{\pi}{\Delta t_s}, \frac{\pi}{\Delta t_s})$, we define the normalized instantaneous frequency $f_i[n]$ as

$$f_i[n] = \frac{\phi[n] - \phi[n-1] \bmod .2\pi}{2\pi} \tag{3.5}$$

which will be used in Chapter 5.

Let us consider the special case where the signal has a constant carrier frequency and has a constant phase offset (like signals from EASD and microwave ovens). In this case we see that the instantaneous frequency will be exactly the same as $\Delta f(t)$, which is a constant. The mean of the instantaneous frequency, in the presence of zero mean noise, will yield an unbiased estimate of the constant carrier frequency offset. The variance will be zero in the absence of noise, or equal to the noise variance in the presence of any noise, thus indicating that the signal has a constant carrier frequency and phase.

In the case of FM signals, where the carrier frequency is constant, but the phase

is not, the instantaneous frequency will be

$$\frac{d\phi_{FM}(t)}{dt} = 2\pi\Delta f_{FM}(t) + \frac{d\theta_{FM}(t)}{dt} \tag{3.6}$$

We recall from equation (2.5) of Chapter 2 that the term $\frac{d\theta_{FM}(t)}{dt}$ is actually a scaled form of the the message transmitted. If the message has a zero mean over the duration of observation, then the mean value of the instantaneous frequency will yield the carrier frequency offset. The variance of the instantaneous frequency will be the scaled variance of the message transmitted (the scaling occurs because of the factor k_f in equation (2.5)).

In the case where the carrier frequency offset is zero, like for the case of Gaussian noise or the emissions from photocopiers, the instantaneous frequency will simply be the derivative of $\theta(t)$. Since the phase of $R_{PC}(t)$, as we showed in Proposition 2.2, takes only discrete values, the derivative of $\theta_{PC}(t)$ will be zero most of the time, with impulses occurring at times when there is a new pulse. Hence the mean and variance of the instantaneous frequency will be small. In the case of $R_{GN}(t)$, the phase is uniformly distributed, and we expect the mean of the derivative of $\theta_{GN}(t)$ to be close to zero, but the variance will be large.

When both the phase and the carrier frequency are a function of time, like in the case of SSB/FH (where the carrier frequency is hopping) we see that the instantaneous frequency is

$$\frac{d\phi_{SSB/FH}(t)}{dt} = 2\pi\frac{\Delta f_{SSB/FH}(t)}{dt} + 2\pi\Delta f_{SSB/FH}(t) + \frac{d\theta_{SSB/FH}(t)}{dt} \tag{3.7}$$

From equation (2.7), we know that $\theta_{SSB/FH}(t)$ is a function of $g(t)$, the transmitted message which is bandlimited to audio frequencies. Since the bandwidth of $g(t)$ is very small compared to the sampling rate, $g(t)$ will vary very slowly in time, and hence $\theta_{SSB/FH}(t)$ will vary very slowly in time. So the contribution of $\frac{d\theta_{SSB/FH}(t)}{dt}$ to the right hand side of equation (3.7) will be negligible, and we rewrite equation (3.7) as

$$\frac{d\phi_{SSB/FH}(t)}{dt} \approx 2\pi\frac{\Delta f_{SSB/FH}(t)}{dt} + 2\pi\Delta f_{SSB/FH}(t) \tag{3.8}$$

If a hop occurs during the duration of observation, then the instantaneous frequency will take discrete values, with a "jump" to the new frequency at the point when the hop occurs, and in this case the mean of the instantaneous frequency will be a weighted average of the two (or more) carrier frequency offsets, and the variance will provide a measure of the distance between the hopping frequencies. If a hop does not occur during the duration of observation, then the mean would yield an estimate of the carrier frequency, and the variance will be close to zero.

Higher order moments in the form of skewness and kurtosis of the instantaneous frequency will be useful when there are several frequency and phase modulated signals, occupying a wide bandwidth, that have to be distinguished. For our purpose, the mean[5] and the variance of the instantaneous frequency should be sufficient. We will refer to these two features as v_3 and v_4, respectively.

3.2.4 Estimated Phase and Feature v_5

In the previous section, we attempted to estimate $\Delta f(t)$ from $\phi(t)$, but ended up also estimating $\frac{d\Delta\theta(t)}{dt}$. In order to distinguish a signal which has a constant $\Delta\theta(t)$ from a signal that has a $\Delta\theta(t)$ that varies very slowly in time, $\frac{d\Delta\theta(t)}{dt}$ will not be useful and we still need to estimate $\Delta\theta(t)$ from $\phi(t)$.

In the case where the carrier frequency is constant, we could estimate $\Delta\theta(t)$ by first computing the mean instantaneous frequency, which provides an estimate of the carrier frequency offset, obtaining $\Delta\hat{f}_c$, and subtracting the product $\Delta\hat{f}_c t$ from $\phi(t)$. Since we will not know the initial value of t that should be used, this procedure will result in an additional constant phase error in the estimate of $\Delta\theta(t)$, but this should not pose any additional problem since there is already a constant phase error due to the mismatch of the signal phase to the phase of the local oscillator. In the

[5]The reader may wonder why we include the mean of the instantaneous frequency, since only the microwave oven has a unique carrier frequency among all of our models. Although we have not considered restricting the carrier frequencies of the other sources, in practice, many sources do have a restricted frequency range. For example the electronic article surveillance devices manufactured by Sensormatic Inc., use only the 902-905 MHz range of the 902-928 MHz band[59]. As we will see in Chapter 4, our algorithms for learning and classification automatically reduce any bias that may result when the mean frequency is not unique for a given source.

case where the carrier frequency is not a constant, like the SSB/FH signal, the mean instantaneous frequency will not yield an unbiased estimate of the carrier frequency, but in this case the phase will not be particularly useful in classification, and as such we will neglect the error in the estimate of the phase.

Now that we have an estimate of the phase, the next step is to decide on a suitable feature to extract from the estimated function. The important information we wish to extract from the estimated phase is not the mean value of the phase, since this is simply an uninteresting random variable, but the *phase spread* of the signal. The phase spread will be a critical statistic in distinguishing narrowband signals that have a constant envelope. For example, in distinguishing between single tone signals from EASD and narrowband FM signals, we will find (in Chapter 6) the phase spread to be the critical differentiating feature.

Although a measure of the phase spread could be obtained by calculating a suitable value of variance, this step is not straightforward because the values of the estimated phase are the principal values in the range $[-\pi, \pi]$ as observed previously, and therefore we cannot use the linear variance measure. The answer to this problem is in the form of *circular variance*, defined by Mardia [48]. If a set of observations consist of phase values $\theta_i, i = 1, \ldots, L$, then the sample circular variance Q_o for the observation is given by

$$Q_o = 1 - \frac{1}{L}\left[\left(\sum_{i=1}^{L}\cos\theta_i\right)^2 + \left(\sum_{i=1}^{L}\sin\theta_i\right)^2\right]^{\frac{1}{2}} \tag{3.9}$$

The circular variance results in a value between 0 and 1, with 0 indicating that the phase is a constant, and a value close to 1 indicating that the phase values are widely dispersed. We expect the phase spread to be close to 0 for EASD signals, and very large for FM signals.

Since the mean value of the phase is not useful, we will not attempt to include the circular mean of the phase in our feature vector. However, the circular skewness and kurtosis will be useful in classifying signals that exhibit a variety of different distributions in their phase values, particularly when the signals occupy a narrow

bandwidth. This is a pleasant contrast to the statistics of the instantaneous frequency which will be useful when the phase and frequency modulated signals (both analog and digital) have a wide bandwidth.

However, for our limited set of interfering signals, the phase spread factor in the form of circular variance is sufficient. This feature will be referred to as v_5.

3.2.5 Time Variant Periodogram and Feature v_6

We have extracted a number of statistics for the feature vector, but none of them contain any information about the *spectral bandwidth* of the signals. Recognizing that our basis for categorizing the signals was the spectral bandwidth, we would like to obtain some measure of the bandwidth of the interfering signals. The reader should note that there is no obvious relationship between instantaneous frequencies and Fourier components, as observed by Mandel [47]. In particular, Fourier components are defined only over the infinite time domain, whereas instantaneous frequencies are defined at an instant of time.

Time-frequency analysis is a rich field, and a variety of spectral analysis techniques have been proposed for the study of time-varying signals, including the short-time Fourier transforms, Wigner-Ville distributions, discrete Zak transforms, and Gabor representation [3], [16], [17], [24]. However, incorporating such detailed spectral analysis techniques is not within the scope of our project, and the curious reader is referred to [16], which is a classical tutorial on time-frequency distributions.

For the purpose of extracting a suitable measure of the signal bandwidth during the observation period, we will use the *time-variant periodogram.* We define the time-variant periodogram $S_T(t,f)$ of our complex signal $R(t)$ as [26], [62]

$$S_T(t,f) = \frac{1}{T}|\mathcal{R}_T(t,f)|^2 \tag{3.10}$$

where $\mathcal{R}_T(t,f)$ is given by

$$\mathcal{R}_T(t,f) = \int_{t-T/2}^{t+T/2} R(u)e^{-i2\pi fu}du \tag{3.11}$$

$S_T(t,f)$ is the normalized squared magnitude of the Fourier transform of the signal segment of length T centered at time t. The time $(t - T/2)$ corresponds to the time when the trigger signal of the signal acquisition hardware is set, which in the case of an ideal system should always correspond to the same event, and for a given system, the duration of observation T will be fixed. Hence, we will drop the T and t in subsequent references to $S_T(t,f)$, and simply use the notation $S(f)$. The magnitude of Fourier transform could easily be computed using Fast Fourier Transform (FFT) algorithms.

We are interested in the local[6] spectral bandwidth of the signal as depicted in the acquired periodogram. We define the *local spectral bandwidth* $\sigma^2_{S(f)}$ as

$$\sigma^2_{S(f)} = \frac{\int_{-W}^{W}(f - <f>)^2 S(f)df}{\int_{-W}^{W} S(f)df} \tag{3.12}$$

where the mean frequency $<f>$ is given by

$$<f> = \frac{\int_{-W}^{W} fS(f)df}{\int_{-W}^{W} S(f)df} \tag{3.13}$$

and the complex baseband signal $R(t)$ is contained in the band $[-W, W]$.

Although higher order central moments will provide useful information in characterizing a variety of spectral distributions, we will restrict ourselves to $\sigma^2_{S(f)}$, the variance of the spectral frequency. This feature will be referred as v_6.

3.3 Chapter Summary

In this chapter, we discussed the constraints of the feature extraction problem, and set our objectives to search for a feature extraction strategy. We used the interfering signal models of Chapter 2 as our basis, and we discovered that there is an intermediate stage of waveform characterization prior to feature extraction. Features discovered

[6]The spectral bandwidth is *local* because the acquired periodogram corresponds to a finite window in time, representing the spectral content only for the event of interest.

Feature	Description
v_1	Variance of the slow varying envelope
v_2	Variance of the fast varying envelope
v_3	Mean of the instantaneous frequency
v_4	Variance of the instantaneous frequency
v_5	Phase spread (circular variance of the phase)
v_6	Local spectral bandwidth (variance of spectral frequency)

Table 3-1: Summary of the Six Components of the Feature Vector **v**

as being viable to perform the classification were in the form of mean and variance of the characterized waveforms, with higher order moments in the form of skewness and kurtosis recommended for advanced applications. There is an added advantage in extracting features in the form of statistical moments, since this will eventually lead us to assume a multivariate Gaussian distribution for the feature vector, thereby greatly simplifying the adaptive learning and classification stages, as we will see in Chapter 4.

Chapter 4

Adaptive Learning and Signal Classification

This is the final chapter of Part I of this report, where we conclude our development of architecture and algorithms for adaptive classification of interfering signals, by developing the appropriate algorithmic design for the Learning and Diagnostic modes. We first discuss the constraints and objectives for the learning and classification stages, and initiate the search for a suitable decision rule. We decide on the Maximum Likelihood rule, a special case of the more general Bayes minimum error rule, and explore the details of the Learning and Diagnostic (classification) modes. We conclude the chapter by reviewing the assumptions that were made in arriving at the Maximum Likelihood rule, and we describe a method to estimate an upper bound on the Bayes error, which we could later use to assess the performance of our decision rule.

4.1 The Search for a Decision Rule

4.1.1 Constraints and Objectives

The learning and classification stages are very closely related, since the learning stage is the preparation for the classification stage. Thus, we should first search for a suitable *decision rule* for assigning classes in the Diagnostic mode (the classification

stage), keeping in mind that the decision rule should require as simple a learning process as possible, and minimal supervision from the user. Specifically, the user should be required only to ensure that the interfering signal being introduced to the diagnostic tool is present for the duration of learning.

We have maintained simplicity in designing the signal acquisition stages and the feature extraction stage which do not require any modification in the architecture or algorithms when new sources are found. Now, we would also like to explore the possibility of maintaining such simplicity in the design of the learning and classification stages, such that once again no modification of the decision rule is required when new sources are found. If no change in the decision rule or the features to be extracted are required, then obviously no change in the learning process will be necessary when new sources are found. So we set our target as to search for a decision rule that has the *universal* applicability to all of the interfering signals that are in our limited set, and possibly to an even larger set of interfering signals.

4.1.2 Investigation of Possible Approaches

As promised in the introduction to this report, we will discuss the approaches used in the area of *modulation recognition*, which appears to be the classification problem that is closest to ours. We postponed the discussion of the techniques used in modulation recognition until this chapter because, with respect to signal acquisition and feature extraction, the techniques were not directly applicable. In particular, impulsive noise and transient events are not present in the case of modulation recognition. Further, the bandwidth and carrier frequencies are assumed to be known, which greatly simplifies the problem.

A considerable amount of work appears to have been done with respect to modulation recognition, [32], [38], [39], [46], [52], [70], [71]. The difficulty in adapting the methods used in modulation recognition for our purpose is due to the fact that the feature extraction used in some of the work is very specific to a given problem (like [52], [32]) or the classification procedure is specific to a given problem (like [46]). In particular, we note that any form of classification algorithm that involves

sequential classification of the signals in question, where at each stage of the procedure the extracted features are compared to either eliminate certain signals or to make a conclusion, is not suitable for our application. Such a sequential procedure will have to be revised every time a new source is found. To equip the diagnostic tool with the capability of reorganizing the sequential classification procedure every time a new source is found would make the system software intensive, and we prefer to avoid such complexity.

The approaches used in [38], [39], [70], [71] appear to be very similar. Histograms were constructed for the acquired and processed waveforms, and the cell heights of the histograms were used as the components of the feature vector. The classification was performed by using either a linear or a polynomial decision function. Unfortunately, the coefficients of the decision functions have to be recalculated every time a new source is found, and this complicates the adaptive learning process.

By extracting specific features from the acquired data, we have inadvertently eliminated the use of distribution free methods[1] like histogram method discussed above. Although the histogram method has the advantage that we do not need to characterize the distribution of the feature vector, this method makes the feature vector very large, which we could tolerate, but the additional problem that arises in defining the cell width of the histograms is difficult to solve. In the case of modulation recognition, uniform cell division is possible because the bandwidth and the carrier frequency of the signals are known, the signals are modulated and typically the frequency hopping phenomenon is not considered, the problem of resolution due to different signals having a wide variety of different bandwidths does not occur, and impulsive or otherwise random behavior of the signals is not encountered[2].

[1]For more details on other distribution free methods like kernel estimators, k-nearest-neighbor methods and series expansion methods see [28].

[2]Once again we return to the well discussed example of distinguishing between narrow band FM signals (with frequency deviation less than 500 KHz) and pure tone signals. If the bandwidth spanned is 100 MHz, with 100 uniformly separated cells, both signals will have identical instantaneous and spectral frequency histograms. The reader should think about how to deal with comparing frequency histograms of the same signal which can have different mean frequencies at different times, or if the signal is hopping in frequency. Another problem is normalizing the amplitude envelope data, where in the case of modulation recognition, we could remove the spurious noise by median filtering, and

Although [32] uses a sequential evaluation procedure, in one of the stages, the comparison was made by using a *likelihood ratio* test, by assuming a Gaussian distribution for the feature in question. Since the likelihood ratio test is based on the *Bayes minimum error rule*, which is a fundamental algorithm often used in classification problems, we will explore the possibility of using this algorithm to generate our decision rule.

4.2 Algorithms for Learning and Classification

The Bayes minimum error rule states that, if a finite number of interfering signals c_k are considered, and a vector $\mathbf{v}$ is observed, then assign $\mathbf{v}$ to the class c_i if [28]

$$P(c_i|\mathbf{v}) > P(c_j|\mathbf{v}) \; for \; all \; j \neq i \tag{4.1}$$

For our application, the classes c_k correspond to the interfering signals, and the vector $\mathbf{v}$ is the feature vector. Since the *a posteriori* probabilities $P(c_k|\mathbf{v})$, are rarely known, we use the well known Bayes theorem

$$P(c_k|\mathbf{v}) = \frac{P(\mathbf{v}|c_k)P(c_k)}{P(\mathbf{v})} \tag{4.2}$$

and noting that $P(\mathbf{v}) = \sum_k P(\mathbf{v}|c_k)P(c_k)$, and assuming that all of the signals have the same *a priori* probability $P(c_k)$ (all signals are equally likely), and by defining

$$g_i(\mathbf{v}) = \frac{P(\mathbf{v}|c_i)}{\sum_k P(\mathbf{v}|c_k)} \tag{4.3}$$

we write the decision rule (4.1) as: *Assign* $\mathbf{v}$ *to class* c_i *if*

$$g_i(\mathbf{v}) > g_j(\mathbf{v}) \; for \; all \; j \neq i. \tag{4.4}$$

By assuming that all signals have the same a priori probability $P(c_k)$, we have reduced

simply normalize the data with respect to the maximum value, but in our case, we would like to preserve those spurious components since they may correspond to impulsive noise.

the Bayes minimum error rule to the *Maximum Likelihood* (ML) rule [66].

Now, if we knew the $P(\mathbf{v}|c_i)$, the probability distribution of the feature vector for a given interfering signal, then we could apply the decision rule (4.4). Although we could attempt to find approximate distributions for feature vectors of each of the interfering signals separately, this process will not only be tedious, but will also complicate the learning and classification process. So, we would like to explore the possibility of finding one distribution function that has acceptable approximation for all feature vectors from all of the interfering signals, with only the parameters of the distribution being different for different signals.

Naturally, the first candidate for such an attempt would be the well known Gaussian distribution. Although the Gaussian distribution is merely an abstract mathematical form, it often provides a good approximation to many natural distributions [28], probably due to the Central Limit Theorem. The Central Limit Theorem states that under rather general conditions, the distribution of the sum of a sufficiently large number of random variables tends to be Gaussian, even if the individual random variables are not Gaussian [20], [58]. We recognize that the features we have extracted were all in the form of statistical moments, and in most cases the features can be thought of as sums of random variables. Motivated by this discovery, we will assume that the feature vector extracted has a multivariate Gaussian distribution. Implications of this assumption on the performance of the classification will be discussed in Section 4.3.

4.2.1 The Learning Mode

Given that we have assumed a multivariate Gaussian distribution for the feature vector, the learning process is straightforward. Since a Gaussian distribution is completely characterized by its mean $\mathbf{m}$, and covariance matrix $\mathbf{\Sigma}$ [37], during the learning stage only these two values need to be estimated from the observed set of feature vectors.

The maximum likelihood estimate[3] of the mean, $\hat{\mathbf{m}}$, and the covariance matrix,

[3]The choice to use maximum likelihood estimation was arbitrary. We could have also used *Bayes*

$\hat{\boldsymbol{\Sigma}}$ are given by [21]

$$\hat{\mathbf{m}} = \frac{1}{N}\sum_{k=1}^{L} \mathbf{v}_k \tag{4.5}$$

and

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{k=1}^{L}(\mathbf{v}_k - \hat{\mathbf{m}})(\mathbf{v}_k - \hat{\mathbf{m}})^T \tag{4.6}$$

where N is the number of training sets used in the Learning mode.

Therefore, during the Learning mode, the user has to only *label*[4] the interfering signal, and ensure that the signal is present for the duration of learning. The diagnostic tool will be capable of making the required number of N measurements, compute the N feature vectors, estimate the mean and the covariance matrix for the interfering signal, and store them in the system library.

The value of N is one of the system parameters that we need to decide upon. Naturally a higher value of N would result in an improved estimate of the mean and covariance matrices, but this would also make the Learning mode more tedious for the user.

4.2.2 The Diagnostic Mode

The p-dimensional Gaussian density for the feature vector $\mathbf{v} = [v_1 \ldots v_p]^T$ has the form [37]

$$f(\mathbf{v}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{v}-\mathbf{m})^T\boldsymbol{\Sigma}^{-1}(\mathbf{v}-\mathbf{m})} \tag{4.7}$$

Since our feature vector consists of 6 components, $p = 6$. From the Learning mode, we will have estimates of $\mathbf{m}_i$ and $\boldsymbol{\Sigma}_i$ corresponding to signals c_i, and for our set of six interfering signals, $i = 1 \ldots 6$. So we could rewrite equation (4.3) as

$$g_i(\mathbf{v}) = \frac{f(\mathbf{v}|c_i)}{\sum_k f(\mathbf{v}|c_k)} \tag{4.8}$$

---

estimation, but the results obtained by the two procedures are often identical [21].

[4]This type of learning is termed *supervised learning*[21], because the user has to provide the label for the signal, i.e. inform the diagnostic tool that the signal is coming from a given signal, during the Learning mode.

where $f(\mathbf{v}|c_i)$ is given by

$$f(\mathbf{v}|c_i) = \frac{1}{(2\pi)^3|\mathbf{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}(\mathbf{v}-\mathbf{m}_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{v}-\mathbf{m}_i)} \tag{4.9}$$

Now, we could simply apply the ML decision rule (4.4) and make a decision in favor of the signal c_i that has the maximum $g_i(\mathbf{v})$ for the feature vector that is observed.

Recognizing that the denominator $\sum_k f(\mathbf{v}|c_k)$ in equation (4.8), and the factor $(2\pi)^3$ in the denominator of equation (4.9) are common to every signal, we could simplify the mathematical representation of the ML rule. However, the current form described by (4.4), (4.8) and (4.9) is convenient in deriving the criteria for declaring the *no diagnosis* state (introduced in Chapter 1) as a possible output from the Diagnostic Mode. The maximum value of $g_i(\mathbf{v})$, corresponding to the signal c_i which receives the favorable decision, will have values in the range $(1/r, 1)$, where r is the number of known interfering signals, for which the estimated mean and covariance matrix exists in the system library. In order declare the no-diagnosis state we should choose a threshold value p_{th} in the range $(1/r, 1)$, such that if the observed maximum value of $g_i(\mathbf{v})$ is below this threshold, then the no-diagnosis state should be declared. There is a trade-off to be considered in determining p_{th}. High values of p_{th} will not only reduce the misclassification rates, but also the correct classification rates. We recognize that p_{th} is a critical system parameter, but since the choice of p_{th} depends on the desired level of misclassification and correct classification rates, which will be specific to a given application, we will not discuss this issue any further.

4.3 The Performance of the ML Decision Rule

In Chapter 3, we focused on deriving a feature vector which has universal applicability, and a decision was made to extract the same features for all interfering signals. There are two concerns raised by this feature extraction process that the decision rule should account for. First, there may be correlation between the features, which need not be the same for different interfering signals. Second, the variance of the different features will in general be different for different signals, and it may be necessary to incorporate

appropriate weighting in the decision rule, to emphasize critical features and de-emphasize unreliable features for a specific comparison (for example, in comparing EASD and FM signals, the phase spread is critical, but the mean instantaneous frequency is unreliable).

The ML rule, based on the multivariate Gaussian assumption for the features, incorporates the covariance matrix in the decision making, thereby accounting for both of the above concerns. In particular, *automatic* weighting of the features takes place, emphasizing features with small variance, and de-emphasizing features with large variance, for a given comparison.

In deriving the Maximum Likelihood decision rule specified by (4.4), (4.8), and (4.9), we have made three assumptions. First, we assumed that the criterion specified by the Bayes minimum error rule is applicable to our problem. The Bayes minimum error rule it targeted towards minimizing the mean error in classification. There could be special circumstances where it may be necessary to set the target slightly differently. For example, if the specific error of classifying signal c_i as signal c_j is costlier that any other error, we may have to target towards minimizing the cost of this given error, instead of minimizing the mean error. However, since we are not aware of any such special circumstances, we will assume that minimizing the mean error is an acceptable target.

Second, we have assumed that the a priori probabilities of all of the signals, $P(c_i)$, are equal. In a given environment, the interfering signals encountered may not be equally likely to occur. Again, since we are concerned with the most general case, the assumption that the all of the interfering signals have equal a priori probabilities appears to be reasonable.

Third, we have assumed that the feature vector extracted has a multivariate Gaussian distribution. Although this is only an approximation, motivated by the Central Limit Theorem, we realize that if the error in the approximation is sufficiently large, then the performance of the ML decision rule will be significantly affected. Estimating the error in the approximation, either theoretically or experimentally, is not particularly useful since we are not concerned with the actual distribution of the fea-

ture vectors, but the effect of the approximation on the performance of the decision rule. In the following subsection, we describe a method by which we could assess the effects of the approximation.

4.3.1 The Theoretical Bayes Error

If we could compute the theoretical Bayes error that should result if the distribution of the feature vectors is strictly Gaussian, then we could compare the observed error rate with the theoretical error rate. Instead of computing the actual Bayes error, which will be a tedious task since the feature space that we have developed has 6 dimensions, we use the upper bound $\epsilon_u(c_i, c_j)$, for the actual Bayes error $\epsilon_{Bayes}(c_i, c_j)$ for two distributions c_i and c_j that are Gaussian, which is given by [25]

$$\epsilon_{Bayes}(c_i, c_j) \leq \epsilon_u(c_i, c_j) = \sqrt{P(c_i)P(c_j)}e^{-B(c_i,c_j)} \tag{4.10}$$

where the *Bhattacharyya Distance*, $B(c_i, c_j)$, between the two distributions c_i and c_j described by $N(\mathbf{m}_i, \mathbf{\Sigma}_i)$ and $N(\mathbf{m}_j, \mathbf{\Sigma}_j)$, respectively, is

$$B(c_i, c_j) = \frac{1}{8}(\mathbf{m}_i - \mathbf{m}_j)^T(\frac{1}{2}[\mathbf{\Sigma}_i + \mathbf{\Sigma}_j])^{-1}(\mathbf{m}_i - \mathbf{m}_j) + \frac{1}{2}\ln\frac{|\frac{1}{2}(\mathbf{\Sigma}_i + \mathbf{\Sigma}_j)|}{\sqrt{|\mathbf{\Sigma}_i||\mathbf{\Sigma}_j|}} \tag{4.11}$$

Since the actual mean vectors and covariance matrices are usually not known, we will have to use the maximum likelihood estimates for the mean vector and covariance matrix for each distribution. The upper bound of the Bayes error prescribed by (4.10) could be used to compare the distributions of the interfering signals, two at a time. By assigning $P(c_i) = P(c_j) = \frac{1}{2}$, we could write (4.10) as

$$\epsilon_u(c_i, c_j) = \frac{1}{2}e^{-B(c_i,c_j)} \tag{4.12}$$

So, if the observed error rates are less than or in the order of the upper bound on the Bayes error, than we would be able to conclude that the multivariate Gaussian approximation of the feature vectors has not significantly affected the performance of

the ML decision rule.

4.4 Chapter Summary

In this concluding chapter for Part I, we solved the problem of finding algorithms for learning and classification. The careful design of the feature extraction stage of Chapter 3 allowed the assumption of Gaussian distribution for the features extracted, and the use of the Maximum Likelihood decision rule, a special case of the Bayes minimum error rule, for the classification stage greatly simplified the learning and classification processes. Hence, we achieved our target for adaptive learning capability through a simple solution. The upper bound on the Bayes error was discussed to provide a method to assess the performance of the derived decision rule.

The developed architecture and algorithms will be implemented through simulation to evaluate the performance of the system, and the details will appear in Part II of this report.

Part II:
Performance Evaluation

Chapter 5

System Implementation

In Part I we developed a comprehensive approach towards adaptive classification of interfering signals. In Part II we will be concerned with evaluating the performance of the proposed scheme for interference diagnosis. In this chapter, we will discuss procedures that were written to stochastically simulate the interfering signals of Chapter 2, and to implement the architecture and algorithms developed in Chapters 1, 3 and 4. The procedures written have been divided into four packages, and the complete documented software appears in Appendix B. The first package provides some basic tools for the construction of models for interfering signals. The second package implements the models for the interfering signals based on the findings of Chapter 2, with several assumptions made to make the simulation complete. The third package implements the feature extraction strategy of Chapter 3, and the fourth package implements the remainder of the system, including the system parameters discussed in Chapter 1 and the algorithms for learning and classification discussed in Chapter 4. The optional hardware module described in Appendix A will not be implemented in our simulated system because we would like to understand the performance of the system without the optional module. All of the procedures are written in *Mathematica* and the reader who is unfamiliar with *Mathematica* should refer to [72].

5.1 The Basic Tool Kit

The basic tool kit for simulation consists of procedures **Gauss**, **Expo**, **FFT**, **IFFT**, **Mean**, **Variance**, **Audio**, and **Pulse**, and will be used extensively in constructing the models for interfering signals. The actual software written appears in the package **BasicTools.m**.

The procedures **Gauss** and **Expo** generate Gaussian and exponential random variables using the method described in [19], [43]. The procedures **FFT** and **IFFT** compute the Fast Fourier Transform and Inverse Fast Fourier Transform, respectively, of the input list. The procedures **Mean** and **Variance** compute the sample mean and variance of the input list. The procedure **Audio** generates a Gaussian noise that is bandlimited to audio frequencies (about 20 kHz).

The procedure **Pulse** takes as arguments an integer **L**, corresponding to the desired number of samples per observation, **bw** (in MHz), corresponding to the bandwidth captured, **rt** (in ns), corresponding to the risetime of the desired pulse, and **tc** (in ns) corresponding to the time constant for decay of the pulse, and returns a pulse that rises linearly and decays exponentially, satisfying the input parameters. This procedure will be used extensively, and we would like to make the following remarks concerning its usage:

- Since $e^{-5} \approx 0$, we will assume that the duration of the pulse approximately equals five time constants, provided that the risetime is very short.
- The bandwidth simulated and the sampling rate are equal, as per our findings through Theorem 1.3 of Chapter 1 (a bandpass bandwidth of $2W$ requires ideal sampling rate of $2W$ on two channels).
- The risetime of a pulse t_r, observed using a *lowpass* bandwidth of B_{lp} is related through [12]

$$B_{lp} \geq \frac{1}{2t_r} \tag{5.1}$$

Noting that the equivalent bandpass bandwidth B_{bp} is twice the lowpass band-

width, we write

$$B_{bp} \geq \frac{1}{t_r} \tag{5.2}$$

Relation (5.2) will be useful in manipulating rise times that were measured using a fixed bandwidth that is different from our bandwidth of interest

- Similarly, we will assume that the duration-bandwidth product is a constant in order to manipulate pulse duration statistics that were measured using a fixed bandwidth, provided that the actual bandwidth (unfiltered) is larger than our simulated bandwidth.

The procedures described in the preceding paragraphs will be used in the following section for implementing models of interfering signals.

5.2 Models for Interfering Signals

The procedures written to simulate the interfering signals are contained in the package `SourceModels.m`. The package consists of six parts, corresponding to the simulation of the wideband IQ signal acquisition described in Chapter 1, and a supplement that corresponds to the slow varying envelope acquisition.

In interpreting the models, the reader should keep in mind that the primary purpose of implementing the models in software is to allow Monte Carlo simulation of the interfering signals, and therefore the models may have several built in random variables. For example, in the case of EASD signals, the operating frequency is assumed to be uniformly distributed in a given range. This *does not* mean that the operating frequency is different for subsequent samples in the same observation, but simply means that the frequency need not be the same for different observations and hence we will generate a random variable for every observation (simulation) to represent this parameter.

5.2.1 Simulating the EASD Interference

As we discussed in Section 2.3.1 of Chapter 2, we will model the interference from electronic articles surveillance devices (EASD) as pure tone signals. We will assume that the operating frequency is uniformly distributed in $[f_o - 40$ MHz, $f_o + 40$ MHz], and thus the factor Δf_{EASD} will be uniformly distributed in [-40 MHz, 40 MHz]. The implementation of this models is very straightforward, and the procedure written could be found in `Part 1` of the `SourceModels.m` package.

5.2.2 Simulating the FM Interference

In Section 2.3.2 of Chapter 2, we developed a model for narrowband frequency modulated signals from amateur radio stations. Once again we will assume that the carrier frequency is uniformly distributed in $[f_o - 40$ MHz, $f_o + 40$ MHz], and thus the factor Δf_{FM} will be uniformly distributed in [-40 MHz, 40 MHz]. We will also assume that the maximum frequency deviation is uniformly distributed in [50 kHz, 500 kHz], and the message to be transmitted is Gaussian noise bandlimited to audio frequencies (hence we could use the output of the procedure `Audio` as the message). The procedure written to implement the model could be found in `Part 2` of the `SourceModels.m` package.

5.2.3 Simulating the SSB/FH Interference

Simulation of frequency hopped single sideband signals from amateur radio stations (SSB/FH) is somewhat more complicated than the previous two signals because of the frequency hopping phenomenon. In [65], it was recommended that amateur radio stations could use SSB signals with the carrier frequency hopping about ten times a second to reduce the distortions due to signal fading. However, such a slow hopping rate would make the problem very uninteresting since a hop would be very unlikely to be encountered in duration of observation that is much less than 1 ms. So, we will assume that the hop duration is uniformly distributed in [.1 ms, 1 ms].

As we will see in Chapters 6 and 7, duration of observation that is less than 100 μs

is often sufficient for the wideband signal acquisition, and the slow-varying envelope acquisition is not affected by the frequency hopping. So, we can expect to see no more than one hop (two carrier frequencies) in a given observation. In the procedure contained in `Part 3` of the `SourceModels.m` package, we first generate two carrier frequencies `z1` MHz and `z2` MHz. To make sure that the two frequencies do not turn out to be the same, `z1` will be an even integer uniformly distributed in [-40, 40] and `z2` will an odd integer uniformly distributed in [-39, 41]. Then the value for the hop duration, represented by the random variable `hop` is generated, and the probability `p` that a frequency hop could occur for the generated value of `hop` is computed, and incorporated into the procedure. The message to be transmitted is again obtained form the procedure `Audio`, and since the remaining details of the implementation are straightforward, the curious reader is referred to the procedure contained in `Part 3` of the `SourceModels.m` package.

5.2.4 Simulating the MWO Interference

The model for interference from microwave ovens (MWO) is by far the most complicated among the six models. The eight procedures written to simulate the interference, `MWoven`, `Collapse`, `Build`, `Npulse`, `Mpulse`, `Risec`, `StayC`, and `DropC`, could be found in `Part 4` of the `SourceModels.m` package.

We continue with the proposed state model of Figure 2-1 that was discussed in Chapter 2. We recall that the interference energy is emitted only during one half of the period of the 60 Hz cycle, and when the interference is emitted, the quantity Δt which has an approximate value of 3 μs, plays a key role.

There are approximately 2767 bins of width $\Delta t \approx 3$ μs in one half of the period of the 60 Hz cycle. Of these bins, the first and last n bins are occupied by the *build-up* and *build-down* pulses respectively, with one pulse per bin, and the peak amplitudes of the pulses rises uniformly from 0 to 1 in $n\Delta t$, during the build-up period, and drops uniformly during the build-down period. In the main procedure `MWoven`, we first generate the value of `nn`, corresponding to the n discussed above, which we model as an integer uniformly distributed in [1,10]. The values corresponding to the peak

amplitude during the build-up period is entered into `buf1`, and later a reversed version of `buf1` will be appended to represent the build-down period.

We define another variable $m = 2767 - 2n$, corresponding to the remaining bins in the one half of the 60 Hz cycle. During this period, continuous pulses are emitted, spaced Δt apart, and pulses may be missing (with probability .075, approximately) or pulses may collapse together to form continuous radiation. Now the main procedure `MWoven` makes a call to the procedure `Collapse` whose responsibility is to introduce the pulse collapsing phenomenon. We will assume that during the first and last 100 bins, pulse collapsing does not occur, and so `Collapse` returns a list of $m - 200$ elements, where a 1 indicates a normal pulse, 2 indicates the start of a collapsing event, 3 indicates that the pulses remain collapsed, and 4 indicates the end of the collapsing event. We have assumed that the probability that pulses may collapse at any given time (during the $m - 200$ bins) is .1, and that the number of bins for which they remain collapsed is uniformly distributed in [1, 10], not including the two bins for the rise and fall of the collapsing event.

The list returned by `Collapse` and the reversed version of `buf1` (to account for the pulse build-down period) are appended to `buf1` to form `buf2`. Further, `buf2` is randomly rotated to account for the fact that the trigger signal may be set at anytime during the one half period of 60 Hz when emissions occur, keeping in mind that half of the time we expect to see the trigger set at the beginning of build-up pulses. Then we take the first `s` elements from the list, where `s` is number of bins of width 3 μs in the duration of observation, which determines the type of emission that should occur in each of the observed bins. The procedure `Build` is given the responsibility of building the appropriate emissions for each of the bins. The procedure `Build` makes calls to subroutines `Npulse` (corresponding to the n bins of build-up and build-down pulses), `Mpulse` (corresponding to the remaining m bins), `RiseC`, `StayC`, and `DropC` (corresponding to the pulse collapsing event). Specifically, if the element is less than 1, then build-up (or build-down) pulses are constructed using `Npulse`. If the element is 1, then normal pulses, with the pulse missing phenomenon incorporated, are constructed using `Mpulse`. Likewise, the elements 2, 3, and 4, will

yield the emissions corresponding to the pulse collapsing event, using `RiseC`, `StayC`, and `DropC`.

We would like to make the following remarks concerning the envelope of the emissions:

- The pulses constructed by the procedure `Pulse` which rises linearly and decays exponentially will be assumed to describe the pulses emitted by the microwave oven.

- Although in [15] the risetimes of pulses were observed to be 5-10 ns, taking the relation (5.2) into consideration, we will assign the risetimes of the pulses as `Max[(1000/bw),5]` ns, where `bw` is the bandpass bandwidth (in MHz) captured.

- We will assume that the duration of the pulses are uniformly distributed in [200, 400] ns, and thus the time constant for decay will be uniformly distributed in [40, 80] ns.

- As we observed in Section 2.3.4 of Chapter 2, amplitude distribution for the pulses is always higher than the adjacent continuous carrier by about 10 dB, and so we add the appropriate envelope of the continuous carrier (indicated by the buffer `rf`, in subroutines `Npulse`, `Mpulse`, `RiseC`, `StayC` and `DropC`) to the pulses.

Since the microwave oven operates at a nominal frequency of 2.45 GHz, which is 8.25 MHz away from the center frequency of 2.4175 GHz of the 2.44 GHz band, we generate a carrier at $\Delta f = 8.25$ MHz, and modulate the carrier with the envelope obtained from the previous steps. The reader should compare the state model of Figure 2-1, with the procedures written in `Part 4` of the `SoureModels.m` package.

5.2.5 Simulating the PC interference

We rewrite equation (2.13) for the equivalent baseband representation of the Photocopier (PC) interference:

$$R_{PC}(t) = \sum_{k=1}^{q} A_k \hat{c}(t - T_k) e^{-i(2\pi f_o T_k + \theta_o)} \tag{5.3}$$

So, we could express the interference from photocopiers as a superposed sum of interference from q pulses, where q is the number of pulses that occur during a given observation.

We recall from Section 2.3.5 that the interarrival times of the pulses have an exponential distribution. The procedure **PhotoSpace** takes **L**, **bw** (in MHz), and **b** (in ns) as arguments and returns a list of pulse interarrival times that are exponentially distributed with mean **b** and sufficient to cover the duration of observation specified by the number of samples per observation **L** and the sampling rate of **bw** MHz. We will use a value of 220 ns for **b**, as observed in Section 2.3.5.

The individual pulses could be described by $A_k \hat{c}(t - T_k) e^{-i(2\pi f_o T_k + \theta_o)}$, and the procedure **PhotoPulse** is responsible for constructing the pulses. **PhotoPulse** takes the standard arguments **L** and **bw**, and the arguments **z** (the number of samples by which the pulse is delayed; corresponds to T_k) and **theta** (the phase offset θ_o of the first pulse). Since we have assumed that A_k is Gaussian, using the values of 12.7 dB, and 3.9 dB for the mean amplitude and standard deviation observed in [7] for photocopier emissions in the 2.44 GHz band, we generate a random variable with distribution N(1, .36) to represent A_k (this was done by converting the values 12.7 dB and 3.9 dB to a linear scale, and normalizing the mean to 1). Proposition 2.2 prescribed a $\hat{c}(t - T_k)$ in the form of response of a lowpass filter to a delayed ideal impulse, due to the assumptions made through Lemma 2.1. However, we will assume that a pulse generated by the procedure **Pulse**, and delayed accordingly, is sufficient for our purpose. In particular, using the relation (5.2) and that the actual bandwidth of the unfiltered pulse is much larger than the bandwidth of observation, we note that the risetime of the pulse should be 1/**bw**, where **bw** is the bandwidth of observation.

As for pulse duration, since the observation made in [7] used a bandpass filter with bandwidth 30 MHz, we need to estimate the pulse duration for bandwidths other than 30 MHz, which is done by assuming that the time-bandwidth product is constant (as we discussed in Section 5.1). Pulse duration statistic for the photocopier emissions in the 2.44 GHz band, using a bandpass filter with bandwidth 30 MHz, had a mean value of 143 ns, and we assume, for convenience, that the pulse durations are also exponentially distributed. Then the appropriate value for the time constant for decay is computed. With the knowledge of T_k and θ_o (phase offset of the first pulse), and by assuming f_o =2.4175 GHz, which is the center frequency of the 2.44 GHz band, we could compute the corresponding phase offset for each pulse.

The procedure `PhotoCopier` is the main procedure that first generates θ_o, and calls for the procedure `PhotoPulse` to construct the first pulse. Then the procedure `PhotoSpace` is called to obtain the pulse interarrival times, and repeated calls to `PhotoPulse` are made to construct all of the pulses, and finally the superposed sum of all the pulses is computed. The three procedures `PhotoCopier`, `PhotoSpace` and `PhotoPulse` could be found in `Part 5` of the `SourceModels.m` package.

5.2.6 Simulating the GN Interference

The interference from the theoretical bandlimited white Gaussian noise (GN) is implemented in the procedure `Noise`, contained in `Part 6` of the `SourceModels.m` package. The model is a straightforward implementation of the model discussed in Section 2.3.6.

5.2.7 Simulating the Slow-Varying Envelopes

There are two types of slow-varying envelopes, $s(t)$, that we discussed in Chapter 2. The first one is the $s(t)$ in the form of a 60 Hz square wave, corresponding to the emissions of microwave ovens, and is implemented in the procedure `MWOenv`. The second corresponds to the remaining five interfering signals, where $s(t) = 1$ for all t, and this is implemented in the procedure `STDenv`. The two procedures incorporate the lowpass filter with cutoff 300 Hz, and the sampling rate of 1200 HZ that was decided

in Chapter 1. Both procedures take one argument L, which is the number of samples per observation, and we will use L = 20, again as we decided in Chapter 1. The two procedures could be found in the Supplement to the SourceModels.m package.

5.3 Procedures for Feature Extraction

The package Feature Extraction.m consists of procedures InstFreq, CircVar, PSD, and Extract. The procedures InstFreq, CircVar, and PSD, which compute the instantaneous frequency, circular variance of estimated phase, and the power spectral density (periodogram), respectively, are direct implementations of equations (3.5), (3.9) and (3.10), respectively. The main procedure Extract takes a slow-varying envelope and a complex baseband signal as arguments and returns the the corresponding feature vector consisting of the six features. Since all of these procedures are a direct implementation of the strategy for feature extraction described in Chapter 3, no further details details will be discussed.

5.4 Procedures for System Simulation

The package SystemSimulation.m is the last of the four packages written, and consists of 9 procedures. The procedures Transmit1 and Transmit2 take a complex list and a real list respectively, and add appropriate Gaussian noise at the desired signal-to-noise ratio (SNR; expressed in dB). The procedures Digitize1 and Digitize2 take a complex list and a real list respectively, and the desired number of bits per sample, and return a digitized list. The procedures Fjitter and Pjitter, introduce frequency jitter and phase jitter from the local oscillator circuitry, at the desired rate. The details of the procedures Digitize1, Digitize2, Fjitter and Pjitter will be further discussed in Chapter 7.

The procedure Run takes arguments n, L, bw (in MHz) and snr (in dB), simulates each of the interfering signals n times (using the parameters L and bw), transmits these signals at the stated value of snr, optionally introduces digitization, frequency

jitter or phase jitter, and extracts the feature vectors. The results, containing the feature vectors, are left as global variables `bufA1` ... `bufC2`, which will be later used by the procedure **Learn**. The procedure **Learn** computes the maximum likelihood estimate of the mean vector and covariance matrix for each interfering signal, which is a direct implementation of the Learning Mode of Section 4.2.1, and leaves the results as global variables to be used by **Diagnose**. The procedure **Diagnose**, having access to the mean vectors and covariance matrices from the Learning Mode, takes a list of feature vectors and computes the likelihood values $g_i(\mathbf{v})$ according to equation (4.8), and returns a list of likelihood values for each feature vector in the input list. The classification (or declaration of the no-diagnosis state) can be made by visual inspection.

5.5 Chapter Summary

In this chapter, we discussed the procedures that were written to simulate the interfering signals, and to implement the Learning and Diagnostic modes discussed in Part I of this report. The procedures are contained in Appendix B of this report and will be used extensively for the purpose of performance evaluation, in Chapters 6 and 7.

Chapter 6

Validating the Scheme

This chapter is the first of the two chapters that will address the issue of system performance. We will first discuss a model for the simulation process and use the procedures of Appendix B, which were written to satisfy the model, to perform several experiments. There are two specific objectives that we would like to achieve through these experiments. First, we would like to evaluate the performance of the ideal system at different noise levels for validating the proposed scheme and to investigate if the system performance is within the theoretical bounds discussed in Chapter 4. Second, since the strategy for feature extraction played a very important role in the development of the scheme, we would like to understand the significance of each of the features, and compare the findings with the theoretical expectations discussed in Chapter 3.

6.1 Simulation Model

We introduce the concept of channel and receiver noise that we did not consider in Chapter 2. We assume that the noise is independent of the signal, and has a zero mean Gaussian distribution. So, the received slow-varying envelope $\tilde{s}(t)$ can be written as

$$\tilde{s}(t) = s(t) + n(t) \tag{6.1}$$

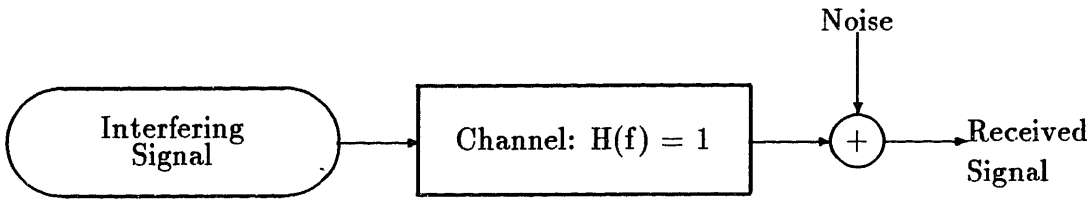

Figure 6-1: The Simulation Process: Both the interfering signal and the received signal will consist of a complex baseband signal, and a slow-varying envelope.

where $s(t)$ is the uncorrupted slow-varying envelope, and $n(t)$ is the zero-mean Gaussian noise with a non-zero variance σ_n^2, which is independent of $s(t)$. Likewise the received complex baseband signal, $\tilde{R}(t)$, consisting of in-phase and quadrature components, could be written as

$$\tilde{R}(t) = R(t) + n_I(t) + in_Q(t) \tag{6.2}$$

where $R(t)$ is the actual baseband signal, and $n_I(t)$ and $n_Q(t)$ are the in-phase and quadrature Gaussian noise components, each of them having zero mean, and non-zero variance σ_n^2, and independent of each other, and independent of the signal. For any given simulation, we define the signal-to-noise ratio (SNR), in dB, as

$$SNR = 10 \log_{10} \frac{E[y^2(t)]}{E[n^2(t)]} \tag{6.3}$$

where $y(t)$ could be the slow-varying envelope, the in-phase component, or the quadrature component, and $n(t)$ is the noise.

A model for the simulation process is shown in Figure 6-1. In addition to the assumption that the channel frequency response is constant for all frequencies of interest, we also assume that the receiving system is ideal. Specifically, this means that all filters exhibit an ideal behavior, the mixers (in the downconversion process) are strictly linear, and the sampling control circuitry is ideal, thus generating the trigger signal at exactly the start of the event of interest.

In the experiments to be performed in this chapter, we also assume that the local

oscillator and the associated power splitter do not exhibit any frequency or phase jitters, and that the analog-to-digital converters have an infinite number of bits. We will remove these two assumptions in the experiments to be performed in Chapter 7.

6.2 Ideal System Performance

The procedures `Run`, `Transmit1`, and `Transmit2`, which we introduced in Chapter 5, are capable of generating the received signals according to the model of Section 6.1. The Learning and Diagnostic modes were implemented using the procedures `Learn` and `Diagnose`. All experiments will be performed assuming an ideal bandpass bandwidth BW of 100 MHz, and as such the ideal sampling rate would be 100 MHz on each of the in-phase and quadrature channels. Sampling rate for the slow-varying envelope acquisition will be 1.2 kHz, as we decided in Chapter 1.

In the experiments to be performed in this chapter, we will fix the record length per observation, L, to be 2000 samples. The number of training sets to be used in the Learning mode, N, will be fixed at 100 sets per signal. We will also use the same sets for both learning and classification[1].

Now we are ready to perform our first experiment, which will be for the purpose of validating the proposed scheme. We generated 100 independent sets for each interfering signal at SNR 15 dB, and simulated the Learning and Diagnostic modes. The results are tabulated in the form of a *Confusion Matrix*, shown in Table 6-1. We repeated the experiment at SNR 10 dB and SNR 5 dB, and the results are tabulated in Table 6-2 and Table 6-3, respectively.

Perfect classification resulted when SNR was at 15 dB. The performance deteriorated to misclassification rates of 4.3 % and 7.5 % when the SNR was at 10 dB and 5 dB, respectively, and all of the confusion was between the signals EASD and FM. This observation does not surprise us because both of these signals are Type A interfering signals. Both EASD and FM signals have identical slow-varying and rapid-varying envelopes, and in both cases, the mean instantaneous frequency does

[1] Using different sets for training and classification will be introduced in Chapter 7

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2		100				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0 %

Table 6-1: Ideal System Performance at 15 dB SNR. Other experiment parameters were $BW = 100$ MHz, $L = 2000$ samples/set, $N = 100$ sets.

not provide any useful information. Although the FM signal has a marginally larger bandwidth than the EASD signal, instantaneous frequency and the spectral frequency data will not be helpful in distinguishing the two signals because the bandwidth captured was 100 MHz, which is much larger than the bandwidth of either of the signals. When the signals are corrupted by noise, the difference in the distribution of both the instantaneous frequency and the spectral frequency for the two signals will not be significant. Therefore, we see that the phase spread is the critical feature that is useful in distinguishing the two signals. The phase of the EASD signal, which ideally should be a constant, becomes more dispersed when the noise level rises, and thus the difference in the phase spread values for the two signals decreases. Hence we conclude that the results of the experiment are as expected.

In Chapter 4 we decided that we will compute the upper bound on the Bayes error, $\epsilon_u(c_i, c_j)$ by comparing the distributions of the interfering signals two at time, according to equations (4.11) and (4.12). However, since confusion resulted only between the EASD and FM signals, we need to compute the value of $\epsilon_u(c_i, c_j)$ only for the distributions of these two signals. Table 6-4 shows the computed values of the Bhattacharyya Distance, the upper bound on the Bayes error and the actual observed error, between the the signals EASD and FM, for the three noise levels. The Bhattacharyya Distance was computed using the maximum likelihood estimates

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	89	11				
FM	A2	15	85				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 4.3 %

Table 6-2: Ideal System Performance at 10 dB SNR. Other experiment parameters were $BW = 100$ MHz, $L = 2000$ samples/set, $N = 100$ sets.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	79	21				
FM	A2	24	76				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 7.5 %

Table 6-3: Ideal System Performance at 5 dB SNR. Other experiment parameters were $BW = 100$ MHz, $L = 2000$ samples/set, $N = 100$ sets.

Noise Level (SNR)	Bhattacharyya Distance, $B(c_i, c_j)$	Upper Bound on Bayes Error, $\epsilon_u(c_i, c_j)$	Observed Error
15 dB	2.72	0.033	0.000
10 dB	0.64	0.262	0.130
5 dB	0.34	0.355	0.225

Table 6-4: Confusion Between EASD and FM Signals. The observed error was computed from the results shown in Tables 6-1, 6-2, and 6-3.

of the mean vectors and covariance matrices, instead of the actual mean vectors and covariance matrices which are not known.

The observed error was less than the upper bound on the Bayes error at all levels of SNR for the confusion between EASD and FM signals. Since the confusion between the remaining pairs of signals was observed to be 0, we conclude that the performance of the ML decision rule, based on the multivariate Gaussian assumption for the features vectors, was within the limits of the upper bound on the Bayes error.

6.3 Understanding the Feature Vector

We have shown the viability of the proposed scheme for interfering signal classification. Since the strategy for feature extraction played a very important role in the development of the scheme, we would now like to understand the significance of each component of the feature vector, before attempting to the study the system performance with respect the system parameters.

In the experiments performed to evaluate the importance of each of the features, we used the same data sets that we generated in the experiment at SNR 15 dB of Section 6.2. However, we modified the learning and classification stages slightly. First, we removed feature v_1 from all of the feature vectors, and simulated the Learning and Diagnostic modes. Then we replaced feature v_1, and removed feature v_2. We repeated this step for all of the six features. The results of the six experiments are tabulated in Tables 6-5, 6-6, 6-7, 6-8, 6-9 and 6-10.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2		100				
SSB/FH	B1			99	1		
MWO	B2			2	98		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0.5%

Table 6-5: System Performance with Feature v_1 (Variance of the Slow-Varying Envelope) Removed. Parameters of the experiment were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

When the feature v_1, the variance of the slow-varying envelope, was removed from the classification process, the performance dropped from the 0 % misclassification of Table 6-1, to 0.5 % misclassification of Table 6-5. All of the confusion was between the signals from SSB/FH (frequency hopped single sideband signals) and MWO (microwave oven emissions). Naturally, since the slow-varying envelope was unique for the MWO, in the absence of this feature, confusion resulted between the MWO and SSB/FH, which is another Type B interfering signal.

When the feature v_2, variance of the rapid-varying envelope, was removed, the misclassification rate rose to 3 % as shown in Table 6-6. Ten of the SSB/FH data sets were classified as EASD signals and eight were classified as FM signals. When a hop does not occur during the observation, the SSB/FH signals will have a very narrow bandwidth, and a constant carrier frequency. In such a case the SSB/FH signals could be distinguished from the EASD signals by examining the rapid-varying envelope, since the envelope should be ideally a constant for the EASD signals. So, when this key feature is missing, confusion results between EASD and SSB/FH signals. Similarly, if the SSB/FH exhibits a hop to another carrier frequency that is very close to the first carrier frequency during the observation, or if the phase noise is sufficiently high, the SSB/FH signals would appear very similar to the FM signals if

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2		100				
SSB/FH	B1	10	8	82			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 3 %

Table 6-6: System Performance with Feature v_2 (Variance of the Rapid-Varying Envelope) Removed. Parameters of the experiment were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

the rapid-varying envelope is not used in the classification.

Removal of the feature v_3, the mean of the instantaneous frequency, resulted in a misclassification rate of 0.3 %, as shown in Table 6-7. One of the MWO signals was classified as SSB/FH, and one of the SSB/FH signal was classified as FM. The misclassification of the MWO signal was expected since the microwave oven has a nominal frequency of 2.45 GHz, and when the mean frequency is not considered in the classification, the MWO signal is likely to be confused for another Type B signal. But the confusion between SSB/FH and FM signal seems surprising at the first glance because we did not restrict the range of carrier frequencies for either of these signals. However, there is a logical explanation for this observation. The carrier frequency offset Δf_{FM} for the FM signals was assumed to be uniformly distributed in [-40 MHz, 40 MHz], and hence we expect the feature v_3 for FM signals to take values in the same range, with approximately equal probability. Although we made a similar assumption for the carrier frequency of the SSB/FH signals, these signals exhibit frequency hopping, and when a hop occurs during the observation, the value of v_3, which will be the weighted mean of the two carrier frequency offsets, is less likely to take values that are close to the two edges of the [-40 MHz, 40 MHz] range, and more likely to be closer to 0. So we see that the mean instantaneous frequency *does* play a

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2		100				
SSB/FH	B1		1	99			
MWO	B2			1	99		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0.3 %

Table 6-7: System Performance with Feature v_3 (Mean Instantaneous Frequency) Removed. Parameters of the experiment were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

role in classifying signals which do not have a restricted range of carrier frequencies. In the absence of this feature, the confusion between signals with constant carrier frequencies and signals that exhibit frequency hopping, increases marginally.

When the feature v_4, the variance of the instantaneous frequency, was removed from the feature vector, the misclassification rate was 0.3 %, as shown in Table 6-8. Two of the FM signals were misclassified as EASD signals. Although we do not expect the feature v_4 to be particularly useful in distinguishing between FM and EASD signals, the variance of the instantaneous frequency is marginally different for the two signals, and the slight increase in the confusion when v_4 is removed is not surprising.

In Chapter 3, we discussed the importance of the phase spread in distinguishing between EASD and FM signals. Results shown in Table 6-9, corresponding to the classification without v_5, the phase spread, confirms our argument. The misclassification rate rose significantly to 9.5 %, with all of the confusion occurring between the signals EASD and FM, as expected.

What was previously not expected was the lack of symmetry in the confusion because only 9 of the EASD signals were misclassified as FM signals, but almost half of the FM signals were misclassified as EASD signals. Since the variance of the slow-

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2	2	98				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0.3 %

Table 6-8: System Performance with Feature v_4 (Variance of Instantaneous Frequency) Removed. Parameters of the experiment were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	91	9				
FM	A2	48	52				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 9.5 %

Table 6-9: System Performance with Feature v_5 (Phase Spread) Removed. Parameters of the experiment were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	99	1				
FM	A2		100				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0.2 %

Table 6-10: System Performance with Feature v_6 (Local Spectral Bandwidth) Removed. Parameters of the experiment were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

varying and rapid-varying envelopes, and the mean instantaneous frequency are not useful in distinguishing between FM and EASD signals, in the absence of the phase spread, the weight of the classification falls on the unreliable features v_4, variance of the instantaneous frequency, and v_6, the local spectral bandwidth. The EASD signals generate approximately constant values for v_4 and v_6 for different observations, but since the maximum frequency deviation of the FM signals was a random variable, the values of v_4 and v_6 for FM signals will vary significantly between observations. When the maximum frequency deviation is relatively large, resulting in relatively large values of v_4 and v_6, the FM signals are classified correctly. But when the maximum frequency deviation is small, dilemma results. Since the ML decision rule make decisions in favor of the *most likely* signal, and since the EASD signal is more likely to exhibit small values of v_4 and v_6, the FM signal is misclassified as an EASD signal.

When the feature v_6, the local spectral bandwidth (variance of the spectral frequency), was removed from the feature vector, the misclassification rate was 0.2 %, as shown in Table 6-10. One of the EASD signals was confused for an FM signal. The local spectral bandwidth appears to be the least significant of all the features.

6.4 Chapter Summary

In this chapter we discussed the model that was used in simulation, and performed various experiments to investigate the viability of proposed scheme, and to understand the significance of the components of the feature vector. We found that the performance of the decision rule was within the theoretical limits specified by the Bhattacharyya distance and the upper bound on Bayes error. We also found that the features extracted had different levels of significance, with the phase spread and the variance of the rapid-varying envelope being the most critical, and the local spectral bandwidth being the least critical.

Chapter 7

System Parameters and System Performance

We continue with the performance evaluation of Chapter 6, taking some of the system parameters into consideration. The parameters that are of interest to us are the dynamic range of the sampling module, the duration of observation for the wideband IQ signal acquisition, frequency and phase jitters in the downconversion process, and the number of training sets required during the Learning mode. Experiments will be performed to characterize the dependence of the system performance on the system parameters. We set our target as to experimentally derive the viable range of the parameters such that the resulting mean misclassification rate does not exceed 1 %, with the channel and receiver noise at 15 dB SNR and when only one system parameter is considered. The derived set of initial system parameters can then be used as a background for future hardware experiments.

7.1 The Dynamic Range of the Sampling Module

In Chapter 6, we assumed that the available number of bits per sample was infinite, thus resulting in no noise due to digitization. Now we will remove that assumption and characterize the performance of the system with finite number of bits per sample. However, we will assume that the attenuator and the attenuation control circuit of

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2		100				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0 %

Table 7-1: System Performance with 8-bit Digitization. Other system parameters were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets

Chapter 1 are ideal, resulting in the maximum utilization of the dynamic range of the analog-to-digital converter.

We used the procedures `Digitize1` and `Digitize2` of Appendix B to introduce digitization of the signals. We generated 100 independent sets for each interfering signal, and performed the learning and diagnosis. The experiments were performed at SNR 15 dB, using a sampling rate of 100 MHz, training set size of 100, and a record length of 2000 samples per observation for the wideband IQ signal acquisition (the slow-varying envelope was acquired using a sampling rate of 1.2 kHz with a record length of 20 samples per observation, as before).

Three experiments were performed with the dynamic range represented by 8 bits, 6 bits and 4 bits per sample. The results are shown in Tables 7-1, 7-2, and 7-3. Perfect classification resulted with 8-bit digitization. The performance deteriorated to misclassification rates of .2 % and 1% with 6-bit and 4-bit digitizations, respectively. So, we conclude that 4-6 bits per sample should be sufficient to keep the mean misclassification rate at 1 % or less. However, memory is typically addressable in 8-bit elements, and 8-bit analog-to-digital converters currently available can support the desired sampling rates (as we saw in Chapter 1). As such, we could use an 8-bit digitization, leaving greater flexibility in the design of the attenuation control circuitry.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2	1	99				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: .2 %

Table 7-2: System Performance with 6-bit Digitization. Other system parameters were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	99	1				
FM	A2	5	95				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 1 %

Table 7-3: System Performance with 4-bit Digitization. Other system parameters were $SNR = 15$ dB, $BW = 100$ MHz, $L = 2000$ samples/set, and $N = 100$ sets.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	94	6				
FM	A2	11	89				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 2.8 %

Table 7-4: System Performance with $L = 1500$. Other system parameters were $SNR = 15$ dB, $BW = 100$ MHz, and $N = 100$ sets.

7.2 Duration of Observation

In Chapter 1, we decided on a fixed duration of observation of approximately 16.7 ms for the acquisition of the slow varying envelope. We also fixed the sampling rate at 1.2 kHz, requiring 20 samples for the intended duration of observation. Further verification of the choice of these parameters will require hardware experiments.

Now, we consider the issue of duration of observation for the wideband IQ signal acquisition. For a given sampling rate, the duration of observation will determine the record length L, corresponding to the number of samples per observation. The record length will subsequently determine the size of the memory required for the signal acquisition. Since the memory has to support high sampling rates (as we discussed in Chapter 1) and fast memory integrated circuits are expensive, and since a large record length would require more processing time, thereby delaying the diagnosis, we naturally would like to minimize the duration of observation.

In all of our experiments in Chapter 6, we used a record length L of 2000 samples/set with the sampling rate at 100 MHz, corresponding to a duration of observation of 20 μs. When the noise level was at SNR 15 dB, as we saw in Table 6-1, perfect classification resulted when $L = 2000$. We repeated the experiment at $L = 1500$, with the other parameters of the experiment remaining the same. The results are shown in Table 7-4. The misclassification rate was 2.8 %.

Since our target was to keep the mean misclassification rate at 1 % or below, we conclude that the record length L should be 2000, corresponding to a duration of observation of 20 μs. With 8 bits per sample, a memory of 2 kbytes for each of the in-phase and quadrature components does not appear to be unreasonable. However, it should be noted that the value of L would be higher in practice for the same duration of observation because the required sampling rate would be higher than the ideal sampling rate of 100 MHz that we have used (recall the factor of oversampling discussed in Chapter 2).

It should also be noted that the duration of observation of 20 μs is relevant only the given set of interfering signals that we have considered. For example, since the confusion was only between FM and EASD signals when L was 1500 samples/set, if one of the two signals was not in the set of interfering signals, then the choice of $L = 1500$ is sufficient. Also, the longer duration of observation was necessary because we have severely limited the bandwidth of the FM signal. If the maximum frequency deviation of the FM signal was larger than the values we had assumed in Chapter 5, then a smaller observation duration could be tolerated.

With the presence of frequency hopped signals, the choice of duration of observation becomes even more complicated. For example, if a given environment consists of mostly wideband signals (Types B2, C1 or C2), then we would like the frequency hopped signal (Type B1) to appear as a narrowband signal. So, we should set the observation duration to be less than the mean hop duration anticipated, so that the observed signal is less likely to involve a frequency hop. However, if the environment consists of mostly narrowband signals, then we would like the frequency hopped signal to appear as a wideband signal, and so we should set the duration of observation to be larger than the hop duration.

Therefore we conclude that the choice of the duration of observation is heavily dependent on the actual interfering signals anticipated in a given environment. When such information is not available, a nominal value of 20 μs may be used for the duration of observation of the wideband IQ signal acquisition.

7.3 Frequency and Phase Jitters

The topic of frequency and phase jitters, or phase noise, is actually a subset of the broader category of *frequency stability* [51]. The stability of an oscillating source decreases if the signal is anything other than a perfect sine function. Our concern with frequency stability is due the extensive use of phase dependent time-domain parameters in the feature extraction process. In particular, estimation of the phase offset to compute the phase spread (feature v_5) requires the correction for the carrier frequency offset, and we suspect that the frequency stability of the local oscillator would affect the estimated carrier frequency offset. In addition to that, the values for features v_3 and v_4 (mean and variance of the instantaneous frequency) would also be affected by frequency stability of the local oscillator.

The local oscillator circuitry, consisting of a local oscillator and a 90^o power splitter, is responsible for generating $\cos(2\pi f_o t)$ and $\sin(2\pi f_o t)$. These two signals will then be used in the quadrature downconversion process. There are specifically two kinds of phase noise that we are interested in. First, we will consider the case where the *same* jitter is present in both of the quadrature signals. Since $\sin^2\alpha + \cos^2\alpha = 1$ regardless of any jitter in α, in this case the amplitude of the signal will not be affected. Second, we will consider the case where the jitter is different on the two quadrature channels, thereby affecting the phase and the amplitude of the resulting baseband signal.

For convenience of modelling, we will assume that the frequency or phase jitter will be a uniformly distributed zero-mean stochastic process, where the adjacent samples, spaced apart by the sampling interval, are independent. We recognize that such modelling may not accurately describe the actual jitters likely to be encountered in the hardware implementation. However, since our purpose is only to investigate if such jitters should be given importance in the system design, we will not venture in to modelling the jitters accurately, and the curious reader is referred to [63] for a detailed treatment on phase noise in signal sources.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	97	3				
FM	A2	2	98				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 0.8 %

Table 7-5: System Performance with Frequency Jitter Uniformly Distributed in [-5 MHz, 5 MHz]. The adjacent samples, spaced apart by the sampling interval, were independent. Other system parameters were $L = 2000$, $SNR = 15$ dB, $BW = 100$ MHz, and $N = 100$ sets.

7.3.1 Frequency Jitter

Frequency and phase jitters may originate from both the local oscillator, which should ideally generate a sine wave at f_o, and the 90^o power splitter, which should ideally generate $\sin(2\pi f_o t)$ and $\cos(2\pi f_o t)$ to be used in the IQ downconversion process. In this subsection, we will assume that the 90^o power splitter is ideal, and that the phase jitter from the local oscillator can be absorbed into the frequency jitter. So we will consider only the frequency jitter from the local oscillator, and write the output $\hat{f}_o(t)$ of the non-ideal oscillator as $\hat{f}_o(t) = f_o + f_j(t)$ where $f_j(t)$ is a zero-mean stochastic process, and we assume that the adjacent samples of $f_j(t)$ that are spaced τ_s apart (where τ_s is the sampling interval) are independent and uniformly distributed. The procedure `Fjitter` of Appendix B is capable of introducing the frequency jitter.

We performed several experiments by varying the range of the distribution of the frequency jitter and Table 7-5 contains the results for the experiment that resulted in a mean misclassification rate that was close to 1 %. In this experiment $f_j(t)$ was uniformly distributed in [-5 MHz, 5 MHz]. The other system parameters were the same as the standard experiment of Table 6-1.

We recognize that the frequency jitter that we have assumed does not affect the

amplitude of the downconverted signal because the same frequency jitter is present in both the in-phase and quadrature components. Further, since we have assumed the frequency jitter to have a zero mean, the estimation of the carrier frequency offset is not significantly affected, and the phase spread factor remains reliable in distinguishing between the EASD and FM signals. The maximum tolerable range of [-5 MHz, 5 MHz] for the frequency jitter appears to be lenient enough to allow the use of most commercially available local oscillators.

7.3.2 Phase Jitter

Now we consider the case where the phase noise on the two quadrature channels are independent. For convenience of modelling, we assume that the local oscillator and the 90° power splitter exhibit only phase jitter (the frequency jitter, if any, is assumed to be absorbed by the phase jitter). Specifically this means that the two outputs of the local oscillator circuitry could be represented by $\cos[2\pi f_o t + \theta_I(t))]$ and $\sin[2\pi f_o t + \theta_Q(t))]$, and we recognize that the phase difference between the two outputs need not be 90° as it would be in an ideal system. We further assume that $\theta_I(t)$ and $\theta_Q(t)$ are two independent zero-mean stochastic processes, that have uniform distributions. Hence the phase noise in the two quadrature channels will in general not be the same. The procedure `Pjitter` of Appendix B is capable of incorporating the phase jitter.

We performed several experiments by varying the range of the jitter, until the observed mean misclassification rate was close to 1 %. The other parameters of the experiment were maintained at the same values of the experiment of Table 6-1. The results of the experiment where the phase jitter was uniformly distributed in $[-7^o, 7^o]$ are shown in Table 7-6.

One of the SSB/FH signals were confused for an FM signal. This observation is not surprising because the independent phase jitters on the two quadrature channels will affect the amplitude. As we saw in Table 6-6, the variance rapid-varying envelope (feature v_2) plays an important role in distinguishing between the SSB/FH and FM signals, and when the rapid-varying envelope is distorted, the error rate increases.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	98	2				
FM	A2	3	96	1			
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 1 %

Table 7-6: System Performance with Phase Jitter Uniformly Distributed in $[-7^o, 7^o]$. The phase jitter on the two quadrature channels were independent, and the adjacent samples of the jitter were also independent. Other system parameters were $L = 2000$, $SNR = 15$ dB, $BW = 100$ MHz, and $N = 100$ sets.

Since the system tolerates a phase jitter that is uniformly distributed in $[-7^o, 7^o]$, which appears to be within the reach of currently available local oscillators and 90^o power splitters, we conclude that phase jitter in the downconversion process is not a critical issue to be addressed in future hardware system design stages.

7.4 Training Set Size

In all of our previous experiments, we used 100 training sets during the Learning mode, with the same data sets used for the Diagnostic mode. As noted in Chapter 4, the number of training sets N is a critical parameter and we would like to explore the significance of N.

First, we would like understand the change in performance when independent sets are used for learning and classification. We performed an experiment similar to that of Table 6-1, maintaining $N = 100$, but by using independent sets for the learning and classification. The results are shown in Table 7-7.

The change in performance from perfect classification of Table 6-1 to a misclassification rate of .2 % when independent sets were used for learning and classification, appears to be negligible. Next, we repeated the experiments at $N = 50$, $N = 25$, and

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2	1	99				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: .2%

Table 7-7: System Performance with $N = 100$. The sets used for learning and classification were independent. Other system parameters were $L = 2000$, $SNR = 15$ dB, and $BW = 100$ MHz.

$N = 10$. The results are tabulated in Tables 7-8, 7-9 and 7-10.

Since there was no change in performance when N was reduced from 100 to 50, we conclude that the value of N should not be more than 50. The value of $N = 25$ resulted in a misclassification rate of 1 %, satisfying our target. However, we cannot conclude that $N = 25$ as being optimal because the choice of N is heavily dependent on the set of interfering signals in consideration. Clearly, as shown in Table 7-10, if the signals EASD and FM are not involved, then a value of 10 is sufficient for N. A critical parameter in determining the appropriate value of N is the variance of the features involved. If the variance of the features are small for an interfering signal, only a small number of measurements will be needed to characterize the distribution of the features for that signal. If the variance is large, then more measurements will be needed. For example, in the case of EASD and FM signals, the carrier frequency takes a wide range of values. So, the mean instantaneous frequency (feature v_3) will have a large variance. Therefore, to accurately characterize the distribution of v_3 for the two signals, a large number of measurements will be needed.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	100					
FM	A2	1	99				
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: .2%

Table 7-8: System Performance with $N = 50$. The sets used for learning and classification were independent. Other system parameters were $L = 2000$, $SNR = 15$ dB, and $BW = 100$ MHz.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	98	1	1			
FM	A2	2	97	2			
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 1 %

Table 7-9: System Performance with $N = 25$. The sets used for learning and classification were independent. Other system parameters were $L = 2000$, $SNR = 15$ dB, and $BW = 100$ MHz.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	98	1	1			
FM	A2	4	90	6			
SSB/FH	B1			100			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 2 %

Table 7-10: System Performance with $N = 10$. The sets used for learning and classification were independent. Other system parameters were $L = 2000$, $SNR = 15$ dB, and $BW = 100$ MHz.

7.5 Putting the Parameters Together

So far we performed experiments varying one parameter at a time, by setting our target as to achieve a mean misclassification rate of no more than 1 % at SNR 15 dB. Now, we remove the constraint concerning the mean misclassification rate and incorporate all of the system parameters discussed in this chapter, to perform one final experiment. The experiment was performed once again at SNR 15 dB, with the standard sampling rate of 100 MHz. The parameters introduced were 6-bit digitization, record length L = 2000 samples/set, frequency jitter that is uniformly distributed in [-5 MHz, 5 MHz], phase jitter that is uniformly distributed in $[-7^o, 7^o]$, and training set size $N = 25$, with independent sets for training and classification. The results are tabulated in Table 7-11.

The mean misclassification rate was 4 %. Although we did not set a target for the performance of the overall non-ideal system, the correct classification rate of 96 % provides sufficient encouragement for future hardware implementation of the scheme.

Interfering Signal		Classified As					
		A1	A2	B1	B2	C1	C2
EASD	A1	90	6	4			
FM	A2	7	87	6			
SSB/FH	B1	1		99			
MWO	B2				100		
PC	C1					100	
GN	C2						100

Mean Misclassification Rate: 4 %

Table 7-11: Performance of a Non-ideal System. Parameters of the system were SNR = 15 dB, BW = 100 MHz, 6-bit digitization, L = 2000, frequency jitter uniformly distributed in [-5 MHz, 5 MHz], phase jitter uniformly distributed in $[-7^o, 7^o]$, and N = 25.

7.6 Chapter Summary

In this chapter, we investigated the effect of some of the system parameters on the performance of the system. The parameters considered were the dynamic range of the sampling system, duration of observation for the wideband IQ signal acquisition, frequency and phase jitters, and the size of the training set. Whenever possible, we made specific conclusions about the viable values for the system parameters, emphasizing the fact that the interfering signals in consideration will dictate the final choice of the parameter values. The results of the investigation, consisting of a set of initial system parameters, is expected to provide sufficient background for future hardware experiments.

Chapter 8

Open Problems

We have achieved our objectives stated in the introductory chapter of this report. We have developed a scheme for adaptive classification of interfering signals, incorporating an architecture for signal acquisition, a strategy for feature extraction, and algorithms for classification and adaptive learning. We have also established a background for future work by categorizing the interfering signals and constructing mathematical models for some of the known interfering signals. The proposed scheme has been verified through Monte Carlo simulations and the performance of the system with respect to several of the system parameters has been characterized.

However, there are several tasks that have to be undertaken before the development of the desired diagnostic tool could be considered to be complete and ready for commercial production. Clearly, the logical next step is to implement the architecture and algorithms in hardware and verify the viability of the proposed scheme. There are several hardware design challenges that have to be met in the process, for example in the design of the sampling control circuitry to generate the trigger and attenuator input signals. The conclusions made in Chapter 7 concerning the specifications for the system parameters is expected to be helpful in the hardware implementation of the scheme. Although we used data reported by independent radio frequency surveyors in modelling some of the interfering signals, it is desirable that some empirical data concerning the interfering signals be obtained through hardware measurements, prior to the final hardware implementation of the system. Such measurements can

be used to verify our models, and will also be helpful in making design choices. For example, the actual strength of the received signals in a given environment will aid in designing the attenuation control circuitry. Experiments should also be performed to investigate the significance of the non-ideal behavior of some of the system components, like the filters and the mixers, that we did not consider in out theoretical treatment.

We did not address the issue of performing the diagnosis when multiple interfering signals are present, and when the network in consideration is still in operation. The presence of multiple interfering signals is not an unlikely event, and in order to incorporate the capability to perform the diagnosis, additional hardware and software design may be needed to separate the signals. Design issues related to performing the diagnosis while the network is still in operation is another related problem that will be a suitable topic for further research. Another interesting area for research would be to study the requirements to perform *dynamic* diagnosis, where the system makes repeated diagnosis. This option is possible because of the modest durations of observation required, as we discovered in Chapter 7.

Other topics that need to be addressed before the commercial production of the diagnostic tool involves issues like the appropriate user interface for the diagnostic tool, and the actual form of implementation the tool should take. As we discussed in Chapter 1, the diagnostic tool may be implemented in several ways, including in the form of a stand-alone diagnostic tool, or a diagnostic subsystem to be interfaced to another host system, which could be a personal (possibly wireless) computer or a communications test set.

The topics discussed above are directly related to the implementation of the desired diagnostic tool for indoor radio LANs. There are also several other areas which could form the subject of future research. As we mentioned in Chapter 2, the developed architecture and algorithms could be adapted to perform the interfering signal classification for applications other than indoor radio LANs. For example, we could construct a diagnostic tool for the diagnosis of interference faced by an electronic

article surveillance device[1]. Since the development of the architecture and algorithms were maintained to be independent of the specific application as far as possible, we expect the adaptation to other applications to be straightforward.

Another optional area for future work would be the extension of the proposed scheme to support a larger set of interfering signals. In particular, the categorization that we proposed could be revised as the characteristics of more interfering signals are understood, and models for a larger set of interfering signals could be constructed. The models will be useful in the verifying the applicability of the proposed strategy for feature extraction, and the algorithms for classification and learning, for a larger set of interfering signals. The models could also be used to provide a deeper understanding of the interference characteristics in a shared spectrum, which may result in improvements in the design of indoor radio networks.

[1]In a retail store for example, a point-of-sale terminal on a spread-spectrum network located within 50 feet of an electronic article surveillance device could conceivably jam the device [41].

Bibliography

[1] A. A. Al-Arainy, N. H. Malik, and M. K. Al-Bahloul, "Statistical Variation of ac Corona Pulse Amplitudes in Point-to-plane Air Gaps, " *IEEE Transactions on Electrical Insulation*, Vol. 24, No. 4, August 1989.

[2] Analog Devices, *Data Conversion Products Databook*, 1989/90.

[3] Louis Auslander, Izidor C. Gertner, and Richard Tolimieri, "The Discrete Zak Transform Application to Time-Frequency Analysis and Synthesis of Nonstationary Signals," *IEEE Transactions on Signal Processing*, vol.39, No. 4, pp. 825-835, April 1991.

[4] Henry Benitez, "Use a Spectrum Analyzer to Make EMI Measurements," *Microwaves & RF*, pp.134-142, April 1984.

[5] Charles Bertomier, "Instantaneous Frequency and Energy Distribution of a Signal," *Signal Processing 5*, pp.31-45, 1983.

[6] Kenneth L. Blackard, Theodore S. Rappaport, and Bruce Tuch, " Radio Frequency Noise Measurements and Models for Indoor Wireless Communications at 918 MHz, 2.44 GHz and 4.0 GHz," *EMC Test & Design*, pp. 25-33, July/August 1991.

[7] Kenneth L. Blackard, " Measurements and Models of Radio Frequency Impulsive Noise Inside Buildings," Master's Thesis, Virginia Polytechnic Institute and State University, October 1991.

[8] Boualem Boashash, Graeme Jones, and Peter O'Shea, "Instantaneous Frequency of Signals: Concepts, Estimation Techniques and Applications," *SPIE Vol. 1152 Advanced Algorithms and Architectures for Signal Processing IV*, pp. 382-400, 1989.

[9] Boualem Boashash, and Langford B. White, "Instantaneous Frequency Estimation and Automatic Time-Varying Filtering," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp.1221-1224, 1990.

[10] J. L. Brown, "On Quadrature Sampling of Bandpass Signals," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 366-371, Vol. AES-15, No. 3, May 1979.

[11] John L. Brown, "First Order Sampling of Bandpass Signals - A New Approach," *IEEE Transactions on Information Theory*, pp. 613-615, Vol. IT-26, No.5, September 1980.

[12] A. Bruce Carlson, *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*, Third Edition, McGraw-Hill, 1986.

[13] J.Y.C. Cheah, "IEEE 802.4L submission on microwave oven interference measurement", *IEEE p802.4L/90-08a.*, Feb. 1990.

[14] J.Y.C. Cheah, "Interference Characteristics of Microwave Ovens in Indoor Radio Communications", *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 280-285, London, Sept. 1991.

[15] J. Y. C. Cheah, Hughes Network Systems, private communication, Sept. 1991.

[16] Leon Cohen, "Time Frequency Distributions - A Review," *Proceedings of the IEEE*, Vol.77, No.7, pp. 941 - 981, July 1989.

[17] Leon Cohen, and Chongmoon Lee, "Local Bandwidth and Optimal Windows for the Short Time Fourier Transform," *SPIE Vol.1152 Advanced Algorithms and Architectures for Signal Processing IV*, pp. 401-425, 1989.

[18] George C. Cooper, and Clare D. McGillem, *Modern Communications and Spread Spectrum*, McGraw-Hill, 1986.

[19] Narsingh Deo, *System Simulation with Digital Computer*, Prentice-Hall, 1979.

[20] John J. Downing, *Modulation Systems and Noise*, Prentice-Hall, 1964.

[21] Richard O. Duda, and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[22] Luis A. Duran, "Characteristics of Electromagnetic Interference in Industrial Environments," Master's Thesis, Purdue University, 1987.

[23] John E. Freund, and Ronald E. Walpole, *Mathematical Statistics*, Fourth Edition, Prentice-Hall, 1987.

[24] B. Friedlander, B.Porat, "Detection of Transient Signals by Gabor Representation", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No.2, February 1989, pp. 169-180.

[25] Keinosuke Fukunaga, and Raymond R. Hayes, "Effects of Sample Size in Classifier Design," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 8, pp. 873-885, August 1989.

[26] William A. Gardner, *Statistical Spectral Analysis*, Prentice Hall, 1988.

[27] Madhu Sudan Gupta, "Definition of Instantaneous Frequency and Frequency Measurability," *American Journal of Physics*, vol. 43, No.12, pp.1087-1088, December 1975.

[28] D. J. Hand, *Discrimination and Classification*, John Wiley & Sons, 1981.

[29] N. Harid, and R. T. Waters, "Statistical Study of Impulse Corona Inception Parameters on Line Conductors," *IEE Proceedings-A*, Vol. 138, No.3, May 1991.

[30] Hewlett-Packard Co., *HP 71000 Modular Spectrum Analyzer Operation Manual*, pp. Glossary 15-16, August 1989.

[31] Hewlett-Packard Co., "RF Communications Test Set Services Radio Telephones," *Microwaves & RF*, p.131, June 1991.

[32] S.-Z. Hsue, Samir S. Soliman, "Automatic Modulation Classification Using Zero Crossings," *IEE Proceedings*, Vol. 137, Pt. F, No.6, December 1990, pp.459-464.

[33] Keng D. Hsueh and Roger P. Hamernik, "A Generalized Approach to Random Noise Synthesis: Theory and Computer Simulation," *Journal of Acoustical Society of America*, Vol.87, No.3, pp.1207-1217, March 1990.

[34] C. Huang, R. Khayata, "Wideband Indoor Radio Propagation Channel Measurements At ISM Bands", Submitted for publication in *IEEE Transactions on Vehicular Technology.*

[35] Annette B. Jaffe, and Donald M. Burland, "Electrophotographic Printing, " *Output Hardcopy Devices*, pp. 221-260, Ed. Robert C. Durbeck, and Sol Sherr. Academic Press, 1988.

[36] Abdul J. Jerri, "The Shannon Sampling Theorem - Its Various Extensions and Applications: A Tutorial Review," *Proceedings of the IEEE*, pp. 1565-1596, Vol.65, No. 11, November 1977.

[37] Richard A. Johnson, and Dean W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice-Hall, 1982.

[38] F. Jondral,"Automatic Classification of High Frequency Signals", *Signal Processing*, Vol. 9, Oct. 1985, pp. 177-190.

[39] Friedrich Jondral, "Foundations of Automatic Modulation Classification," *Stochastishe Modelle und Methoden in der Informationstechnik, ITG-Fachbericht n. 107*, Published by VDE-Verlag GmbH, Berlin 12, 1989 (in English).

[40] Mohammed M. Khalifa, Ahmad A. Kamal, Assaad G. Zeitoun, Roshdy M. Radwan, and Sayed El-Bedwaihy, "Correlation of Radio Noise and Quasi-Peak Measurements to Corona Pulse Randomness," *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-88, No. 10, pp. 1512-1521, October, 1969.

[41] Matt Kramer, "Wireless LANs Are Sluggish but Useful," *PC Week*, p. 91, June 3, 1991.

[42] B. P. Lathi *Signals, Systems and Communication*, John Wiley & Sons, 1965.

[43] Averill M. Law, and W. David Kelton, *Simulation Modelling and Analysis*, McGraw-Hill, 1982.

[44] LeCroy Corporation, *Reference Guide to Digital Waveform Instruments*, 1990.

[45] Edward A. Lee, and David G. Messerschmitt, *Digital Communications*, Kluwer Academic Publishers, 1988.

[46] F.F. Liedtke, " Computer Simulation of an Automatic Classification Procedure for Digitally Modulated Communications Signals with Unknown Parameters", *Signal Processing*, Vol. 6, pp. 311-323, August 1984.

[47] L. Mandel, "Interpretation of Instantaneous Frequencies," *American Journal of Physics*, vol. 42, pp. 840-846, October 1974.

[48] K. V. Mardia, *Statistics of Directional Data*, Academic Press, 1972.

[49] N. H. Malik, and A. A. Al-rainy, "Statistical Variation of DC Corona Pulse Amplitudes In Point-To-Plane Air Gaps, " *IEEE Transactions on Electrical Insulation*, Vol. EI-22, No. 6, pp. 825-829, December, 1987.

[50] Michael J. Marcus, "Regulatory Policy Considerations for Radio Local Area Networks," *IEEE Communications Magazine*, Vol. 25, No. 7, pp. 95-99, July, 1987.

[51] Catherine M. Merigold, "Phase-Noise Measurements," *Telecommunications Measurements, Analysis, and Instrumentation*, pp. 361-402, Ed. Dr. Kamilo Feher, Prentice-Hall, 1987.

[52] Stanton B. McMillan, Brian P. Flanagan, and Tom K. Doong, "Determination of the Modulation Type of Communication Signals," *Proceedings of The IEEE*

International Conference on Acoustics, Speech, and Signal Processing, pp. 1683-1686, 1990.

[53] David Middleton, "Statistical-Physical Models of Electromagnetic Interference," *IEEE Transactions on Electromagnetic Compatibility*, Vol.EMC-19, No. 3, pp. 106-127, August 1977.

[54] Nirode Mohanty, *Random Signals Estimation and Identification*, Van Nostrand Reinhold Company, 1986.

[55] D.B. Newman Jr., "FCC authorizes spread spectrum", *IEEE Communications Magazine*, pp. 46-47, July, 1986.

[56] Alan V. Oppenheim, and Ronald W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, 1989.

[57] Northe K. Osbrink, and James F. Girand, "New Offerings Put Pressure on Public-Use Spectrum," *Microwaves & RF*, pp.45-50, April, 1991.

[58] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Second Edition, McGraw-Hill, 1984. and 13.

[59] Hap Patterson, Director of Research and Development, Sensormatic Electronics Corp., private communication, October 1991.

[60] Peyton Z. Peebles, Jr., *Digital Communication Systems*, Prentice-Hall, 1987.

[61] Pike & Fisher, Inc. *Equipment Rules Services*, 1991.

[62] M. B. Priestley, *Spectral Analysis and Time Series*, Volume 1 and 2, Academic Press, 1981.

[63] W. P. Robins, *Phase Noise in Signal Sources* , Peter Peregrinus Ltd. (London, UK), 1982.

[64] K. Sam Shanmugan, and A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*, John Wiley & Sons, 1988.

[65] Donald L. Schilling, Laurence B. Milstein, Raymond L. Pickholtz, Marvin Kullback, and Frank Miller, "Spread Spectrum for Commercial Communications," *IEEE Communications Magazine*, pp. 66-79, April, 1991.

[66] Jeffrey H. Shapiro, and Alan S. Willsky, *6.432 Supplementary Notes*, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Fall 1988.

[67] W. M. Siebert,*Circuits, Signals and Systems*, MIT Press, 1986, pp 554-555.

[68] Edward N. Skomal, *Man-Made Radio Noise*, Van Nostrand Reinhold Company, 1978.

[69] Edward N. Skomal, and Albert A. Smith, Jr., *Measuring The Radio Frequency Environment*, Van Nostrand Reinhold Company, 1985.

[70] L. Vergara Dominguez, J. M. Paez Borralo, J. Portillo Garcia, and B. Ruiz Mezcua, " A Radiocommunication Signal Classifier," *Proc. EUSIPCO-88*, Grenoble, France, September 1988, pp. 1361-1364.

[71] L. Vergara Dominguez, J. M. Paez Borralo, J. Portillo Garcia, and B. Ruiz Mezcua, " A General Approach to the Automatic Classification of Radiocommunication Signals," *Signal Processing 22*, pp.239-250, 1991.

[72] S. Wolfram, *Mathematica: A System For Doing Mathematics by Computer*, Addison-Wesley, 1988.

[73] Serena M. Zabin, and H. Vincent Poor, "Recursive Algorithms for Identification of Impulsive Noise Channels," *IEEE Transactions on Information Theory*, Vol. 36, No.3, May 1990.

[74] Rodger E. Ziemer, and Roger L. Peterson, *Digital Communications and Spread Spectrum Systems*, Macmillan Publishing Company, 1985.

[75] Goran D. Zivanovic, "On the Instantaneous Frequency of Cyclostationary Random Signal," *IEEE Transactions on Signal Processing*, vol.39, No.7, July 1991, pp. 1604-1610.

Appendix A

Narrowband Signal Acquisition

This appendix could be thought of as an extension to Chapter 1 because of the hardware nature of the topic. However, to fully appreciate the discussion, knowledge of the interfering signal Types A1, A2, and B1 of Chapter 2, and the feature extraction process of Chapter 3 will be necessary, and the reader is advised to have read these two chapters prior to reading this appendix.

One of the difficulties in the acquisition and analysis of interfering signals is that the bandwidth and the central frequency of the signals is always an unknown. In particular, if the signal occupies only a fraction of the bandwidth allocated for a given band, but located at a center frequency which will appear to be a random variable to an independent observer, then the analysis of the signal based on the wideband I and Q signal acquisition we discussed previously, would be very difficult because of *frequency resolution* issues. For example, consider distinguishing between a pure tone signal, and a narrowband FM signal with frequency deviations less than 500 KHz, both in a band having a width of 100 MHz. Both of these signals will almost look alike in the amplitude plots, and instantaneous or spectral frequency plots (see Chapter 3 for details). Although we have developed methods to solve this difficulty by proper choice of feature extraction strategy, it may still be desirable to incorporate an additional (but optional) hardware module capable of acquiring narrowband signals with greater resolution. The primary purpose of the optional hardware module would be to capture only the small fraction of the bandwidth containing the narrowband

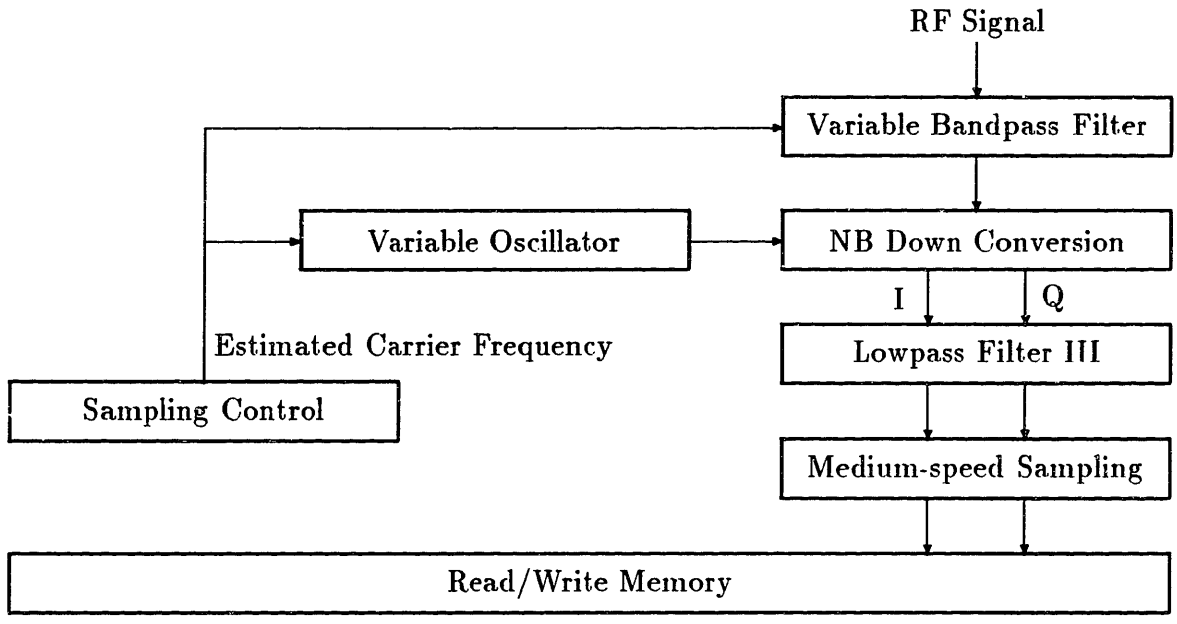

Figure A-1: The Optional Hardware Module for Narrowband Signal Acquisition

signal, instead of capturing the entire band.

Figure A-1 illustrates a possible approach that could be taken for acquiring narrowband signals. Naturally, the first problem is to estimate the carrier frequency of the signal, from the first few samples obtained form the wideband signal acquisition hardware. Therefore, we assign an additional task to the Sampling Control circuitry we saw before in Figure 1-3. The carrier frequency may be estimated by computing the mean of the instantaneous frequency of the first few samples, similar to the method of computing feature v_3 (the mean instantaneous frequency) discussed in Chapter 3. We estimate the carrier frequency using only the first few samples because, if we wait any longer, the interfering signal might disappear, or may hop to a different carrier frequency. Since only the carrier frequency can be estimated reliably in a short period of time, the bandwidth of the variable bandpass filter should be fixed (to a fraction of the total bandwidth of the band), with only the center frequency as the variable.

Therefore, the narrowband signal acquisition will proceed simultaneously with the wideband signal acquisition, with a small start-up delay due to the carrier frequency

estimation and settling times for the variable oscillator and the variable bandpass filter. Since the data from the narrowband signal acquisition will have no significance when the signal encountered is a wideband signal (i.e. wider than the bandwidth of the variable bandpass filter), during the signal analysis stage, only the data from the wideband signal acquisition should be looked at first. If it is found that the signal occupied only a fractional bandwidth during the duration of observation, then the wideband data should be replaced with the narrowband data. Otherwise, the narrowband data should be simply discarded. Hence, the purpose of the narrowband signal acquisition is not to provide additional information about the signal, but more reliable information when the signal is a narrowband signal.

Appendix B

Procedures for Interfering Signal Simulation and Classification

```
(*************************************************************************)
(*  Package Name: BasicToolKit.m                                         *)
(*                                                                       *)
(*  (c) Ganesh N. Ramaswamy                              November, 1991  *)
(*************************************************************************)
(* PROCEDURES IN THE PACKAGE:                                            *)
(* Gauss: Takes one argument L and returns a list of length L            *)
(*        containing Gaussian random variables.                          *)
(* Expo: Takes one argument b and returns an exponentially               *)
(*       distributed random variable with mean b.                        *)
(* FFT: Computes the Fast Fourier Transform of the input list            *)
(*      (Mathematica's Fourier Expansion corresponds to the              *)
(*      conventional Inverse Fourier Transform).                         *)
(* IFFT: Computes the Inverse (Fast) Fourier Transform of the input      *)
(*       list.                                                           *)
(* Mean: Computes the mean of the input list.                            *)
(* Variance: Computes the variance of the input list.                    *)
(* Audio: Takes arguments L, and bw (in MHz), and returns L samples      *)
(*        of Gaussian noise bandlimited to audio frequencies             *)
(*        (about 20 kHz), sampled at rate bw MHz.                        *)
(* Pulse: Takes arguments L, bw (in MHz), rt (in ns), and tc (in ns),    *)
(*        and returns L samples corresponding to one pulse starting      *)
(*        at time = 0, with linear risetime rt ns, and exponential       *)
(*        fall time with time constant tc ns, sampled at bw MHz.         *)
(*************************************************************************)
```

```
Gauss[L_Integer] :=
 Block[ {ans, r1, r2, r3, r4},
  r1  = Table [Random[], {L}];
  r2  = Table [Random[], {L}];
  r3  = Sqrt[-2 Log[r1]];
  r4  = Cos[2 Pi r2] //N ;
  ans = r3 r4;
  Return[ans] ]

Expo[b_Integer] :=
 Block[ {ans, r1},
  r1  = Random[];
  ans = -b Log [r1];
  ans = Round[ans];
  Return[ans] ]

FFT[list_] := InverseFourier[list]

IFFT[list_] := Fourier[list]

Mean[list_] := Apply[Plus, list]/Length[list]

Variance[list_] := Mean[(list - Mean[list])^2]

Audio[L_Integer, bw_Integer] :=
 Block[ {ans, buf1, buf2, z1, z2, n},
  z1   = (20 10^3) / (.5 bw 10^6);
  n    = Round[8 / z1];
  n    = Max[L, n];
  z2   = Round[.5 z1 n];
  buf1 = FFT[Gauss[n]];
  buf2 = Join[Table[1, {z2}], Table[0,{n - 2 z2}], Table[1,{z2}]];
  buf2 = buf1 buf2;
  ans  = Re[IFFT[buf2]];
  ans  = Take[ans, L];
  ans  = ans / Max[Abs[ans]];
  Return[ans] ]

Pulse[L_Integer, bw_Integer, rt_Integer, tc_Integer] :=
 Block[ {ans, buf1, buf2, z},
  z    = Divide[1, bw 10^6];
  buf1 = Table [ Divide[t z, rt 10^-9], {t, 0, Ceiling[ rt 10^-9 / z]}];
  buf1 = buf1 / Max[buf1];
  buf2 = Table[ E^(-t z / (tc 10^-9)), {t, 1, (L - Length[buf1])}];
```

```
  ans  = Join[buf1, buf2];
  ans  = ans // N;
  Return[ans]  ]

(************************ End of Package ****************************)
```

```
(*********************************************************************)
(*  Package Name: SourceModels.m                                     *)
(*                                                                   *)
(*  (c) Ganesh N. Ramaswamy                          November, 1991  *)
(*********************************************************************)
(* PROCEDURES IN THE PACKAGE:                                        *)
(* The procedures in this package are divided into six parts,        *)
(* corresponding to each interfering signal, and one supplement      *)
(* for the slow varying envelope generation.                         *)
(*********************************************************************)

(************************** PART 1 ***********************************)
(* Interfering Signal Simulated: Electronic Article Surveillance     *)
(*                               Devices (EASD); Type A1.            *)
(* Implemented in a single procedure EASD. Takes L and bw (in MHz)   *)
(* as arguments, and internally generates a frequency offset that is *)
(* uniformly distributed in [-40 MHz, 40 MHz], and returns L samples *)
(* of EASD signals, sampled at bw MHz                                *)
(*********************************************************************)

EASD[L_Integer, bw_Integer] :=
 Block[ {ans, z, w, theta,  per},
  theta = 2 Pi Random[];
  z     = Random[Real, {-40, 40}];
  w     = 2 Pi z 10^6 // N;
  per   = Divide[1, bw 10^6];
  ans   = Table[Cos[w t per + theta] + I Sin[w t per + theta], {t, 1, L}];
  ans   = ans // N;
  Return[ans] ]

(************************** PART 2 ***********************************)
(* Interfering Signal Simulated: Narrowband FM signals (FM);         *)
(*                               Type A2.                            *)
(* Implemented in a single procedure FM. Takes L and bw (in MHz) as  *)
(* arguments, and internally generates a carrier frequency offset    *)
(* that is uniformly distributed in [-40 MHz, 40 MHz], and calls for *)
(* the procedure Audio, and generates a frequency modulated signal,  *)
(* with maximum frequency deviation (indicated by k) that is         *)
(* uniformly distributed in [50 kHz, 500 kHz], and returns L samples *)
(* sampled at bw MHz                                                 *)
(*********************************************************************)
```

```
FM[L_Integer, bw_Integer] :=
 Block[ {ans, buf1, buf2, z, w, theta, k,  per},
  theta = 2 Pi Random[];
  z     = Random[Real, {-40, 40}];
  w     = 2 Pi z 10^6 // N;
  per   = Divide[1, bw 10^6];
  buf1  = Audio[L, bw];
  buf2  = Table[ Apply[Plus, Take[buf1, t]], {t, 1, L}];
  buf2  = buf2 per;
  k     = 2 Pi 10^3 Random[Real, {50, 500}];   (* k in kHz *)
  buf2  = k buf2 // N;
  ans   = Table[Cos[w t per + buf2[[t]] + theta]
            + I Sin[w t per + buf2[[t]] + theta], {t, 1, L}];
  ans   = ans // N;
  Return[ans] ]

(************************** PART 3 ***********************************)
(* Interfering Signal Simulated: Frequency Hoppped Single Sideband   *)
(*                               Signals (SSB/FH); Type B1           *)
(* Implemented in a single procedure SSBfh. Takes L and bw (in MHz)  *)
(* as arguments. Assumes that no more than two carrier frequencies   *)
(* will be involved in one observation, and generates two carrier    *)
(* frequencies (z1 and z2) that are always different, and uniformly  *)
(* distributed in [-40 MHz, 40 MHz]. A hop duration (hop), which is  *)
(* uniformly distributed in [.1 ms , 1 ms], and the probability that *)
(* a hop occurs during observation (p) are calculated. Internal call *)
(* for Audio provides the message for transmission, and an SSB signal*)
(* is generated, and L samples sampled at bw MHz are returned.       *)
(*********************************************************************)

SSBfh[L_Integer, bw_Integer] :=
 Block[ {ans, buf1, buf2, buf3, z1, z2, hop, p, w1, w2, l,theta1,
         theta2,  per},
  theta1 = 2 Pi Random[];
  theta2 = 2 Pi Random[];
  z1     = 2 Random[Integer, {-20, 20}];
  z2     = 2 Random[Integer, {-20, 20}] + 1;
  hop    = 0.1 Random[Integer, {1, 10}];  (* hop dur .1 - 1 ms *)
  p      = L / (bw 10^3 hop);
  p      = .5 (p + 1) // N;
  p      = Round[Random[Real, {0, p}]];
  l      = p Random[Integer, {1, L}];
  w1     = 2 Pi z1 10^6 // N;
  w2     = 2 Pi z2 10^6 // N;
```

```
  per    = Divide[1, bw 10^6];
  buf1   = FFT[Audio[2 L, bw]];
  buf1   = Drop[buf1, -L];
  buf1   = IFFT[buf1];
  buf2   = Table[Cos[w1 t per + theta1] + I Sin[w1 t per + theta1],
                 {t, 1, (L - 1)}];
  buf3   = Table[Cos[w2 t per + theta2] + I Sin[w2 t per + theta2],
                  {t, 1, 1}];
  buf2   = Join[buf2, buf3];
  ans    = buf1 buf2;
  ans    = ans // N;
  Return[ans] ]

(************************** PART 4 ********************************)
(* Interfering Signal Simulated: Microwave Oven Emissions (MWO);     *)
(*                              Type B2.                             *)
(* PROCEDURES IN PART 4:                                             *)
(* MWoven: The main procedure that takes L and bw (in MHz) as        *)
(*         and returns L samples, sampled at bw MHz. First generates *)
(*         heights of build up pulses, and then calls for Collapse,  *)
(*         and Build. Accommodates for the fact that observation     *)
(*         may start at any time when emissions are occurring (so    *)
(*         buf2 is rotated randomly). See text for more details.     *)
(* Collapse: Takes argument L, and returns a list consisting of L    *)
(*           elements, where a value 1 indicates a normal pulse, and *)
(*           values 2, 3, 4, indicate pulses collapsing (2 is the    *)
(*           start and 4 is the end of collapsing). Probability of   *)
(*           pulses collapsing is assumed to be .1, and the duration *)
(*           of pulses collapsing is uniformly distributed in [1, 10]*)
(*           bins (bins are spaced 3 us apart), not including the    *)
(*           bins corresponding to the rise and fall of the          *)
(*           collapsing event.                                       *)
(* Build: Builds the pulses or continuous rf energy emissions, using *)
(*        the input list. If the current value of list is less than  *)
(*        1, then build-up pulses are generated. If the value is 1,  *)
(*        then continuous pulse trains are generated. If the value is*)
(*        2, 3, or 4, emissions corresponding to pulses collapsing is*)
(*        generated.                                                 *)
(* Npulse: Takes arguments c and bw (in MHz), and calls for procedure*)
(*         pulse. Returns a pulse with peak amplitude c, and time    *)
(*         constant for decay uniformly distributed in [40, 80] ns.  *)
(* Mpulse: Similar to Npulse, but now the peak amplitude is 1,       *)
(*         and the missing pulse phenomenon, with probability .075 *)
(*         is incorporated.                                          *)
(* RiseC: Returns samples that correspond to the rise period of      *)
```

```
(*        pulses collapsing. Similar to rise of Npulse / Mpulse.     *)
(* StayC: Returns samples that correspond to the constant period of  *)
(*        pulses collapsing.                                         *)
(* DropC: Returns samples that correspond to the decay period of     *)
(*        pulses collapsing. Similar to fall of Npulse / Mpulse).    *)
(* (rf energy, with amplitude uniformly distributed in [.1,.2] is    *)
(*  added to all five of Npulse, Mpulse, RiseC, StayC, and DropC).   *)
(*********************************************************************)

MWoven[L_Integer, bw_Integer] :=
 Block[ {ans, buf1, buf2, w, theta,  per, nn, m, s},
  nn    = Random[Integer, {1, 10}];
  buf1  = Table [x/nn , {x, 1, nn}];
  m     = 2767 - (2 nn);
  buf2  = Collapse [m - 200];
  buf2  = Join [ Table [1, {t, 1, 100}], buf2,
  Table[1, {t, 1, (Length [buf2] - m -100)}]];
  buf2  = Join [buf1, buf2, Reverse[buf1], Table [0, {t, 1, 100}]];
  buf2  = RotateLeft[buf2, Random[Integer] Random[Integer, {1, 2766}]];
  s     = Ceiling [ L / (3 bw)];
  ans   = Build[Take[buf2, s], bw];
  ans   = Take[ans, L];
  w     = 2 Pi 8.25 10^6 // N;
  per   = Divide[1, bw 10^6];
  theta = Random[];
  buf3  = Table[Cos[w t per + theta] + I Sin[w t per + theta], {t, 1, L}];
  ans   = ans buf3;
  ans   = ans // N;
  Return[ans]  ]

Collapse[L_Integer] :=
 Block[ {buf},
  buf = {};
  While[Length[buf] < L,
    buf =  Join [buf,
     Join[ {{1}}, {{1}}, {{1}},{{1}}, {{1}}, {{1}},{{1}}, {{1}}, {{1}},
       {Join [{2}, Table[3, {t, 1, Random[Integer, {1, 10}]}], {4}]}]
           [[Random[Integer, {1, 10}]]]];];
  Return[buf] ]
```

```
Build[list_, bw_Integer] :=
 Block[ {res, i},
  res = {};
  i   = 1;
  While [ i <= Length[list],
   If[list[[i]] < 1,  res = Join[res, Npulse[list[[i]], bw]]];
   If[list[[i]] == 1, res = Join[res, Mpulse[bw]]];
   If[list[[i]] == 2, res = Join[res, RiseC[bw]]];
   If[list[[i]] == 3, res = Join[res, StayC[bw]]];
   If[list[[i]] == 4, res = Join[res, DropC[bw]]];
   i++;];
  Return [res] ]

Npulse[c_, bw_Integer] :=
 Block[ {ans, rf, l, rt, tc},
  l   = 3 bw;
  rf  = Table [ Random[Real, {.1, .2}], {t, 1, l}];
  rt  = Max [(1000 / bw), 5];
  tc  = Random[Integer, {40, 80}];
  ans = Pulse[l, bw, rt, tc];
  ans = c ans;
  ans = ans + rf;
  ans = ans // N;
  Return[ans] ]

Mpulse[ bw_Integer] :=
 Block[ {ans, rf, l, m, rt, tc},
  l   = 3 bw;
  rf  = Table [ Random[Real, {.1, .2}], {t, 1, l}];
  rt  = Max [(1000 / bw), 5];
  tc  = Random[Integer, {40, 80}];
  m   = Round [Random[Real, {.425, 1.425}]];
  ans = Pulse[l, bw, rt, tc];
  ans = m ans;
  ans = ans + rf;
  ans = ans // N;
  Return[ans] ]

RiseC[ bw_Integer] :=
 Block[ {ans, rf, buf, l, z, rt},
  l   = 3 bw;
  rf  = Table [ Random[Real, {.1, .2}], {t, 1, l}];
  z   = Divide[1, bw 10^6];
  rt  = rt Max[1000/bw, 5];
  buf = Table[ Divide[t z, rt 10^-9], {t, 0, Ceiling[ rt 10^-9 / z]}];
```

```
  buf = buf / Max[buf];
  ans = Join[buf, Table [1, {t, 1, (l - Length[buf])}]];
  ans = ans + rf;
  ans = ans // N;
  Return[ans] ]

StayC[ bw_Integer] :=
 Block[ {ans, rf, l,},
  l   = 3 bw;
  rf  = Table [ Random[Real, {.1, .2}], {t, 1, l}];
  ans = Table [1, {t, 1,l}];
  ans = ans + rf;
  ans = ans // N;
  Return[ans] ]

DropC[ bw_Integer] :=
 Block[ {ans, rf, l, z, tc},
  l   = 3 bw;
  rf  = Table [ Random[Real, {.1, .2}], {t, 1, l}];
  z   = Divide[1, bw 10^6];
  tc  = Random[Integer, {40, 80}];
  ans = Table[ E^(-t z / (tc 10^-9)), {t, 1, l}];
  ans = ans + rf;
  ans  = ans // N;
  Return[ans]  ]

(************************** PART 5 ***********************************)
(* Interfering Signal Simulated: Photocopier Emissions (PC);         *)
(*                               Type C1.                            *)
(* PROCEDURES IN PART 5:                                             *)
(* PhotoCopier: The main procedure that takes L and bw (in MHz) as   *)
(*              arguments and returns L samples, sampled at bw MHz.  *)
(*              Internally calls for PhotoSpace to obtain the        *)
(*              pulse spacing and PhotoPulse to construct the       *)
(*              pulses.                                              *)
(* PhotoSpace: Takes arguments L, bw (in MHZ) and b (in ns) and      *)
(*             generates a list of exponentially distributed random  *)
(*             variables with mean b ns, sufficient to cover the     *)
(*             duration of observation.                              *)
(* PhotoPulse: Takes L, bw (in MHz), z (number of samples by which   *)
(*             the pulse is supposed to be delayed) and theta (phase *)
(*             offset of the first pulse) as arguments. Risetime     *)
(*             of pulses is fixed by bandwidth, and the fall time    *)
(*             has time constant that is exponentially distributed.  *)
(*             (See text for details). The peak amplitude of the     *)
```

```
(*             pulse is N[1, .36], and the appropriate phase is      *)
(*             added, corresponding to the value z and the center    *)
(*             frequency 2.4175 GHz.                                 *)
(*********************************************************************)

PhotoCopier[L_Integer, bw_Integer] :=
 Block[ {ans, buf1, buf2, theta, i},
  theta = 2 Pi Random[];
  ans   = PhotoPulse [L, bw, 0, theta];
  buf1  = PhotoSpace[L, bw, 220];
  i     = Length[buf1];
  buf2  = Table [ Apply[Plus, Take[buf1, q]], {q, 1, i}];
  While[ i > 0,
   ans   = ans + PhotoPulse[L, bw, buf2[[i]], theta];
   i--;];
  ans   = Take[ans, L];
  ans   = ans // N;
  Return[ans] ]

PhotoSpace[L_Integer, bw_Integer, b_Integer] :=
 Block[ {buf, bb},
  bb  = Round[0.001 (b bw)];
  buf = {};
  While[Apply[Plus, buf] < L,
    buf =  Append [buf,  Expo[bb]];];
  Return[buf] ]


PhotoPulse[L_Integer, bw_Integer, z_Integer, theta_] :=
 Block[ {ans, buf1, buf2, rt, tc, pd, q, phase},
  pd    = Max[80, Expo[143]];
  pd    = pd (30 / bw);
  rt    = 1000 / bw;
  tc    = Ceiling [pd / 5];
  q     = .36 Gauss[1];
  q     = Max[.1, (q + 1)];
  buf1  = Pulse[L, bw, rt, tc];
  phase = theta - (2 Pi 2.4175 z 1000 / bw);
  buf2  = Table[Cos[phase] + I Sin[phase], {t, 1, L}] //N;
  buf1  = buf1 buf2;
  buf1  = Join [ Table[0, {z}], buf1];
  buf1  = Take [buf1, L];
  buf1  = q buf1;
  ans   = buf1 // N;
  Return[ans] ]
```

```
(************************** PART 6 ***********************************)
(* Interfering Signal Simulated: Bandlimited White Gaussian Noise    *)
(*                              (GN); Type C2.                       *)
(* Implemented in a single procedure Noise. Takes L and bw (in MHz)  *)
(* as arguments, and internally calls for procedure Gauss, and       *)
(* generates the in-phase and quadrature components of the noise.    *)
(* (bw is not needed, but included to adhere to convention thus far) *)
(*********************************************************************)

Noise[L_Integer, bw_Integer] :=
 Block[ {ans, buf1, buf2},
  buf1   = Gauss[L];
  buf2   = Gauss[L];
  ans    = buf1 + I buf2;
  Return[ans] ]

(************************** SUPPLEMENT *******************************)
(* The Slow-Varying Envelope for the interfering signals:            *)
(*                                                                   *)
(* PROCEDURES IN THE SUPPLEMENT:                                     *)
(* MWOenv: Returns L samples, sampled at 1200 Hz, for the filtered   *)
(*         (by lowpass filter with cutoff 300 Hz) version of the     *)
(*         slow-varying envelope of the microwave oven (MWO)         *)
(*         emissions (which exhibits the 60 Hz behavior).Accommodates*)
(*         for the fact that observation may begin at any time (hence*)
(*         RotateLeft[buf1, q] command) when interference energy is  *)
(*         emitted.                                                  *)
(* STDenv: Returns L samples, corresponding to s(t) = 1, filtered    *)
(*         by the same ideal lowpass filter as in MWOenv.            *)
(*********************************************************************)


MWOenv[L_Integer] :=
 Block[ {ans, buf1, buf2, q, s},
  s      = Ceiling[L/20];
  q      = Random[Integer, {1, 9}];
  buf1   = {1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0};
  buf1   = RotateLeft[buf1, q];
  buf2   = {1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1};
  buf1   = FFT[buf1];
  buf1   = buf1 buf2;
  buf1   = IFFT[buf1];
  buf1   = Re[buf1];
  ans    = Table[buf1, {s}];
```

```
  ans   = Flatten[ans];
  ans   = Take[ans, L];
  ans   = ans // N;
  Return[ans] ]

STDenv[L_Integer] :=
 Block[ {ans, buf1, buf2, q, s},
  s     = Ceiling[L/20];
  q     = Random[Integer, {1, 9}];
  buf1  = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};
  buf1  = RotateLeft[buf1, q];
  buf2  = {1,1,1,1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1};
  buf1  = FFT[buf1];
  buf1  = buf1 buf2;
  buf1  = IFFT[buf1];
  buf1  = Re[buf1];
  ans   = Table[buf1, {s}];
  ans   = Flatten[ans];
  ans   = Take[ans, L];
  ans   = ans // N;
  Return[ans]  ]

(********************* End of Package ********************************)
```

```
(*********************************************************************)
(*  Package Name: FeatureExtraction.m                                *)
(*                                                                   *)
(*  (c) Ganesh N. Ramaswamy                          November, 1991  *)
(*********************************************************************)
(* PROCEDURES IN THE PACKAGE:                                        *)
(* InstFreq: Computes the instantaneous frequency of the input list. *)
(* CircVar: Computes the circular variance (phase spread) of the     *)
(*          input list, internally correcting for the carrier        *)
(*          frequency offset by calling the procedure InstFreq       *)
(* PSD: Computes the power spectral density of the input list, using *)
(*      the periodogram, and returns a list which contains pairs     *)
(*      (x, y), where x is the frequency (stated as a fraction of    *)
(*      the sampling frequency), and y is the spectral density at x  *)
(* Extract: Takes list1, and list2, where list1 is the slow varying  *)
(*          envelope, and list2 is the complex baseband signal, and  *)
(*          returns the feature vector containing the features       *)
(*          v1 ... v6. Internally calls for procedures InstFreq,     *)
(*          CircVar and PSD.                                         *)
(*********************************************************************)
InstFreq[list_] :=
 Block[ {ans, buf, buf2, buf3, z},
  z    = 2 Pi // N;
  buf  = Arg[list];
  buf2 = Table[ Mod[(buf[[t]] - buf[[t-1]]), z], {t, 2, Length[list]}];
  buf3 = Round[buf2 / z];
  ans  = buf2 - z buf3;
  ans  = ans / z;
  ans  = ans // N;
  Return[ans] ]

CircVar[list_] :=
 Block[ {ans, buf1, buf2, f, l, c, s, r},
  f    = 2 Pi Mean[InstFreq[list]] // N;
  l    = Length[list];
  buf1 = Table[ E^(-f t I), {t,1,l}];
  buf2 = buf1 list;
  buf2 = Arg[buf2];
  c    = Mean[Cos[buf2]];
  s    = Mean[Sin[buf2]];
  r    = Sqrt[ c^2  + s^2];
  ans  = 1 - r;
  Return[ans] ]

PSD[list_] :=
```

```
 Block[ {ans, l, buf1, buf2   },
  l    = Length[list];
  l    = l/2;
  buf1 = FFT[list];
  buf1 = Abs[buf1]^2;
  buf1 = Join[Drop[buf1, 1], Take[buf1, 1]];
  buf2 = Table[t, {t, -l, (l - 1)}];
  buf2 = buf2 / (2 l);
  ans  = {buf2, buf1};
  ans  = Transpose[ans];
  Return[ans] ]

Extract[list1_, list2_] :=
 Block[ {ans, buf1, buf2, buf3, buf4, buf5, buf6, l, v1, v2, v3, v4, v5, v6},
  buf1 = list1 / Sqrt[Mean[list1^2]] //N;
  v1   = Variance[buf1];
  buf2 = Abs[list2];
  buf2 = buf2 / Sqrt[Mean[buf2^2]] // N;
  v2   = Variance[buf2];
  v5   = CircVar[list2];
  buf3 = InstFreq[list2];
  v3   = Mean[buf3];
  v4   = Variance[buf3];
  buf4 = PSD[list2];
  buf4 = Transpose[buf4];
  buf5 = buf4[[2]];
  buf6 = buf4[[1]];
  l    = Apply[Plus, buf5];
  mf   = Apply[Plus, buf5 buf6] / l;
  v6   = Apply[Plus, (buf6 - mf)^2 buf5] / l;
  ans  = {v1, v2, v3, v4, v5, v6} // N;
  Return[ans] ]

(************************* End of Package ****************************)
```

```
(*********************************************************************)
(*  Package Name: SystemSimulation.m                                 *)
(*                                                                   *)
(*  (c) Ganesh N. Ramaswamy                          November, 1991  *)
(*********************************************************************)
(* PROCEDURES IN THE PACKAGE:                                        *)
(* Transmit1: Takes a complex list and a desired SNR, and adds       *)
(*            complex Gaussian noise at the stated SNR to the list.  *)
(* Transmit2: Takes a real list and a desired SNR, and adds real     *)
(*            Gaussian noise at the stated SNR to the list.          *)
(* Digitize1: Takes a complex list, and desired number of bits for   *)
(*            digitization, and returns a digitized version of the   *)
(*            input list.                                            *)
(* Digitize2: Takes a real list, and desired number of bits for      *)
(*            digitization, and returns a digitized version of the   *)
(*            input list.                                            *)
(* Fjitter: Takes arguments list, bw (in MHz), and range (in MHz)    *)
(*          and returns a list that is distorted by frequency jitter *)
(*          of the local oscillator that is uniformly distributed in *)
(*          [-range MHz, range MHz], (range need not be an integer). *)
(* Pjitter: Takes arguments list and range (in degrees), and returns *)
(*          a list that is distorted by phase jitter (on two         *)
(*          quadrature channels), that is uniformly distributed      *)
(*          in [-range (degrees), range (degrees)].                  *)
(* Run: Takes arguments n, L, bw (in MHz), and snr (in dB), and      *)
(*      generates list bufA1 ... bufC2, each containing n feature    *)
(*      vectors, corresponding to the 6 interfering signals.         *)
(*      Internally calls for the procedures that simulate the        *)
(*      interfering signals, using parameters L and bw, and adds     *)
(*      noise at the stated snr, and extracts the features.          *)
(*      Optional procedures Digitize, Fjitter and Pjitter may be     *)
(*      included (See note before the beginning of the procedure).   *)
(* Learn: Uses the global variables bufA1 ... bufC2 generated by Run,*)
(*        and estimates the mean vectors mA1 ... mC2, and covariance *)
(*        matrices cA1 ... cC2, corresponding to the 6 interfering   *)
(*        signals. The mean vectors and covariance matrices are also *)
(*        global variables.                                          *)
(* Diagnose: Takes a list of feature vectors and computes the        *)
(*           likelihood values for each interfering signal. Uses     *)
(*           global variables generated by Learn.                    *)
(*********************************************************************)

Transmit1[list_, snr_] :=
 Block[ {ans, buf, a, b1, b2, c1, c2, l},
```

```
  l    = Length[list];
  a    = 10^(snr / 20);
  b1   = Sqrt[Mean[(Re[list])^2]];
  b2   = Sqrt[Mean[(Im[list])^2]];
  c1   = b1 / a;
  c2   = b2 / a;
  buf  = c1 Gauss[l] +  I c2 Gauss[l];
  ans  = list + buf // N;
  Return[ans] ]

Transmit2[list_, snr_] :=
 Block[ {ans, buf, a, b, c, l},
  l    = Length[list];
  a    = 10^(snr / 20);
  b    = Sqrt[Mean[list^2]];
  c    = b / a;
  buf  = c Gauss[l];
  ans  = list + buf // N;
  Return[ans] ]

Digitize2[list_, bit_]:=
 Block [{buf1, buf2},
  buf1 = Re[list] / Max[Abs[Re[list]]];
  buf2 = Im[list] / Max[Abs[Im[list]]];
  buf1 = 2^(bit - 1) buf1;
  buf2 = 2^(bit - 1) buf2;
  buf1 = Round[buf1] + 0.1;  (* Note: + 0.1 is done to avoid the     *)
  buf2 = Round[buf2] + 0.1:  (* possible 0/0 problem while computing *)
  ans  = buf1 + I buf2;      (* the argument during feature          *)
  Return[ans] ]              (* extraction; the error is negligible  *)

Digitize2[list_, bit_]:=
 Block [{buf},
  buf = list / Max[Abs[list]];
  buf = 2^(bit - 1) buf;
  buf = Round[buf];
  buf = buf + 0.1;           (* See note above *)
  Return[buf] ]

Fjitter[list_, bw_Integer, range_]:=
 Block [{ans, buf, l, per},
  l = Length[list];
  buf = Table[ Random[Real, {-range, range}], {l}];
  per = Divide[1, bw 10^6];
  buf = 2 Pi buf //N;
```

```
  buf = Table[Cos[buf[[t]] t per] + I Sin[buf[[t]] t per ], {t, 1, l}];
  ans = list buf;
  Return[ans] ]

Pjitter[list_,  range_]:=
 Block [{ans, buf1, buf2, l, per},
  l = Length[list];
  buf1 = Table[ Random[Real, {-range, range}], {l}];
  buf2 = Table[ Random[Real, {-range, range}], {l}];
  per = Divide[1, bw 10^6];
  buf1 = (Pi / 180) buf1 //N;
  buf2 = (Pi / 180) buf2 //N;
  buf = Table[ Cos[buf1[[t]]] + I Sin[buf2[[t]]], {t, 1, l}];
  ans = list buf;
  Return[ans] ]

(************ NOTES FOR PROCEDURES Run, Learn and Diagnose  **********)
(* (1)  bufA1... bufC2, mA1 ... mC2, cA1 ..cC2, are GLOBAL variables *)
(* (2)  Internal calls for Digitize, Fjitter and Pjitter may be made *)
(*      after each of the Transmit commands in the procedure Run,    *)
(*      to introduce non-ideal system behavior.                      *)
(*********************************************************************)
Run[n_Integer, L_Integer, bw_Integer, snr_] :=
 Block[ {i, list1, list2, list3, list4 },
  i      = n;
  bufA1  = {};
  While [i > 0,
    list1  = EASD[L, bw];
    list1  = Transmit1[list1, snr];
    list2  = STDenv[20];
    list2  = Transmit2[list2, snr];
    list3  = Extract[list2, list1];
    bufA1  = Append[bufA1, list3];
    i--;];
  i      = n;
  bufA2  = {};
  While [i > 0,
    list1  = FM[L, bw];
    list1  = Transmit1[list1, snr];
    list2  = STDenv[20];
    list2  = Transmit2[list2, snr];
    list3  = Extract[list2, list1];
    bufA2  = Append[bufA2, list3];
    i--;];
  i      = n;
```

```
  bufB1  = {};
  While [i > 0,
    list1  = SSBfh[L, bw];
    list1  = Transmit1[list1, snr];
    list2  = STDenv[20];
    list2  = Transmit2[list2, snr];
    list3  = Extract[list2, list1];
    bufB1  = Append[bufB1, list3];
    i--;]
  i      = n;
  bufB2  = {};
  While [i > 0,
    list1  = MWoven[L, bw];
    list1  = Transmit1[list1, snr];
    list2  = MWOenv[20];
    list2  = Transmit2[list2, snr];
    list3  = Extract[list2, list1];
    bufB2  = Append[bufB2, list3];
    i--;];
  i      = n;
  bufC1  = {};
  While [i > 0,
    list1  = PhotoCopier[L, bw];
    list1  = Transmit1[list1, snr];
    list2  = STDenv[20];
    list2  = Transmit2[list2, snr];
    list3  = Extract[list2, list1];
    bufC1  = Append[bufC1, list3];
    i--;];
  i      = n;
  bufC2  = {};
  While [i > 0,
    list1  = Noise[L, bw];
    list1  = Transmit1[list1, snr];
    list2  = STDenv[20];
    list2  = Transmit2[list2, snr];
    list3  = Extract[list2, list1];
    bufC2  = Append[bufC2, list3];
    i--;];
  ]

Learn:=
 Block[ { listA1, listA2, listB1, listB2, listC1, listC2 },
  mA1    = Mean[bufA1];
  listA1 = Table[ Outer[Times, bufA1[[t]] - mA1, bufA1[[t]] - mA1],
```

```
            {t, 1, Length[bufA1]}];
  cA1    = Mean[listA1];
  mA2    = Mean[bufA2];
  listA2 = Table[ Outer[Times, bufA2[[t]] - mA2, bufA2[[t]] - mA2],
            {t, 1, Length[bufA2]}];
  cA2    = Mean[listA2];
  mB1    = Mean[bufB1];
  listB1 = Table[ Outer[Times, bufB1[[t]] - mB1, bufB1[[t]] - mB1],
            {t, 1, Length[bufB1]}];
  cB1    = Mean[listB1];
  mB2    = Mean[bufB2];
  listB2 = Table[ Outer[Times, bufB2[[t]] - mB2, bufB2[[t]] - mB2],
            {t, 1, Length[bufB2]}];
  cB2    = Mean[listB2];
  mC1    = Mean[bufC1];
  listC1 = Table[ Outer[Times, bufC1[[t]] - mC1 bufC1[[t]] - mC1],
            {t, 1, Length[bufC1]}];
  cC1    = Mean[listC1];
  mC2    = Mean[bufC2];
  listC2 = Table[ Outer[Times, bufC2[[t]] - mC2, bufC2[[t]] - mC2],
            {t, 1, Length[bufC2]}];
  cC2    = Mean[listC2];
 ]


Diagnose[list_] :=
 Block[ {i, a1, a2, b1, b2, c1, c2, dA1, pA1, dA2, pA2, dB1, pB1, dB2,
         pB2, dC1, pC1, dC2, pC2, res },
  i    = Length[list];
  ans  = {};
  a1   = ((2 Pi)^-3)/ Sqrt[Det[cA1]];
  a2   = ((2 Pi)^-3)/ Sqrt[Det[cA2]];
  b1   = ((2 Pi)^-3)/ Sqrt[Det[cB1]];
  b2   = ((2 Pi)^-3)/ Sqrt[Det[cB2]];
  c1   = ((2 Pi)^-3)/ Sqrt[Det[cC1]];
  c2   = ((2 Pi)^-3)/ Sqrt[Det[cC2]];
  While[i > 0,
   dA1    = -.5 (list[[i]] - mA1). Inverse[cA1]. (list[[i]] - mA1);
   pA1    = aa (E^dA);
   dA2    = -.5 (list[[i]] - mA2). Inverse[cA2]. (list[[i]] - mA2);
   pA2    = bb (E^dB)
   dB1    = -.5 (list[[i]] - mB1). Inverse[cB1]. (list[[i]] - mB1);
   pB1    = cc (E^dC);
   dB2    = -.5 (list[[i]] - mB2). Inverse[cB2]. (list[[i]] - mB2);
   pB2    = dd (E^dD);
```

```
   dC1    = -.5 (list[[i]] - mC1). Inverse[cC1]. (list[[i]] - mC1);
   pC1    = ee (E^dE);
   dC2    = -.5 (list[[i]] - mC2). Inverse[cC2]. (list[[i]] - mC2);
   pC2    = ff (E^dF);
   res    = {pA1, pA2, pB1, pB2, pC1, pC2};
   res    = res / Apply[Plus, res] //N ;
   ans    = Append[ans, res];
   i--;];
  ans    = .01 Round[100 ans];
  Return[ans] ]

(************************ End of Package ****************************)
```