Performance Evaluation of Cognitive Radios

Nicholas J. Kaminski

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

Masters of Science
in
Electrical Engineering

Charles W. Bostian, Chair
Allen B. MacKenzie
Tamal Bose

April 20, 2012
Blacksburg, Virginia

Keywords: Cognitive Radio, Cognitive Engine, Performance Evaluation

Performance Evaluation of Cognitive Radios

Nicholas J. Kaminski

(ABSTRACT)

This thesis presents a performance evaluation system for cognitive radio. It considers performance as a complex, multi-dimensional function. Typically such a function would take some record of actions as an argument; however, a key contribution of this work is the addition of background information to the domain of the performance function. Including this information generalizes the performance function across many radios and applications, with the additional cost of complicating the domain. Thus the presented evaluation system organizes the domain information into sets. These sets are divided into two categories, one capturing necessary information that is external to the radio and on capturing necessary information that internal to the radio. These categories highlight the fact that neither the true actions nor the true performance is directly observable at the onset of evaluation. This arises because a cognitive radio can only express its actions in terms of the available knobs and meters, which together form the radio's language. Some understanding of this language and its limitations is required to fully understand the radio's expression of its actions. This parallelism of actions and performance suggests implementing the evaluation method as a composite form of the performance function. The composite performance function is made up of two sub-functions, one of which producing action information and one of which producing performance information. Specifically, the first sub-function is used to determine general measures of the actions' influence on performance; these are labeled Measures of Effectiveness. The second sub-function uses these Measures of Effectiveness to determine application specific performance values, called Measures of Performance. This work covers both these measures in detail. Each measure is determined as the result of a neural network based interpolation. This thesis also provides an examination of artificial neural networks in the scope of performance evaluation. Once these concepts are explored, a walk-through evaluation is presented. The four phases are the Setup Phase, the Logging Phase, the Training Phase, and the Evaluation Phase. Each phase is structured to provide the information necessary to determine the final performance. These phases detail the process of evaluation and discuss the realization of concepts explored earlier. This work concludes with a comparative evaluation example that proves the worth of the presented approach. A full evaluation system is outlined by this thesis and the foundational details for the system are explored in detail.

Grant Information

This work was supported by a Bradley Fellowship through the Bradley Department of Electrical and Computer Engineering.

Dedication

To my family, most especially my wife, Jen, my parents, Arther and Joyce, and my grandfather, James

Acknowledgments

I would like to thank my family for their continued support as I pursue my dreams. I would like to especially thank them for understanding as I focus on school. I would like to especially thank my wife for reading through many revisions of this document and putting up with me as I wrote them.

I would like to thank Dr. Bostian for his support and help throughout my time at Tech. I would like to especially thank him for pushing me to get my ideas on paper and asking the tough questions.

I would like to thank my other committee members (Dr. MacKenzie and Dr. Bose) for offering patient guidance and understanding.

I would like to thank my remaining lab mates (Alex and Al) for giving me ideas and making me think differently.

All figures in this thesis were created by the author.

Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Network
AWGN	Additive White Gaussian Noise
BER	bit error rate
CE	cognitive engine
CR	cognitive radio
CSERE	Cognitive System Enabling Radio Evolution
CWT	Center for Wireless Telecommunications
DSA	Dynamic Spectrum Access
FEC	Forward Error Correction
MoE	Measure of Effectiveness
MoP	Measure of Performance
NNI	Neural Network Interpolator
REM	Radio Environment Map
RF	Radio Frequency
SDR	software defined radio

Chapter 1

Overview

Mitola proposed cognitive radio (CR) as an artificially intelligent extension to software defined radio (SDR) [1]. This marriage between artificial intelligence and flexible radio platforms opens a great number of possibilities. Dynamic Spectrum Access (DSA) represents an exciting and most often cited example of the possibilities offered by CR. However, this example is merely the one that has received the most attention to date. Mitola's introduction of CR opens a wide breadth of directions for the future of radio.

The field of CR is, in many ways, still new, and largely unexplored. There is much foundational work in this field that is left to be done. CR represents a new design paradigm for radios that benefits from specialized treatment and consideration. The union between radio and artificial intelligence forms an extremely complex system with many facets of detail. Thus, advancing in this field is a difficult task, based on melding two divergent fields of research.

A good example of foundational work that eases the difficulties of the field is the cognitive engine (CE) [2]. This concept provides a fundamental approach to building CR based on the abstraction of the radio and artificial intelligence domains. This tool allows existing knowledge to be appropriately utilized to accelerate CR development. Thus, the CE represents a foundational contribution to the field of CR by providing a method to separate the cognitive and radio elements of a CR.

The work presented here builds toward another such foundational contribution. Specifically, this work presents the basis of performance evaluation for cognitive radios. A performance evaluation system brings several benefits to the field of cognitive radio, but chief among them are the ability to compare CRs quantitatively and insight into the potential of a CR. The work presents the framework of a performance evaluation system that will deliver these key benefits to the field of cognitive radio.

1.1 Performance Evaluation Guiding Principles

Measuring CR performance is a complex and multi-layered problem that is further complicated by the inherently subjective basis of performance. Naturally the standardization of the subjective qualities of performance must be a primary goal of any evaluation system. Standardization of subjective elements provides necessary consistency which enables the comparison of CRs. Thus, it is clear that transitioning from a subjective understanding of performance to a objective measure in a key sub-problem of performance evaluation.

Guiding this transition are a number of axioms. First and foremost a measurement of performance is non-sensical if not directly tied to a specific application or mission. This stems directly from the subjective nature of performance; a concept as broad as performance requires context to provide clarity. Second, as a corollary of the first axiom, there is no absolute scale for performance. Third, measurements must be determined consistently. Without consistency of approach and measurement, the subjectivity of performance has not truly been standardized and nothing is gained. Finally, the reference must be derived by carefully examining the situation of evaluation. This axiom, again, stems from the the first axiom. An understanding of the situation of evaluation, including the agent being evaluated, completes the understanding of application or mission. These properties provide guidance for the development of an evaluation framework.

There are two key notions stressed by the above axioms: consistency and situational understanding. Providing either of these notions requires a solid organization of information. From the axioms stated above, any value of performance that has clear meaning, is tied to the situation in which evaluation takes place. This means that understanding the specifics of the situation of evaluation leads to a fuller understanding of the performance. Thus, cataloging, organizing, and applying situation information forms a major part of the evaluation process.

This discussion has covered the general guiding principles of performance evaluation. The remainder of this work builds on these principles to provide a CR performance evaluation system. This performance evaluation system will provide the basis of foundational contribution to the field of cognitive radio.

1.2 Current Approach

In reviewing the literature, there are surprisingly few examples of performance evaluation methods for cognitive radio. Additionally, the “relatively limited” [3] attention this important topic has received is largely limited to the single application of DSA. While DSA certainly represents an exciting CR application, the field needs a more general approach to performance evaluation that is consistent across applications. This is, in fact, the point at which the currently available methodologies fail. For example, the authors of [3] present a well developed report card based system with the power to consider performance at both the nodal and network levels. This system also includes a range of metrics to handle various goals. However, the system presented is aimed only at the evaluation of DSA CRs. Such a narrow aim allows this evaluation system to be extremely detailed, however, the cost of this level of detail is an inability to extend to other applications. Thus, the method cannot consistently compare CRs for various applications. This comparison problem is fairly typically among evaluation systems for DSA CRs, which represent the majority of available evaluation systems.

A different approach to evaluation, presented in [4], is meta-cognition, by which a CR observes its own performance and changes high level strategies to increase this performance more rapidly. Naturally this process requires an evaluation of the CR performance. In this case, the knobs and meters already used for decision making within the radio provide a characterization of the radio's performance. This method is well suited for meta-cognition, as it reduces the overhead of performance evaluation. However, this method suffers from the dual of the problem to the method discussed above. While the meta-cognition approach extends well across various missions of a cognitive radio, it is tied to a single CR implementation. Thus, meta-cognition evaluation lacks the necessary consistency across various radios, again crippling the comparative power of this method.

The two methods discussed above highlight the common problems with available methods. Currently available evaluation systems largely lack the consistency necessary for a wide range of comparison. Providing such comparative abilities is certainly non-trivial, but there is a clear need in the field for an evaluation system that provides both wide comparative ability and wide applicability.

1.3 Proposed Approach and Organization

Based on the above discussion, there is a clear need for a widely applicable and widely consistent evaluation system for general purpose CRs. Achieving this will require solid organization and utilization of information related to the situation of evaluation. Therefore, this work will provide the foundation of the needed system through proper application of available information.

The presentation of this foundational work begins with an in-depth introduction to performance evaluation, in Chapter 2. This introduction examines the functional form of CR performance, expanding the domain to generalize the performance function. However, the consequence of this generalization is additional complexity in the domain of the performance function. Thus, the majority of Chapter 2 is devoted to discussing a solid organization for this domain and the implications of this approach to the system evaluation. The chapter concludes with a discussion of the relation of the actions to both the information captured in the domain and the overall performance of the CR.

Chapter 3 refines the performance function discussion of Chapter 2 in order to work towards an implementation scheme. The limitations of the work presented here set the stage for a discussion of this implementation scheme. Examination of the implementation approach begins with a discussion of the conceptual framework used for implementation of the evaluation system. This discussion presents concepts that form the heart of this evaluation system. Once these concepts have been explored, a more concrete structure is covered. An examination of the enabling technology necessary for this concrete structure rounds out the chapter.

A proof of concept evaluation is then examined in Chapter 4. This proof of concept first walks through the four phases of the evaluation process. Each phase is examined in terms of what the phase accomplishes and how the information discussed in the previous chapters is related. This chapter provides the details of how the ideas of Chapter 2 and the implementation approach of Chapter 3 come together. This chapter concludes with a discussion of two example evaluations that display the power of this approach.

This work concludes in Chapter 5. This chapter summarizes the evaluation system presented here. Conclusions are drawn from the work presented before accomplishments are reviewed. This chapter, and this work, ends with an examination of future methods of extending this evaluation system.

Chapter 2

Introduction to Performance Evaluation

Performance evaluation of any non-deterministic system is an inherently complicated task. Evaluation requires a great deal of information, with some of it encoded in various ways. There are two major types of information, in fact, that may be simply referred to as background information and foreground information. The foreground information relates the commonly conceived information required for performance evaluation, namely the actions of the agent under test and the final performance score. This foreground information is encoded by the situation at hand. The background information provides the information necessary to understand this encoding. Organizing the background information into a collection of independent sets provides a strong framework for general performance evaluation. Viewing the encodings of the foreground information as languages, which must be learned and understood, provides powerful techniques. This chapter is devoted to exploring the application of these concepts to the performance evaluation of CR. It is worth noting that while the discussion here is focused on the performance evaluation of CR, these concepts certainly extend beyond CR applications.

2.1 Domain for Evaluation

The performance of a cognitive radio is most simply described as a function that maps from some n-dimensional domain to the real number line in a consistent manner. The goal of a performance evaluation system is then easily stated as providing as full a characterization of this complex multidimensional function as possible. However, the price that is paid for this simple description is a great deal of subtle complexity hidden within this n-dimensional domain. Understanding this domain provides the means to understand better the unknown mapping that provides quantitative values for performance. For example, considering such functions over a subset of the domain offers a realizable means of probing functional behavior by removing some of the complexity of the function. Such decomposition is often most valuable when considering key dimensions (down to the case of a single dimension) of the domain, which certainly requires a solid understanding of the domain itself. Note that while decomposition methods are based on a number of simplifying assumptions, these methods do reveal much about otherwise inaccessible functions when properly applied. Therefore, providing a clear framework for the domain of evaluation is a key step in constructing a performance evaluation system.

Given such a framework, this decomposition based method is powerful enough to use in isolation. Indeed when faced with a problem as complicated as evaluating the performance of a general purpose cognitive radio, such a decomposition method may provide the only way forward. Additionally, these methods form the foundation of more complicated (and perhaps more ideal) techniques. Finally, this approach to the performance evaluation of CR is novel. For these reasons this work will focus on this decomposition method, evaluating performance over a subset of the entire domain.

The determination of a useful subset of the domain for evaluation requires a solid framework that allows for flexible consideration of any number of the factors that make up the evaluation problem. Defining the domain for the evaluation of performance as the intersection of independent sets provides this ability in a straight forward framework. [1] Each set is used to collect background knowledge into a unit that defines some aspect of the topology of the domain for evaluation. By construction, these sets are completely independent from one another, as such self-contained sets allow this framework to consider any number of the individual factors of performance. Additionally, this framework provides additional metrics to probe the performance function, such as the distance between concepts within a set. While such a distance is somewhat arbitrary, it does provide a relative understanding of the factors. The benefits of this relative understanding are further explored below. Such additional benefits make this framework a strong foundation for the consideration of CR performance.

Each set captures a different factor of the evaluation situation, capturing a defines a complete aspect of performance. To determine these aspects, when conducting an evaluation of CR performance, several questions provide the necessary background information that describes the situation. Each of the questions defines an independent manner in which performance can be considered, or, loosely, a dimension of the topology over which the performance function is defined. Examples of such questions include what is the radio trying to do, what can the radio express, and what should the radio express. All of the possible answers to one of these questions defines a set for the evaluation of a cognitive radio. Each question and its corresponding set are independent of all the other pairs of questions and each defines a different aspect of the situation. Each answer to this set of questions provides the value for one dimension of the specific evaluation problem.

---

[1] Thanks to Dr. William Floyd, Mathematics Professor at Virginia Tech, for helping to clarify vocabulary for this discussion

Adding this background information into the domain of the performance is unusual because the performance of a cognitive radio is typically discussed as a function of the actions of the radio. In the typical approach, the information captured by these sets is either entered as parameters to the performance function or else shapes the selection of the function itself. Thus, typically evaluation approaches are well suited only to the single case described by the function used. Considering the background information as part of the domain of the function itself helps to generalize the approach. Evaluating the performance of CRs for a multitude of possible missions requires this unusual generalization.

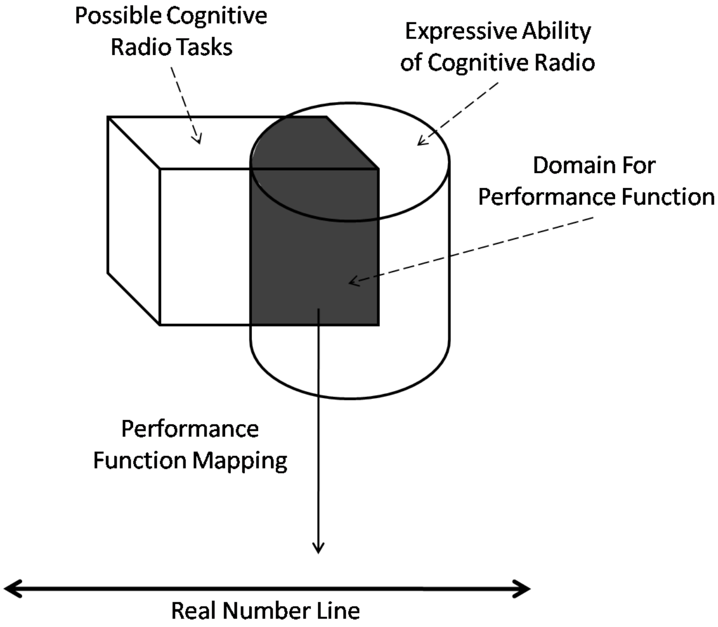

Figure 2.1: Performance Function Visualization

Figure 2.1 provides a visualization of how the various sets and the true CR performance fit together. In this example, the performance is measured over all the possible missions some CR has the ability to express. This region is expressed as the intersection of the concepts that the CR under test has the power to express and the collection of all possible missions. The performance function then provides the mapping from this intersection to the real number line. Note this figure simply displays the role of the background information in this process to provide a straight forward introduce to the process. For a complete view, the actions of the radio are also necessary for this mapping; however, the inclusion of the actions clouds the concepts related here. With both the background information and the actions in its domain, the performance function is of high dimension and difficult to visualize accurately. The figure provided here is therefore intended simply as an example of the organization of background information discussed above.

2.2 Cognitive Radio Domain

The next step for the evaluation of CR performance is the determination of the specific sets that make up the domain of the CR performance function. Above several questions were used to discuss the set-based framework for the domain of the performance function. These questions, in fact, provide a nearly complete picture of the sets necessary to form an evaluation domain. In fact, only the single question of where is the radio operating is required to complete the evaluation picture. That is, some understanding of the nature of the environment is required for full performance evaluation. In short, answering the question of how well did the radio perform requires knowledge of what the radio is trying to do, where the radio is trying to do it, what the radio can understand, and what the radio should understand.

Understanding the implication of how each of these questions is answered is key to the evaluation of CR performance. The context of the situation in which performance is evaluated can greatly affect the final outcome of performance. This well known principle is evident in all performance evaluation, but perhaps best exemplified in competitive marathon running, where some tracks are known to be faster than others. Both the answer to the question and the manner in which the answer is expressed shape the understanding of this context, and thus the manner in which performance is evaluated. Ideally the contribution of each set to this context for evaluation would be directly evident and comparable to the contribution from any other set. However, in practice this is not the case and understanding the scale of influence over the situation for evaluation provided by each set is a necessary step is achieving in performance evaluation. A specific situation, influenced by a specific realization of sets, has the effect of coloring the results of any testing, which can make the comparison of results very difficult. Understanding the influence of each set over the situation is certainly necessary for a complete approach to performance evaluation.

In order to better understand each answer set, the background information provided by the above questions is readily divided into the two categories of purpose and language. Both categories capture a general aspect of the situation for which performance is evaluated. These categories and their member sets are discussed below.

2.2.1 Purpose

The category of purpose information contains all of the necessary background information for evaluation that exists externally to the cognitive radio. This includes, primarily, the mission of the CR and the environment of the CR. The mission is perhaps the most important set of the domain for evaluation because this set captures the information which defines what goodness is. The environment information, on the other hand, has a weaker effect on the performance in that it helps to shape the context of a performance score. Each of these sets is discussed further below.

Mission

Mission is a somewhat special set. This set is the most important to performance evaluation, as performance evaluation quickly loses meaning if this set is not considered. Additionally, the specifics of this set can, in practice, affect how additional information is applied to evaluation. Thus any performance analysis must consider at least this region of the n-dimensional domain of the performance function.

As such necessary information, mission must be carefully considered. When considering the set of mission information for performance evaluation it is helpful to think of this set as having volume (height, width, and depth); that is to say it is helpful to visualize the mission set as a cube. There are several different levels of mission abstraction that can be considered. Higher levels of abstraction are, by necessity, combinations of lower levels. For example, an engineer working in a lab often considers low level missions such as achieving a certain bit error rate (BER) or throughput threshold. A fire chief in the field, on the other hand, is more concerned about extinguish a building fire safely. From the fire chief's point of view, the mission may be personal safety or speed while extinguishing the fire, which both feel a large impact from the use of radios. In the high level missions for the fire chief, a good radio will have to achieve some combination of BER and throughput levels as well as some other high level tasks such as the prioritization of data. Thus the very high level missions of the fire chief shape the goodness of a radio by considering numerous lower level missions simultaneously. Any level of missions shapes whether a radio performs well or not and thus all levels must be captured within the set of mission information.

Organizing all missions into layers of abstraction helps achieve a flexible system for performance evaluation in various situations. Note that the entire mission set can be thought of as a collection of subsets, each containing all of the possible missions at the same level of abstraction. Visually this organization is well represented by a cube, see Figure 2.2. Note, however, that the mission subsets are typically not all of the same dimension. Organization of missions in this way allows engineers to focus on performance over one subset that suits their interests and fire chiefs to focus on a different subsets. Combining these subsets at particular layers of abstraction provides the mission set with a range of granularity, enabling this set to capture the focus of various evaluators. The self-contained definition of the mission set also provides consistent consideration of any given mission; missions can be examined as a whole or as individual submissions. Effectively this method of organizing mission information helps to enable a decomposition based approach to performance evaluation.

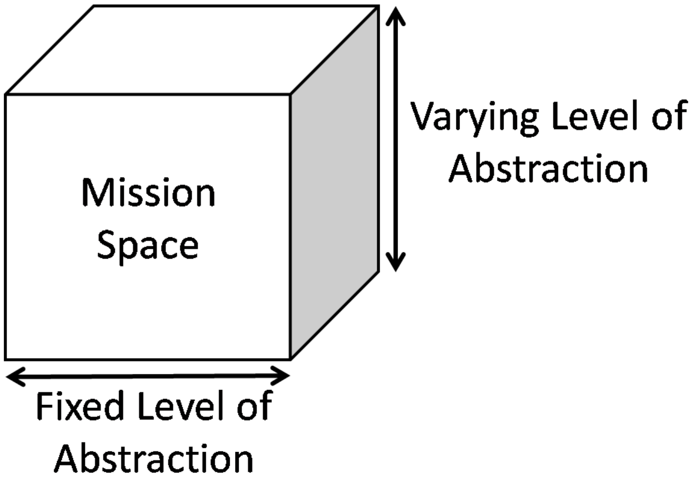

Figure 2.2: Organization of Mission Abstraction Level

Note that the definition of mission information in this way effectively means that the mission set is a convex volume (given proper scaling of combination coefficients). Such convexity generally provides many properties that are useful in searching such volumes. However, because the domain is constructed as the intersection of several sets, the entire domain is generally not convex and thus many of these properties do not often have a large impact on performance evaluation. Considering the set of mission information as a convex volume simply provides additional insight into the organization of this set.

The concept of distance, enabled by consideration of mission information as a set, provides several useful abilities when considering missions. For example, note that a subset of mission information containing missions at a specific level of abstraction contains both the category of missions with the goal of achieving some BER and the category of missions with the goal of achieving some throughput.The distance between these two categories of mission (BER threshold and throughput threshold), provides a relative difference between these two mission types. This difference is advantageous for considering the application of a CR beyond a tested mission, as it enables estimation of performance for missions close to their design target. Naturally, this use of distance requires consistent definition of the position of specific missions within a set and examination of appropriate measure. Note that Euclidean distance may not provide meaningful results, depending on the definition of mission position, and some other measure may be required. The concept of relative distance between missions is an example of a side benefit of the intersecting set framework.

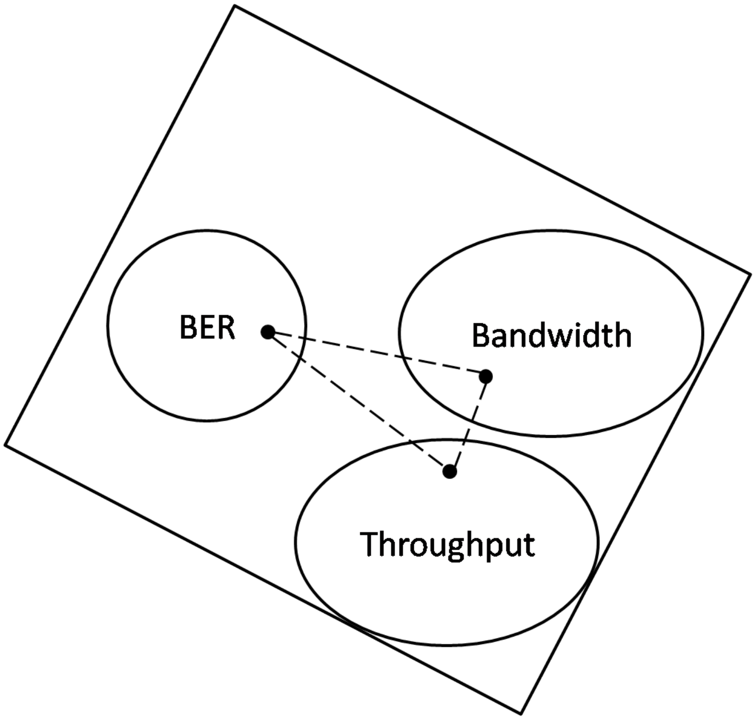

Figure 2.3: Mission Distance Example

Figure 2.3 provides a simplified example of the concept of relative mission distance. The plane shown in Figure 2.3 represents a subset of missions at the same level of abstraction, here the consideration of a target RF parameters. The regions within the plane show the areas in which a particular parameter is regarded as the most important, i.e. the labeled parameter would receive the heaviest weight during optimization. Three example missions are represented by points within these regions. The lines connecting these missions show the distance between each pair of missions. Here the distance between bandwidth focused missions and throughput focused missions is less than that between BER focused missions and throughput focused missions. This relationship provides a quantitative method of saying that achieving a target throughput is more similar to achieving a target bandwidth than to achieving a target BER. Note that this example is simplified to highlight the underlining concepts of relative mission distance. Sets must certainly be logically constructed for relative mission distance to have meaning. Additionally note that the Figure 2.3 need not show a Euclidean space; rather, some other method for determining distance may have more meaning.

Mission distance does have several useful applications, but they are beyond the scope of this work. The concept is introduced here to highlight a benefit of set based organization for background information. Utilizing this concept in a meaningful way requires some foundational work with this set based organization, such as the performance evaluation system presented here, as well as careful construction of mission sets in potentially non-Euclidean space. Thus, the concept is merely raised here as a future method, which will bring additional ability to the performance evaluation system presented.

Environment

The environment provides necessary context to performance evaluation. Knowledge of the environment often directly reveals the reason for the actions of a CR, as the environment captures the various external stimuli that inform the cognitive radio's behavior. In this way, the environment often determines the variety of approaches exhibited by a CR during some situation. For example, a rather difficult environment (fast-fading, multipath, many interferers) may force some radios to exhibit a broad range of skills to handle the environment. Alternatively, a difficult environment may simply overcome a radio that performs very well in a less extreme environment. Thus, the level of difficulty presented by an environment can greatly impact the behavior, and therefore the performance, of CRs in various ways. Understanding this impact of environment on radio performance is key to comparing performance across situations.

As the environment captures the external stimuli for a CR, the environment is most clearly expressed based on the perspective of the radio under test. For example, if the radio under test only understands RF information, the environment should only be expressed in terms of this information. As the cognition of the radio grows to include more information, perhaps the physical location of the radio, these factors become directly important for evaluation as well. Note, however, that the environment in both of these cases does not actually change, the expression of the environment simply emphasized different attributes. Expressing the environment in terms of the agent's understanding most clearly highlights the challenges faced by the cognitive element of the radio.

Capturing environment information in a quantitative manner is perhaps the best studied aspect of performance evaluation. There are, in fact, several techniques available to capture this information, but Radio Environment Maps (REMs) provide, perhaps, the most promising option. REMs developed from the application of Available Resource Maps to Wide Area Networks [5]. Radio Environment Maps provide a method to manage radio data across a network. This information includes propagation environment, policy, prior performance, and node limitations [6]. Since their introduction, REMs have been well studied for application to CR networks, [7] and [8]. A detailed example of the application of REMs to IEEE 802.22 WRAN CRs is given in [9]. While REMs have been envisioned for use in cognitive networks, the principles of this technique could certainly be used to capture environmental data for performance evaluation.

Knowledge of the environment provides the context necessary for comparison across situations. Capturing this knowledge is a well studied problem, as environmental data are often beneficial to radio systems. Once captured this environmental data augments the mission data used in evaluation. Note that environmental information does not directly define the mission or the meaning of good performance, but simply provides additional context. While this additional context certainly provides a more full view of performance and allows for more straight forward comparisons across situations, it is not strictly necessary for performance evaluation. For these reasons and to focus on providing a core foundation for performance evaluation of cognitive radios, this work does not consider the set of environment information further.

2.2.2 Language

The category of language information contains all of the necessary background information for evaluation that exists internally to the cognitive radio. On the face of it, this is purely the knobs and meters provided by the cognitive element of the radio. However, for the purposes of performance evaluation, which knobs and meters the CR uses are as nearly as important as the values of the knobs and meters. In short, CR evaluation must consider both what the cognitive radio says and the language it uses to say it.

Performance evaluation can be conceptually separated into two steps. The first of these steps is determining if the CR took a good action based on what it knew at the time. The second step can then consider whether the information available to cognitive radio was proper for making the decision at hand. The two sets discussed here capture the information that is necessary for these two steps.

Note that the language aspects of a CR are certainly complex and deeply impact its performance. For these reasons it is helpful to first consider some concepts from the study of human language. Human thought provides the only truly observable form of intelligence and as such provides the only truly stable foundation for the consideration of any aspect of intelligence. As language is effectively the means by which intelligence is communicated (both internally and externally), this result applies to the consideration of language. [2]

2.2.3 Human Language

Modern linguistics largely agrees that human language is based on a highly parameterized structure, typically referred to as the Universal Grammar [10]. This structure describes every human language system in use and is based on the setting of various binary parameters. These parameters describe how phrases and mental language lexical items fit together. Mental language lexical items are the smallest units of self-contained meaning, and phrases are collections of these lexical items. Other parameters are then used to determine the pronunciation of these units of language.

---

[2] Thanks to Dr. Joe Eska, English Department Chair at Virginia Tech, for providing background and guidance with this discussion of cognitive language

This language structure provides the the manner in which some pure thought or truth is encoded into a communicable form. Figure 2.4 shows the progress of this encoding in block diagram form. In this figure the elements of the encoding process are organized into three tiers. The pure truth and Universal Grammar both occupy the top tier as Platonic ideals; both of these elements are the same for everyone. This top tier represents the foundation of the encoding. The next tier is the realization of language in a particular person. This realization is specific to one person and built up over time through experience. Such specificity does not mean that only one person understands this language, but rather that the small details of the language are unique to one person. This middle tier represents the encoding system used. Finally the bottom tier gives the statement which is the expression of the pure truth by the realization of the pure Universal Grammar. This tier is separate from the middle tier because its element is the result of a realization of one ideal operating on the other ideal. This bottom tier represents the product of the encoding. Note that no thought is ever expressed in pure form, but each communicated thought is a pure thought that has been encoded with some language. This encoding process allows communication, but results in each thought that is communicated undergoing some language specific encoding.

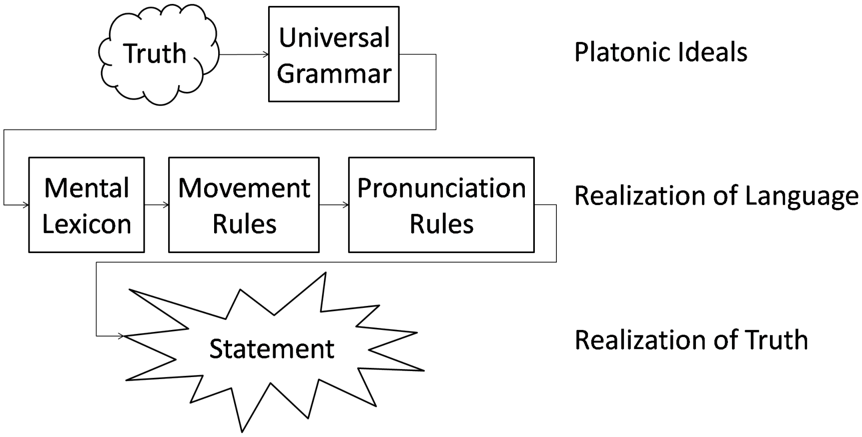

Figure 2.4: Encoding of Some Truth with Language

Note that this encoding of thoughts occurs for both thoughts that are destined for communication with external entities and those intended for internal communication. On a high level, intelligence is the consideration and manipulation of thoughts in a self-contained manner (typically contained in a single brain). Within the brain these thoughts are not generally operated on in their pure form, but are rather encapsulated in some language encoding first. Thus the ability of the encoding used to describe concepts directly affects intelligence. These concepts of the impact of language on intelligence are further explored in the Sapir-Whorf Hypothesis (linguistic relativity) [11].

Extending these concepts to the performance evaluation, it is clear that language both affects how the results of some action are externally reported and how the action itself is determined in the first place. The impact of each of these languages, external language and internal language, must be considered during performance evaluation. The impact of external language is slightly more straight forward in that the language used effectively results in some transform on the information collected for consideration and on the information reported. The goal of a performance evaluation system, in terms of external language, is to apply an inverse transform to determine the true situation observed and the true result of the action taken. Internal language, on the hand, has the more subtle affect of shaping the ability of the cognitive radio to reason about the situation at hand. Taking the knobs and meters of the cognitive radio to be the lexical items of its language, the language used defines the ability of CR to internally express situations. This variability of expression effectively shapes the radio's ability to notice (and react to) the important aspects of the scenario. Thus the internal language of the cognitive radio is another set that defines some limit on the maximum CR performance. Understanding both of these languages is of key importance to the performance evaluation of cognitive radios.

2.2.4 External Language

In the scope of the discussion of the CR's language, the knobs and meters provide the lexical items of the CR's language. These items provide the full description of the situation as observed by the cognitive radio. Effectively these items fully describe what the CR sees before and after it is determining which action to take.

The information used by the CR is transformed based on the available lexical items and their interactions. This transformation takes some truth about the situation and distorts it for consideration within the cognitive radio, in much the same way that data is distorted during the transformation from analog to digital. In both cases it is entirely possible that meaningful portions of the original information are lost (or simply changed) during the transformation. Understanding this transformation has two applications to the performance evaluation of CRs. First, determining how much useful information is lost helps to define the maximum performance of the CR (determines degree of difficulty). Second, achieving a useful inverse transformation allows for the decoding of the results of actions. Decoding the results of the actions provides a more clear understanding of the situation that results from some action that has been taken and helps to remove the possibility of a radio erroneously reporting some result as good. Thus the understanding the transformative nature of the external language of the cognitive radio provides necessary information to the performance process.

The transformation of the external language used by the CR is inherently complex. Borrowing results from the study of human language, the most straight forward way to arrive at this transformation is the setting of parameters based on experience. An experience based approach for determining and removing this transformation is presented in the next chapter.

2.2.5 Internal Language

The internal language of the CR shapes the ability of the radio to reason about situations. As discussed above the choice of the knobs and meters affects the expressive power of the cognitive radio and certain combinations of knobs and meters are certainly better for certain situations. Fully understanding these effects of knob and meter combination on performance is certainly a goal of CR performance analysis. However, this is a very complex problem, requiring a great deal of careful exploration. As this work is focused on providing a foundation for the performance evaluation of CRs, the internal language issue is simply raised here, but not further explored.

2.3 Actions

Note that the actions of a CR are an entirely different aspect of the domain of a CR's performance. While the sets discussed above capture the background knowledge required for evaluation, the actions are the foreground knowledge. As such actions of a cognitive radio must be considered somewhat differently.

Note that in the initial construction of the set based domain for CR performance background information was added to the domain to provide a generalization of performance evaluation. This generalization is necessary to apply the same evaluation framework to CRs in different situations. Actions, on the other hand, are always necessary to determine the performance in every situation. Therefore actions form the fundamental domain of the performance function that has been complicated by the addition of background information discussed above.

However, the addition of the background information to the domain clarifies the role of actions in addition to generalizing the performance function. Note that the manner in which actions are recorded or expressed is captured by the language category of background information discussed above. Thus, in a domain that considers the representation of actions and the actions themselves as separate independent sets, the set of actions have the very straight forward purpose of capturing what the radio truly did, without referring to the transforms engendered by how the radio described what it did.

Note that in practice, an evaluator will typically not have direct access to the actions of the CR, but rather only to the CR's record of its actions. This weakens the practical independence of actions and language, but it is important to note that the conceptual independence certainly exists. An action may, in fact, be expressed in a number of different ways. However, in practice an action is only expressed in one way and understanding the nature of that expression is necessary to unlocking the action, just as understanding the nature of a human language is necessary for unlocking the meaning of an idiom.

In fact, in practice all of the background information discussed above is necessary for the understanding of the performance that results from some action. The entire reason for considering the background information is to understand the connection between obfuscated actions and obfuscated performance. Note that actions and performance are parallel in many ways. First, the quantitative values of both are in fact hidden from view. The true actions of the CR are obfuscated by the manner in which they are expressed and the true performance is obfuscated by the precise situation. Note that a somewhat qualitative description is easily available with examination of a record of a cognitive radio's behavior. Second, note that two categories discussed above are concerned with clarifying either actions or performance. The purpose category of background information defines what high performance is and how the environment shapes it. The language category of background information defines what an expression of an action means and how the available language shapes action selection. This parallelism suggests a structure for performance evaluation that is discussed in the next chapter.

2.4 Conclusion

This chapter has introduced a performance evaluation system based on a domain constructed from the intersection of independent sets. A collection of sets used for this domain has been introduced and explored. As part of this exploration, the domain sets have been divided into the two categories of purpose and language. Each of these categories was then shown to be related to the encoding of one half of the evaluation problem.

Chapter 3

Implementation Approach

This chapter explores the complexity related to implementing the performance evaluation system discussed in the previous chapter. The discussion here begins in Section 3.1 with a reorganization of the performance function that better suits realization. Limitations of the scope of this work are given in Section 3.2. The conceptual framework for implementation based on the reorganized performance function is then given in Section 3.3. The method of implementing this conceptual framework is discussed in Section 3.4. As Artificial Neural Networks play a major role in the implementation of this system, an overview is given in Section 3.5. Following this overview, an exploration of the specific application of Artificial Neural Networks to performance evaluation in Section 3.6. The chapter ends with a brief discussion of the software package used to implement the evaluation system in Section 3.7.

3.1 Composite Performance Function

At the end of the previous chapter, the parallelism between the actions of a cognitive radio and the mission based performance was discussed. This parallelism suggests an approach for implementing an evaluation system. Note that the performance function has been discussed in the form of $\mathbf{f}(\mathbf{x}) = p$, where $\mathbf{f}$ maps from an n-dimensional domain, constructed by the intersection of various sets, to a real number p, which is the performance. In this form x is some combination of the actions of the cognitive radio and the background information associated with the particular evaluation situation. However, the parallelism between actions and performance of the cognitive radio arises because neither is directly accessible. The actions are encoded by the language of knobs and meters used by the cognitive radio and the performance is encoded by the actions and the purpose of the cognitive radio. This parallelism suggests that the performance function be rewritten as $\mathbf{g}(\mathbf{h}(\mathbf{z} \mid \mathbf{l}) \mid \mathbf{o}) = p$, where $\mathbf{h}$ is a transform that maps the recorded knobs and meters, $\mathbf{z}$, to some indicators of action, given $\mathbf{l}$ which is information related to the cognitive radio's language and $\mathbf{g}$ is a transform that maps the result of $\mathbf{h}$ to the performance, given $\mathbf{o}$ which is information related to the cognitive radio's purpose. Finally note that the final performance score is shown here as a scalar; however, in many situations this score may be more useful as a vector. A n-dimensional vector performance score would simply relate the performance in n high level aspects of the mission. If a vector performance score is used, the magnitude of $\mathbf{p}$ would then give an overall performance.

This new form for the performance function provides several advantages while maintaining the same purpose and foundation of the original. The composite structure highlights the parallelism of the actions and performance of the cognitive radio, emphasizing the need to decode each from provided knobs and meters. The use of $\mathbf{l}$ and $\mathbf{o}$ as given knowledge clarifies the use of background knowledge in the evaluation. Note that this form does not alter the domain of the performance function, as the argument of $\mathbf{f}$ can be written as $\mathbf{z}$, $\mathbf{l}$, $\mathbf{o}$. Rather the composite form of the performance function simply divides the usage of the various sets that make up the domain of $\mathbf{f}$. Note that as part of the parallelism of the actions and performance, the category of language information pertains to understanding the actions and the category of purpose information pertains to understanding the performance. The division of the sets engendered by the composite form makes these relationships clear. This composite structure provides a much clearer path to implementation than the non-composite form.

3.2 Decomposition Based Approach

This work is focused on simply providing a foundation for the evaluation of cognitive radio performance. Thus in order to provide a clear example of the evaluation approach presented here, only a subset of the full performance function domain is examined. Note that for decoding both the actions and performance of the cognitive radio, one of the two associated sets provides the fundamental definitions and the other provides additional context. In both cases the additional context is necessary for comparison across a variety of situations. Environment information allows an evaluator to make comparisons of radios operating in different external situations. Internal language allows an evaluator to make comparisons of radios operating with different internal situations. However, neither of these two sets directly defines an encoding for a given situation. The other two sets, mission and external language, provide the information that defines the meanings of the encodings of actions and performance. Thus, both the environment set and the internal language set provide a second layer of information. Capturing the knowledge provided by these two sets is a non-trivial task, but the existence of this information does not change the overall approach to performance evaluation. Therefore, in order to focus on the approach, these two scaling sets are not considered in the proof of concept implementation discussed here.

The two definition sets, mission information and external language information, contain all the information required for a minimal evaluation of cognitive radio performance. As discussed previously, the concept of performance of some endeavor is inherently tied to the goal of the endeavor. The set of mission information is, therefore, very clearly required to define what is good. The external language information serves a parallel purpose for the definition of the actions of the cognitive radio. This set of information defines what the cognitive radio means when it records a collection of knobs and meters for later consideration. Thus, these two sets directly provide the information required to decode both the actions and performance of a cognitive radio. The context sets, environment information and internal language information, only provide the information required to make comparisons across situations.

3.3 Conceptual Framework

The composite function discussed above provides the next step towards the implementation of a performance evaluation system. Splitting the performance function into sub-functions allows for the exploration of parallel aspects of the actions and performance of a CR. However, note that this separation is not perfect and a real implementation approach may take advantage of the benefits of both the composite and non-composite forms of the performance function. Thus, the composite form of the function provides the initial framework for implementation that will be explored here, but several benefits of the non-composite function are also utilized. Such a hybrid approach provides both a clear framework and a realizable method for the implementation of a performance evaluation system.

In order to further explore the composite function framework for the performance evaluation of CRs, it is first helpful to examine more deeply the evaluation approach taken in [12]. The authors approach the performance evaluation from a network viewpoint, but many of the proposed techniques are applicable to this discussion. Their evaluation approach is based on viewing a cognitive radio network as a complex system; *i.e.*, cognitive radio networks have some level of well defined structure, but they also exhibit many variations [13]. Due to this complexity, “what is seen often depends on the size of the observer” [13], or perhaps more plainly “a more holistic approach to performance evaluation is required” [12], compared to the currently available approaches. The authors conjecture that the focus of performance evaluation should be on the ability of a cognitive radio network to deliver required services rather than a detailed characterization of the communication approach. To this end, they introduce six capacity descriptors: Effectiveness, Survivability, Efficiency, Stability, Security, and Legality. Each of these descriptors measures an aspect of the general success of an arbitrary CR network with regard to an arbitrary task. The capacity descriptors of [12] represent very powerful concepts that extend well beyond the use presented by the authors of that work.

In order to better discuss the extension of these ideas, two labels used in [12] are helpful. These labels are Measures of Effectiveness (MoEs) and Measures of Performance (MoPs), which have been borrowed from systems engineering [14]. While MoEs and MoPs do not have agreed upon definitions, they provide good handles for the discussion of these topics. The capacity descriptors discussed above directly provide the MoEs, which will be defined here to provide a measure of some quality of a CR's action. The MoPs, on the other hand, provide a measure of overall performance. Each of these concepts provides a good method for discussing the implementation of a performance evaluation system.

3.3.1 Measures of Effectiveness

In [12], MoEs are used to describe the various aspects of a cognitive radio network, *e.g.*, its effectiveness at accomplishing a given task or survivability in harsh environments. However, the true potential of these descriptors is in their ability to capture a general assessment of a cognitive radio's actions. Each action impacts the capacity of the cognitive radio in each of the six fields. Taken together these MoEs capture the key performance related information provided by the actions of the cognitive radio. Table 3.1 provides brief definitions of each MoE as used here.

Table 3.1: Definitions for MoEs as Related to Actions

MoE	**Measured Quality**
Survivability	Impact of action on long-term operation of radio
Efficiency	Use of radio's resources in performing action
Stability	Likelihood of action causing unaccounted for state/oscillation
Legality	Potential of action to violate policy
Security	Vulnerability to malicious users caused by action
Effectiveness	Impact of action on current situation

Each MoE is focused on a different aspect of a cognitive radio's action. These MoEs can be divided into three groups based on the qualities they measure. The first group considers the impact of a CR's actions on the continued performance of the radio. This continued performance group is made up of the Survivability, Efficiency, and Stability MoEs. The second group measures policy issues of actions. This policy group simply contains the Legality MoE. The final group relates more directly to general performance of a CR. This high level group, the Security and Effectiveness MoEs, captures the slightly higher level information. Each of these groups serves a specific purpose and are designed to express specific qualities that result for the actions of a CR.

The continued performance group of MoEs forms the foundation of performance evaluation, because a radio must be operational to perform well. This group, therefore, relates several aspects of the impact of a CR's actions on its continued operation. Specifically, Survivability provides information related to damage done to a radio by a particular action. In most cases, this metric would measure the direct damage done by an action, and therefore, this MoE would not typically play a large role in performance evaluation. However, in more predictive systems this MoE could be used to capture information about the expected future operation of a CR based on the current action. The Efficiency MoE is more focused on the amount of resources consumed by a particular action. Note that these resources are any limited quantity that is required for and must be expended in continued operation. The resources that limit the operation of a CR most typically are power and computational ability. Finally, the Stability MoE relates the potential of an action to destabilize a CR. This destabilization may either occur in the RF hardware of the CR or, more likely, in the CE. This MoE would likely have the largest impact in the case of CRs with self-monitoring capability or those that have undergone extensive prior testing. Each of the MoEs in this group relates some aspect of an action's impact on the continued operation of a CR and therefore, on the CR's future performance.

The continued performance group of MoEs is the only group that maintains a strict separation between consideration of the language and purpose of a CR. The policy group begins to blur the line between these two aspects and embraces the hybrid approach to performance evaluation. This group directly relates the legality of a CR's action. However, the legality of a CR may be tied to the high level purpose of the CR. For example, the CR of an emergency responder may be restricted from certain frequencies, unless in use during a disaster. More simply, legality of an action tends to vary with the location of the CR. Therefore, rather than decode a CR's record of an action to reveal some quantities that eventually lead to legality, it is much more straight forward simply to include purpose information in the decoding of actions.

The high level group of MoEs continues this idea of utilizing purpose information to remove ambiguity. Easing the separation between language and purpose is especially advantageous for this group of MoEs, as they capture more high level results of actions. For example, the Security MoE relates the vulnerability to some malicious user that results from a CR's action. This vulnerability may vary greatly with the mission of the CR. For example, if a CR is used for some stealth based purpose, vulnerability may equate to detectability. However, if the mission is instead to relay sensitive information, the Security MoE is more tied to the intelligibility of the information to some malicious observer. Also, for some missions the Security MoE may not be of concern. The Effectiveness MoE is also much more straight forward when the separation between language and purpose is eased. This MoE provides the naive performance of a CR. Specifically this MoE captures the progress toward a directly measurable goal provided by an action. For example, if the goal of a mission involves maintaining a BER level, an action that maintains this BER level would have a high Effectiveness, while one that dips below the threshold would have a low Effectiveness. Clearly, for missions of very low abstraction, this MoE provides all of the necessary information. However, as the true potential of cognitive radios lies in the execution of high level missions, Effectiveness only provides a small piece of the overall performance. With a strict separation of language and purpose information, determination of an Effectiveness MoE would require achieving a perfect decoding of the CR's language to reveal the pure expression of an action. This pure expression would contain all the information related to the action, including why the CR applied the action. However, such an ideal decoding is impractical. Clearly, easing separation of background information allows for a realizable implementation.

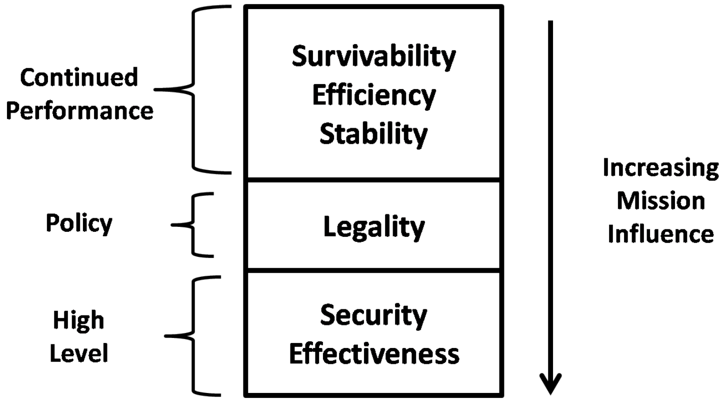

Figure 3.1: Summary of MoE Groups

Figure 3.1 summarizes the MoE groups. This figure highlights the degree of mission information needed to determine values for each of the MoEs. This mission information provides context to language information that streamlines translation of coded actions. Techniques for applying such information to determine MoE values are explored in Section 4.3.1

MoEs provide the central aspects of the implementation of this performance evaluation system. These factors describe a general set of factors that give the performance for any given mission. A realizable determination of several of these MoEs required a deviation from the composite structure of the performance evaluation function discussed above. However, the effect of this deviation is simply that purpose information, which is already required for performance evaluation, is used to provide context to the decoding of a CR's language. Thus, the structure of the composite function is still applied to the evaluation problem, by first decoding the actions of the CR and then utilizing the result of this to decode the final performance. The MoEs represent the format of the result of the first decoding, providing a consistent approach to performance evaluation.

3.3.2 Measures of Performance

While MoEs provide a consistent core to the evaluation approach, they do not provide, directly, the specificity needed to capture overall mission performance. Instead, MoPs provides a distillation of the overall performance of the cognitive radio. The Measures of Performance change with each mission to provide a holistic view of the performance for each mission. The Measures of Performance completely the picture of CR performance by allowing the examination of performance in high level missions.

Selecting Measures of Performance is inherently a mission specific task. Previously the final performance value, produced by the performance function, was discussed as being either a scalar or vector. This determination largely rests with the needs of the evaluator for a particular mission. Some missions may require examination along more than one dimension, while others only require the consideration of a single factor. As MoPs directly provide this final mission score, all of the same concepts able. MoPs must be selected to suite the mission at hand.

3.3.3 Brief Application of MoEs and MoPs

In order to help clarify the labels MoE and MoP consider a CR with the mission objective of maximizing data transfer in a low power manner. The MoEs in this situation, as in any situation, are those discussed above and summarized in Table 3.1. The MoPs, on the other hand, must be selected based on the mission. For this mission, clearly data transfer and power use are the two aspects of interest. Thus, a measure of good data transfer rates and a measure of power used would provide the two MoPs for this case. Note that other options for MoPs are possible, if the evaluator wanted to probe the situation from a different viewpoint. However, typically the evaluator will want to use MoPs that are directly related to goals of the mission pursued by the CR. In this case, the CR has two separate goals, thus, two separate MoPs are used. This brief example provides only another connection of the labels MoE and MoP to the evaluation problem. Determination of the MoE and MoP values is discussed below.

3.3.4 Conceptual Structure

Given the concepts of MoEs and MoP as discussed above, the scheme for performance evaluation is most clearly defined as a four-tiered structure. This structure is shown in Figure 3.2. On the lowest level are the cognitive radio's actions. The knobs and meters, on the next level, encode these actions into the cognitive radio's language. The MoE, on the next level, represent non-radio specific distillation of the cognitive radio's actions. Note again that, a major purpose of the MoEs in this structure is to provide a consistent abstraction away from details of a specific cognitive radio design. MoE are, in fact, formulated as general aspects of an action to accommodate any given evaluation. The MoPs, on the other hand, provide mission specific values that directly provide the final performance score. This structure is based on the composite performance function and consistently transitions from specific cognitive radio behavior to mission specific performance. It is this consistent translation that allows the method to calculate the performance of an arbitrary cognitive radio with regard to an arbitrary mission.

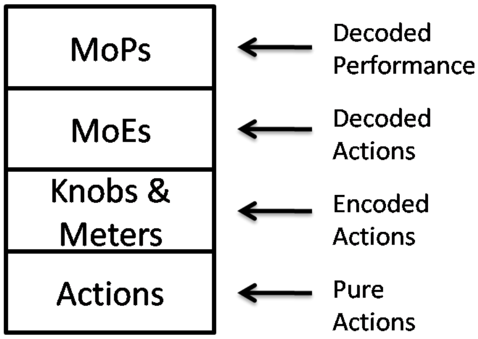

Figure 3.2: Four-Tiered Structure for Performance Evaluation

3.4 Concrete Structure

Although the conceptual structure explains the principles of the performance evaluation system, it does not necessarily reveal the implementation details of this system. The evaluation system is best implemented as two cascaded Artificial Neural Networks (ANNs). Each the ANNs used in the implementation corresponds to one of the two conceptual labels, MoEs and MoPs, discussed above. Figure 3.3 gives a functional structure for the performance evaluation system.

The first ANN is tasked with decoding actions of the CR, record as knobs and meters. This ANN is trained with the external language information to provide the definitions of the CR's language and with the mission information to provide additional context. This training allows the first ANN to determine the MoEs from the language of the cognitive radio. Using an ANN for this purpose is a good approximation of a human brain understanding the meaning of written language. A more detailed examination of the abilities and utilization of ANNs in this regard is given below.

The second ANN implements the second half of the composite function, mapping MoEs to MoPs, based on solely mission information. Once again a comparison to human use of language is apt here. The mission information used to the train the second ANN contains the knowledge of how each MoE relates the overall performance. The second ANN must learn this language of mission to combine the MoEs and produce the MoPs.

It is worth noting that, in this implementation the explicit determination of the MoEs is not strictly necessary. However, the explicit calculation of the MoEs offers two major benefits. First, this calculation provides addition information that may be used to assess a specific CR's performance over several different missions. Recall that the Effectiveness MoE directly provides the naive performance of the CR, which may be useful for debugging either the CR or the evaluation system. Second, the explicit calculation mirrors the composite form of the performance function, which clarifies many aspects of the process. Thus, while not strictly necessary, calculation of the MoEs provides benefits to the evaluation system.

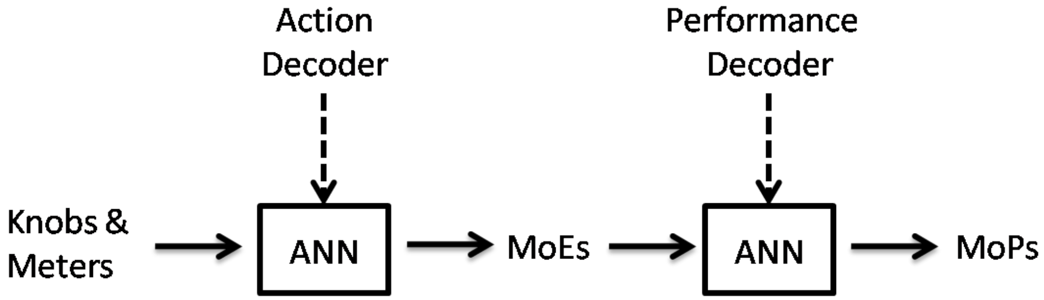

Figure 3.3: Functional Structure for General Performance Evaluation

ANNs are the clear choice for the translation of values in this system. The connection between the various levels is not necessarily clear; in this case providing several examples of desired performance is a far more tractable problem than explicitly describing a functional relationship. ANNs also provide the ability of compact data representation. This means that once a specific cognitive radio or mission is evaluated with this system, the weights and topology of the ANNs used are all that is required for a direct comparison. Thus, ANNs handle providing both translations and consistency when comparing for the task at hand. This application is further discussed in Section 3.6.

3.5 Artificial Neural Network

Artificial Neural Networks play large role in implementing the performance evaluation system outlined here. The properties of ANNs make them especially well suited to the challenges presented by the evaluation problem. This section gives a brief introduction to ANNs, highlighting the attributes that are useful to cognitive radio. The specific application of ANN to the problem of performance evaluation is discussed in the following section. Readers familiar with the use of ANNs may skip to Section 3.6

3.5.1 ANN Background

Most authors attribute the original formulation of ANN to [15], published in 1943. Since then, the study of ANNs has grown into a diverse field with many variations on a central construction. This central construction is based on a model of the human brain as an interconnection of artificial neurons. The simplest possible case, a single neuron, provides a straightforward instructive example for the operation of ANNs. In this case our single neuron ANN produces an output that is the result of applying an aggregation of weighted inputs to a nonlinear activation function. This aggregation is often a linear sum, but can, in fact, be any combination of the inputs. More complicated ANNs are simply collections of these single neuron ANNs arranged into layers. For multi-neuron ANNs there are, typically, at least two layers, an input layer and an output layer. The input layer accepts inputs from some external source and the output layer aggregates the outputs of the input layer into the final outputs. Between these two layers any number of so called hidden layers may be used. A three-layer example of this structure is shown in Figure 3.4. The arrangement of artificial neurons into such layers has the effect of creating arbitrarily complicated sets of nonlinear equations. ANNs,theoretically at least, have the ability to model any system or function. Thus, ANNs provide a powerful and flexible model of the human brain [16].

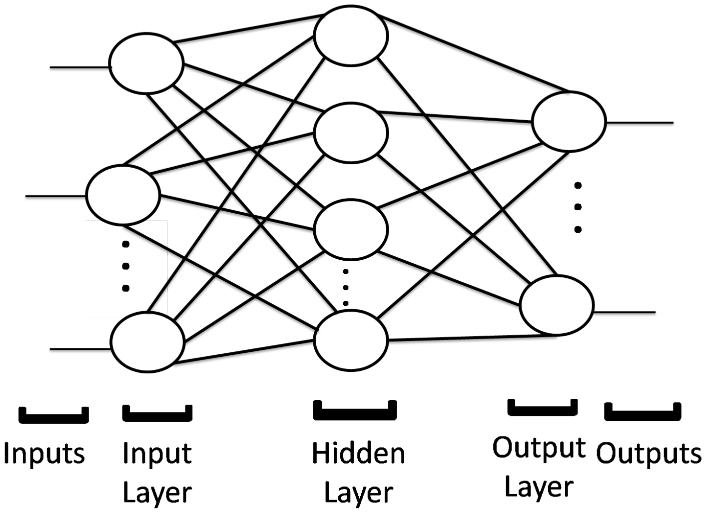

Figure 3.4: Generic Structure for an ANN

Learning in ANNs is achieved through the manipulation of the weights on the inputs to each neuron. Several different learning mechanisms exist, but all are based on the propagation of knowledge from a set of training points through the network. Training points are a set of corresponding inputs and outputs that represent the desired performance of the ANN. Knowledge of both the inputs and outputs from training points is used to adjust the input weights for each neuron until the behavior of the ANN is sufficiently similar to the desired performance. Traditionally this training takes place in a distinct phase prior to operation, but as the training mechanism does not require external guidance, this training phase can occur during autonomous operation of a cognitive radio. It allows ANN to directly learn desired behavior from experience [16].

3.5.2 ANN Training

The process of training an ANN is the process of teaching the ANN some desired functional behavior. Fundamentally, an ANN is nothing more than a function with several parameters to be set. Teaching an ANN is then an optimization problem over these parameters where the error between the desired output and the actual output is minimized. Nearly any optimization technique may be used for this problem, but gradient descent is the most popular technique for this purpose.

Gradient descent offers a straight forward method that is connected to the best known ANN training algorithm, back propagation. Back propagation is an extension of the Least Mean Squares technique for parameter setting, [16], that updates weights to be a multiple of the negative gradient of the error function with respect to the weights. The multiple used here allows control of the learning rate, which is useful for balancing speed of training and cycling. This updating occurs iteratively until the magnitude of the gradient is below some threshold, signifying that desired output and the actual output are sufficiently close.

Note that this discussion of ANN training is brief and does not cover many important details. However, this discussion does highlight some aspects of ANN training that are important to performance evaluation. First, the process of training an ANN is the same as the process of optimizing coefficients of an arbitrary function for a given set of input and output values. Second, back propagation represents a well studied and straight forward training process. Third, back propagation is based on the gradient of the error function with respect to the weights. This means that the activation functions must be differentiable for the use of back propagation. Together these factors present conditions under which ANNs represent a well studied method and straight forward method of implementing arbitrary functional behavior.

3.5.3 General ANN Advantages

Ideal ANNs provide the ability to represent compactly nearly any knowledge as a set of weights. This ability is especially appealing with regard to functions that may be difficult to formulate otherwise [17]. As ANNs were originally formulated as a mathematical model for the human brain [15], they are directly linked to the only truly observable intelligence in existence. In fact, a properly constructed ANN has the ability to implemented any function, in terms of output given some input, with only three layers [16]. As a result of this ability to compactly represent general knowledge or behavior and nearly three quarters of a century's worth of study, ANNs clearly represent an appealing solution to many CR problems.

3.5.4 General ANN Disadvantages

ANNs face two majority issues that conspire to restrict their usefulness: overfitting and the curse of dimensionality. Overfitting robs ANNs of effective generalized performance through specialization to the set of training points applied. This issue is especially common for complex techniques ANNs with too many parameters, as these have the expressive power to represent each nuance of the training data. Avoiding overfitting effectively requires regularizing the ANN, reducing the number of parameters to a level appropriate for the situation at hand. The curse of dimensionality is the dual problem to overfitting; it arises in high-dimensional spaces that require a huge number of points to render any training set statistically significant. The collection and management of such sets tend to prevent the operation of machine learning in high-dimensional spaces. The problems of overfitting and dimensionally effectively combine to define boundaries on the applicability of ANNs [16].

In terms of general CR use, ANNs are plagued by a few more issues as well. Chief among these is a predisposition away from incremental learning. ANN training methods are generally not well suited to starting with a small number of training points and adding more as they are collected [16]. This means that ANNs are ineffective in completely unforeseen scenarios until an appropriate number of training points have been collected and labeled. This predisposition away from incremental learning is perhaps the largest obstacle for using ANNs in non-deterministic, quickly changing cognitive radio applications. This disadvantage applies to the performance evaluation work presented here as a potential limitation of future use. The evaluation system discussed here is presented as an off line based system, but the techniques could be applied to more online usage and help guide CR behavior. However, this issue with dynamic situations may prove to a limiting factor of such applications of this evaluation approach. Other problems related to the accuracy of various training methods have be studied for other applications of ANNs [18], but the inherent learning strategy of ANNs have a larger impact on performance evaluation considerations.

3.5.5 General ANN Application to Cognitive Radio

The classification of patterns based on previous examples likely provides the most common use of ANNs for cognitive radio. Such classification is manifested in various tasks such as signal classification [19], decoding transmitted symbols [20], and even identifying transmitters [21]. ANN classification has also been used to identify more abstract concepts such as the best use of resources in DSA systems, [22] and [23]. Such classification based uses provide the majority of ANN application to cognitive radio, and offer a diverse set of abilities.

But many more exciting applications of ANNs are possible. For example in [24], ANNs are used to model communication performance, and the authors of [25] propose ANNs to provide decision making that better considers available information. These unique approaches more fully take advantage of ANNs' ability to model complex functional behavior based on training examples. There are many novel future applications of ANNs to cognitive radio that stem from this ability to represent complex behavior or knowledge in a more straight forward way. As an example, ANNs provide an alternative method for approaching the Kolmogorov complexity for internal data representation in a cognitive radio. Specifically, once training has occurred, the complete set of training points, and therefore the complex behavior, may be traded for the set of neuron weights. Both this ability of compact data representation and the prowess in implementing arbitrary functional behavior make ANNs a clear choice for performance evaluation.

3.6 ANN Based Evaluation

There are a number of factors that make ANNs well-suited to the challenges of performance evaluation. These include the ability to compactly represent abstract information, the ability to learn from examples, a close connection to the human brain, and a wide base of study. Each of these attributes provides a necessary aspect to a performance evaluation system.

Arriving at a single measure, or small set of measures, for performance requires the distillation of a great deal of information. ANNs perform well in this situation, due to their ability to represent information compactly. Compactly representing information allows the the ANN to reduce the abundance of information required for an accurate evaluation of performance into a manageable form. Use of ANN reduces the information require for performance evaluation to a set of weights and an ANN topology. This smaller set of data distills the spirit of the information captured in all the sets of the domain that describe the situation at hand into a simple set of coefficients.

Perhaps more importantly, though, the ability to learn from examples allows ANNs to transform the abstract into the concrete. Note that to handle the information for an arbitrary evaluation case, the functional form of each of the two transforms associated with performance evaluation would have to be extremely complex and parameterized. This is a direct result of the variety of possible cognitive radio implementations and missions. Handling one such function would quickly become difficult and some systematic method of parameter setting based on the situation would be required. Furthermore some applicable description of the situation would first have to be developed. ANNs directly provide both of these requirements in their ability to learn from example. The training techniques discussed above directly provide a method of setting the parameters of the complex function defined by the ANN to match the situation, as described by the training examples. ANN provide a readily available tool for accomplishing an otherwise difficult task.

The connection to the human brain provides a subtle, but direct connection to the use of ANNs in performance evaluation. Note that in the previous discussions, the major tasks of the presented performance evaluation system revolved around the use of some language. The language of the cognitive radio (the knobs and meters) must first be decoded to reveal the key aspects of the actions of the cognitive radio. These key aspects must then be translated into the language of the mission in order to express the goodness achieved. Each transform of performance evaluation is very clearly expressed as the translation to or from some language. Note that, as far as is known, the only thing to have mastered language is, in fact, the human brain. Whether this mastery stems from the parallel parameter based structure of the human brain and language itself is unclear. What is clear, though, is that the human brain is the leading power in the use of language. Therefore, a language translation system should clearly be based on the design of the human brain. As performance evaluation is, in fact, a language based system, ANNs are the clear choice to provide the translations required.

More concretely, the use of ANNs for performance evaluation rests on their ability in the interpolation of functions. This ability is perhaps most directly explored in [26]. The authors of this work note that use of ANNs to generate outputs for inputs outside of their training set exactly matches the use cases of interpolation. They go on to note that while many techniques exist to interpolate scalar functions, these do not generalize well to vector functions. Additionally, the authors show that in the scalar case ANN based interpolation performs as well as other established methods. The authors note three main properties of ANN based interpolation, which they refer to as a Neural Network Interpolator (NNI):

> 1. It allows complex multidimensional maps with minimal *a priori* information; that is, the functional form of f is not specified in advance.
> 2. The NNI treats the data in a global way, so it is an efficient method to deal with unbinned and/or sparse data.
> 3. The noise resistant property of the neural network techniques gives the NNI robust behavior when one need to deal with statistical fluctuations of real data. [26]

Finally the authors also provide several guides lines for implementing ANN based interpolation with these properties. For purely positive output, the authors suggest sigmoid activation functions as they model biological neurons well. The hyperbolic tangent maintains this shape over both positive and negative outputs, which provides for more general operation. The authors underscore the expressive ability of this type of network, noting that in the ideal case a three layer network with this style of activation function has the expressive ability to realize any continuous function. However, in practice, the authors note that three layer networks, trained with back propagation face convergence issues in some cases. Thus, it is suggested to use one hidden layer for each important local maxima. When followings these guide lines, the authors of [26], display the successful application of ANNs to the problem of interpolation.

Such successful study of the application of ANNs to interpolation, nearly directly shows the applicability of ANNs to the evaluation problem. In fact, the remaining portion rests simply on ensuring that the use of ANNs for evaluation involves interpolation rather than extrapolation. The distinction between the two is made in the training sets used to set the weights of the ANN [27]. ANNs used for interpolation are trained with examples that cover the full range of inputs that will be using in operation. Extrapolation ANNs, on the other hand, are trained based on only a portion of their functional domain. Several use cases for both interpolation and extrapolation ANNs are discussed in [27], but the key distinction between the two is this range of training. This same work shows that, in general, interpolation based ANNs perform well, while extrapolation based ANNs do not. Thus, training the ANNs over the entire range of operational inputs is the final task needed to cement the applicability of ANNs to performance evaluation.

3.7 Enabling Software

The major components of the evaluation system presented here are the ANNs. The above discussion has focused on the applicability of ANNs to general performance analysis of CRs. This discussion focuses on the ease with which such powerful techniques can be implemented.

Several packages to implement ANNs exist for various languages. Python offers several high level capabilities and is the choice the implementation of this evaluation system. The most capable ANN implementation package in this language is PyBrain [28]. This package offers pre-implemented methods for building, training, and testing ANNs. The package allows for flexible creation of ANNs with any number of layers, any number of nodes per layers, and any activation function in the nodes. Common activation functions are pre-built but package includes the capability to add arbitrary activation functions. Similar ability exists to customize any individual aspect of the network, while maintaining connection to all other pre-built features. PyBrain represents an extremely powerful tool for implementing ANNs.

Chapter 4

Proof of Concept

This chapter will discuss a proof of concept implementation of the evaluation system presented by this work. The discussion will walk through the operation of the evaluation of a pre-existing cognitive radio. This step-by-step description of the evaluation process will discuss the key aspects of using this system to determine cognitive radio performance.

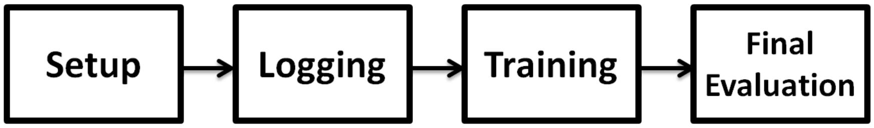

Figure 4.1: Phases of Evaluation

Figure 4.1 shows the four distinct phases of the evaluation process. Section 4.1, discusses the Setup Phase that involves describing the cognitive radio to be evaluated and the situation for evaluation. The Log Phase, covered in Section 4.2, can then focus on collecting data. This data provides the foundation for the Training Phase, explored in Section 4.3.3, to prepare ANNs for the final phase. This final phase, the Evaluation Phase, encompasses determining the final performance of the CR and is the subject of Section 4.4.

4.1 Setup Phase

The purpose of the Setup Phase of evaluation is determining the specifics of the evaluation problem at hand. Naturally, the primary information involved is some specification of the CR under test and the purpose for this radio. Evaluation could certainly not proceed without these elements.

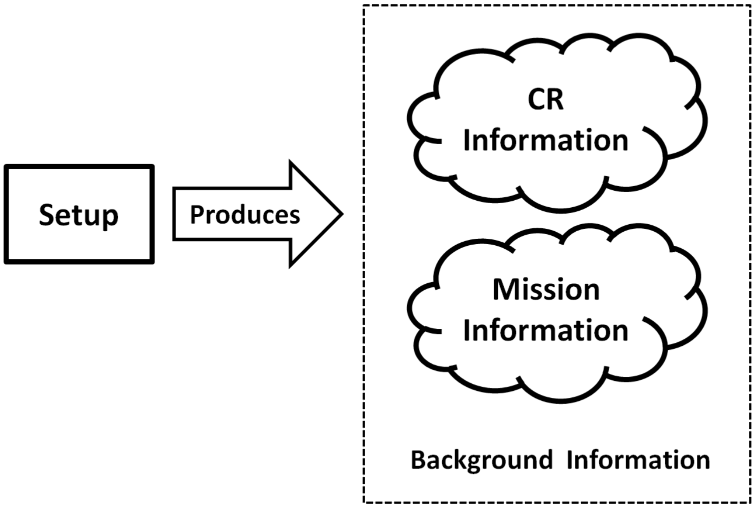

Figure 4.2: Information Relation to the Setup Phase

Note that this phase gathers the background knowledge for performance evaluation, as shown in Figure 4.2. An examination of the cognitive radio being evaluated during this phase will provide the language information necessary for evaluation. The determination of the situation in which the CR will be tested, provides the purpose category of background information discussed in previous chapters. For the purposes of this proof of concept the CR will be enabled by the Cognitive System Enabling Radio Evolution (CSERE) and the situation for testing will be outlined below.

4.1.1 The Cognitive System Enabling Radio Evolution

CSERE represents the next generation approach to CEs developed by Center for Wireless Telecommunications (CWT). [1] CSERE represents a design approach to implementing CEs. This approach evolved from experience with the original CE and focuses on three principles:

1. Modularity
2. Data Introspection
3. User Ease

These principles form the core of CSERE. Modularity in CSERE focuses on the encapsulation of individual components of the system. Such encapsulation allows components to be developed, tested, and used independently. Data Introspection aims to provide transparent access to the internal information of CSERE, which allows for straight forwarding debugging and development. User Ease means that a goal of CSERE is to take the burden off of users.

---

[1] Thanks to Alexander R. Young, GRA in CWT at Virginia Tech for the wording of this CSERE overview material

CSERE is designed to provide CEs that are easy to use, install, modify, and explore. These principles make CSERE CEs well suited to general purpose research use.

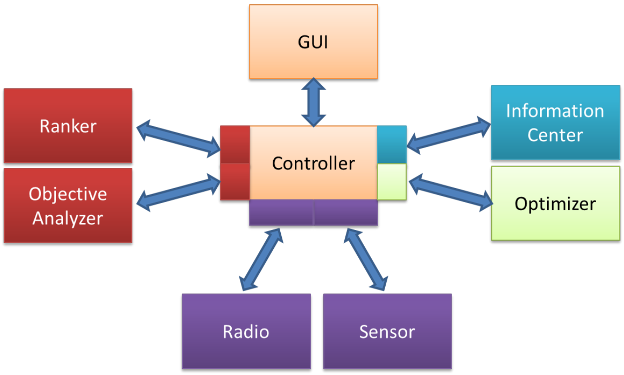

Figure 4.3: CSERE System Organization

Figure 4.3 provides an overview of CSERE's general organization. This star organization allows the controller to manage all of the other components during operation. Each component connected to the controller provides a particular aspect of the system, from user interaction to waveform deployment. When implemented a CE based on this approach the existence of each component is left to the designer.The interaction of these components allows for the cognitive control of a radio.

For the purposes of this proof of concept the CE has three available knobs. These knobs are modulation, coding rate, and transmit power. The modulation of the radio is restricted to digital M-ary modulation, where M is allowed to be 2, 4, or 8. The available transmit power ranges from -10 dBm to 15 dBm, but restricted by policy to less than 10 dBm. Finally, three levels of Forward Error Correction (FEC) coding are available. Here FEC is provided by BCH coding where the options are (15,11), (15,7), or (15,5) codes. These knobs provide a full characterization of what the radio is able to do.

In addition to these knobs, the CE will use two meters for this proof of concept evaluation. These meters are BER and occupied bandwidth. The BER measurement is examined further below, in the discussion of the mission of evaluation. The bandwidth is calculated from the symbol rate and roll of factor ($\alpha = 0.5$) as shown below in Equation 4.1. These two meters provide additional measures of the radio situation.

$$B_{occ} = R_s(1 + \alpha) \tag{4.1}$$

Note that for the purposes of evaluation, the optimization techniques employed by the CR do not matter. Rather, the knobs and meters provide the understanding of the CR's language that is necessary for the evaluation process. Additional information about the radio supplements this understanding further to facilitate evaluation. This information highlights operational details of the CR, such as whether the CR is mobile or the range of input data rates. Here the radio is non-mobile and the data rate at the input will be fixed at 6 Mbps. For this proof of concept, this is the only information that is required to proceed with evaluation.

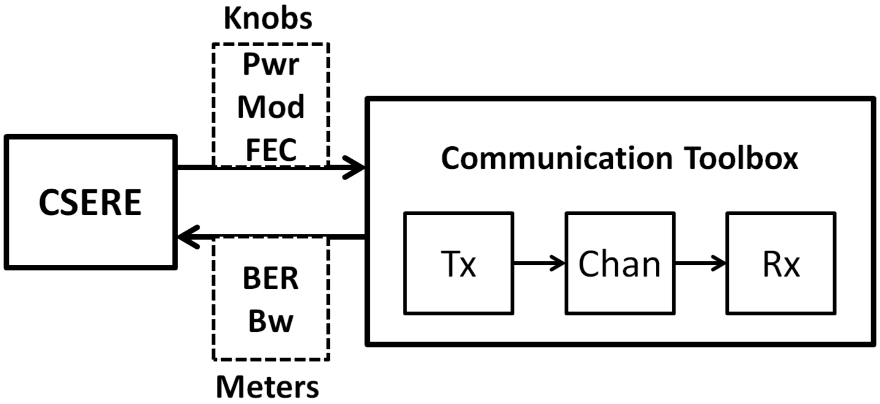

Figure 4.4: Architecture for CE Simulation

Figure 4.4 provides the architecture of the CR used for this proof of concept. In order to provide a straight forward example, the entire CR is implemented in Matlab; see Appendix D. The major elements of the CR are the radio portion, enabled by the Matlab's Communications Toolbox, and the CE, enabled by CSERE. The only communications that pass between these are the knobs and meters, which provide the language used by the CR to describe actions and their results. Knobs and meters, like these, reflect the heart of the interaction between any CE and radio, capturing the essence of the dialog that enables the cognition of CRs. Modeling this dialog is, therefore, the entire purpose of the CR implemented here.

4.1.2 Mission for Evaluation

The mission used in this proof of concept evaluation is straight forward, but embodies the foundation of many much more complex missions. As discussed previously, complex missions are combinations of simpler ones. The mission used here represents a key middle point of this continuum of complexity, as it is a combination of more simple missions and missions of this kind often form the basis for more complex missions. Choosing a mission of this sort provides enough complexity to explore performance evaluation beyond the naive case, where MoEs full capture performance, without being over complicated.

Specifically this mission asks the CR to simultaneously minimize the bandwidth, transmit power, and BER across a radio link. For this mission, the CE is able to control both ends of the link through ideal control channels. The BER meter is therefore the measured BER over the link. The channel for this link is an Additive White Gaussian Noise (AWGN) channel.

The average noise power for this channel is normally distributed with mean -15 dBm and variance 2. However, the CE has no information on the channel and must probe the channel for this information. This need to probe the channel for information forces a discussion between the CE and the radio, which must be decoded to reveal performance information.

Just as additional considerations of the CR complete the necessary information for evaluation, so too additional considerations of the mission are necessary. Recall that determination of the MoEs of the high level group benefit from the application of mission information. Specifically the Security MoE is most clear when based directly on the mission concerns. Note that to avoid over-complicating the scenario for evaluation, the mission used here does not include security concerns. This situation is fairly typical for mission of mid- or low-level complexity. These considerations are not always clearly stated in mission definition, but they certainly are necessary for performance evaluation.

4.2 Logging Phase

Once the Setup Phase of the evaluation process is complete the logging phase can begin collecting data for evaluation. This phase begins with providing the CR with some randomized initial set of knob values and some mission. The CR then begins to work autonomously toward its goal, saving knob and meter values at some interval. The CR will record sets of these values each time the CE takes action to change some knob value. This process results in a time series of the knob and meter values that is sampled non-uniformly, but does highlight the changes made by the CE. Recording continues until some end state is reached. This time series can then be used for subsequent phases of evaluation.

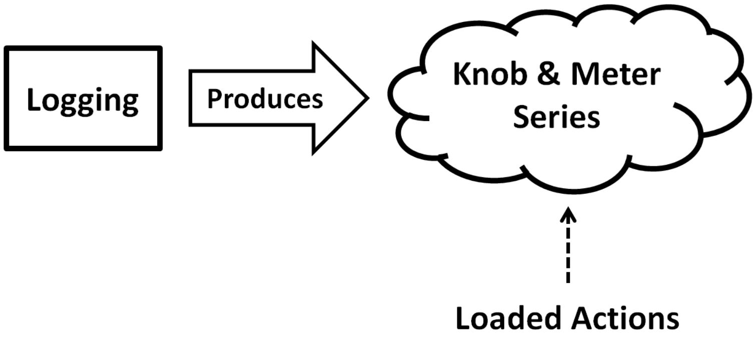

Figure 4.5: Information Relation to the Logging Phase

Figure 4.5 displays the information produced by the Logging Phase. Note that while this stage requires the Setup Phase to define the CR and mission for evaluation, the Logging Phase does not directly utilize any evaluation information. Rather this phase if focused on collecting the coded actions, *i.e.*, the foreground information for evaluation.

There are several possibilities for end states of the logging phase of performance evaluation. For numerous missions, the end step of achieving the target state of the mission is the most desirable. These mission types ease the training process discussed below. However, several missions do not have some end point that may be reached. For example, a mission of achieving some threshold has a clear target state, but missions based on providing service over time do not, necessarily, have a clear target state. Missions without a clear target state require slightly different consideration. Thus, a time-out provides a second natural end state in cases where the radio takes too long to achieve the goal or a clear target state does not exist. The mission used for this proof of concept does have a clear ending point. Specifically, this mission logging ends when the CR achieves a good enough threshold of less than equal to 10 MHz in bandwidth, 0 dBm in power, with nominally 0 BER (a value below the precision of the simulation system). Note that this ending condition does not necessarily represent the optimal condition but simply an acceptable condition. Finally, an end state must be added to account for the situation of CR failure. This end state is reached when a CR crashes. These three end states cover the needs of performance evaluation and provide clear information to the Training Phase.

As the environment is not considered here in order to allow us to focus on the foundational approach to performance evaluation, a precisely control environment is necessary for this proof of concept evaluation. In order to achieve such an environment, the radio portion of the Logging Phase is simulated using Matlab's Communication Toolbox. This allows CSERE to make decisions about the parameters of the radio, as it would in any situation. These parameters were then applied to the Matlab simulated link to provide feedback to CSERE, just as would occur for any other situation. The use of Matlab as a radio block does not alter the operation of CSERE's cognitive elements in any way. As these cognitive elements are the ultimate focus of the evaluation process, this technique is an applicable technique for evaluating a CE in a precisely controlled environment.

4.3 Training Phase

Training the ANNs represents the joining point of a subjective understanding of performance and the objective measure provided by the system presented here. This transition is made by applying consistent consideration of information from the prior two stages to model two mutli-dimensional functions. The first of these functions transforms the time series of knob and meter values into a set of Measures of Effectiveness, as discussed in the previous chapter. The second of these functions then takes the MoEs and produces a time series of Measures of Performance, which provides a characterization of the CR's performance over time. Methods for determining the necessary information to model each of these functions are discussed below.

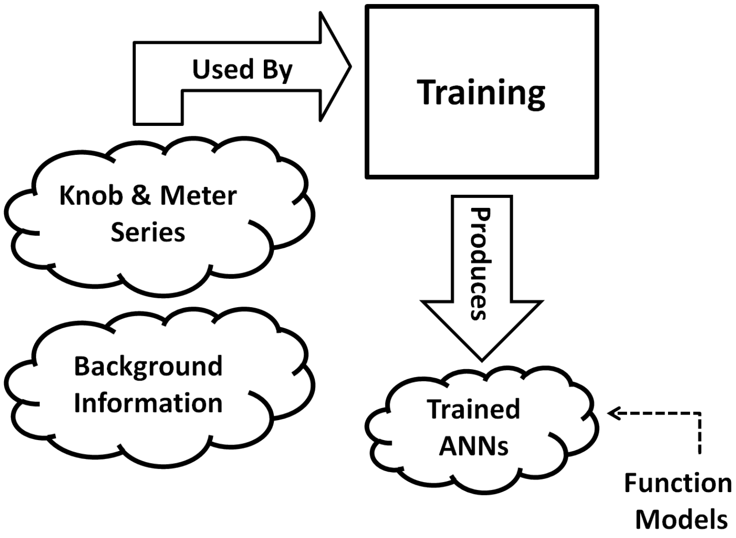

Figure 4.6: Information Relation to the Training Phase

Figure 4.6 shows the information required and produced by the Training Phase. This phase represents the distillation of the background information, collected from previous phases, into a form that is usable for evaluation. Note that this stage does not directly decode the performance score from the coded actions of the CR. Rather this stage examines all of the available information to model functions that can be used to decode this information.

4.3.1 Modeling the MoE Function

The MoEs give a translation of the CR's action from knobs and meters to reveal meaningful quantities, in terms of performance. Achieving this translation requires primarily an understanding of the knob and meter values. However, as discussed previously, determining several of the MoEs benefits from an understanding of the mission as well. Thus, the first step of achieving a translation function that maps from the knobs and meters of the CR to the MoEs is to apply available information in a useful manner. Several heuristics provide the means to build a training set for the first ANN from this information that can then be applied to achieve the necessary translation.

These heuristics are guidelines to add initial MoE values to each set of knobs and meters in the time series produced by the Logging Phase. Such guidelines provide necessary consistency to the subjective element of performance evaluation and allow the decoding of the CR's actions. These guidelines provide initial values for the MoEs associated with each set of knobs and meters in the collected time series. These values simply serve as a starting value for the MoEs that will update with techniques discussed below. Note that MoE values vary between zero and one, where zero represents an absence of the measured quality and one represents full presence of the quality.

Recall that each of the MoE groups requires a different level of knowledge for the determine of values. The continued performance group depends only on language information from the CR. Thus this group only needs background information about the CR, from the Setup Phase, and language context information from the Logging Phase. The policy group group requires the same sort of information, but can also be affected by the specifics of the mission. Policy information may be specified as either part of the CR or as part of the mission information. Mission policy information is typically more complex, as legality can occasionally depend on the situation. Finally, the high level group requires both the CR's language information and the mission information. This last group captures high level information that is complex, but necessary for performance evaluation. The varying level of information required by each these groups shapes the heuristics applied to determine initial values.

Continued Performance Heuristics This group contains the MoEs of Survivability, Efficiency, and Stability. Survivability provides a measure of the cognitive radio's ability to continue operation. As such, the primary feature that determines this ability is the end state of the Logging Phase. Of the three types of end states, achievement, timeout, and failure, failure provides the most information about Survivability. In both of the other two end states, the CR completed the full term of its operation and thus each action provided full Survivability. On the other hand, in the case of ending in failure, the CR did not survive its full operational period. As each action of the CR contributes to the final state of the CR, each action of the CR negatively affects its Survivability, in the case of failure. However, it is likely that the actions closer to the failure contributed more to the failure that more remote actions. To model this, the initial value for Survivability is set by Equation 4.2 in the case of failure.

$$0.5 - \frac{x}{2n} \tag{4.2}$$

In Equation 4.2, x is the position of knob and meter set and n is the position of the failure. In cases other than failure, Survivability is simply to one for each knob and meter set.

Efficiency provides a measure of the CR's utilization of necessary, but limited resources. The characterization of the CR during the Setup phase of evaluation reveals the presence of limited resources. In this case the limited resources are the bandwidth and the transmit power of the radio. Note that these are the limiting resources chiefly due to interference concerns. Marshall convincingly argues that consideration of bandwidth alone is not enough when determining use of spectrum [29]. He notes that consideration of bandwidth alone often leads to high order modulation schemes, to reduce bandwidth, and high transmit power, to reduce BER. However, raising the transmit power to allow high order modulation causes signals to propagate further and interfere over a larger geographic area. Therefore, in order to avoid unduly rewarding CRs that trade width in spectrum for width in geography, determination of the Efficiency MoE must consider this trade off.

In order to provide a consistent value for Efficiency, the technique used here is based on assigning a value for bandwidth efficiency and then penalizing this value based on power usage. Note that this MoE reflects the CR's ability to reason about efficiency as well as efficiency of actions. This reflects the concept that a CR that does not think about efficiency can not control utilization of resources as well as one that does. This insight into the CR's thought process is enable by the language centric approach to action determination. Thus, the method of assigning initial Efficiency values depends on available descriptors, much the same way that the initial Survivability values depend on the ending state of the Logging Phase. For example, in the case of the CR discussed above for this proof of concept, both the bandwidth and transmit power are, in fact, directly available, but supporting information is not recorded. Typically setting an value for bandwidth efficiency involves normalizing the spectral efficiency (data rate over bandwidth) by the Shannon Bound, given in Equation 4.3. This provides a comparison to how the CR performs with respect to the ideal (100% efficient) case. However, determine the Shannon Bound requires knowledge of the signal to noise ratio, which is unavailable in this case. Thus, here an empirical upper bound on the signal to noise ratio, based on radio type, is used to provide an upper limit on the Shannon Bound and thus efficiency. In this case the empirical upper bound of 30 dB is used. Note that use of this upper bound reduces the value for Efficiency, as a penalty for inability of the CR to directly consider Efficiency information. Once the bandwidth efficiency is determined the penalty for power efficiency must be applied. As the power efficiency penalty represents geographical interference, a perfectly efficient signal, from a power perspective, uses no power and a totally inefficient signal uses infinite power. Thus, the power efficiency penalty is modeled by Equation 4.4, where P is the power efficiency penalty and x in power in watts. Once both the bandwidth efficiency and the power efficiency penalty have been determined the Efficiency is simply the bandwidth efficiency minus the power efficiency penalty.

$$\frac{C}{B} = \log\left(1 + \frac{S}{N}\right) \tag{4.3}$$

$$P = 1 - (1 + x^2)^{\left(\frac{1}{2}\right)} \tag{4.4}$$

Stability, the final MoE of the continued performance group, captures the tendency of an action to cause an oscillation. While there are two major types of oscillation that affect CRs, Radio Frequency (RF) and cognitive, CRs typically only detect cognitive oscillations. A cognitive oscillation is a state loop in which a CR may become trapped. State loops occur any time the CR applies the same sequence of knobs repeatedly producing the same sequence of meters. State loops are evident from the time series of knobs and meters. There are two types of state loops, recoverable and non-recoverable. A recoverable state loop is one the CR eventually breaks, and a non-recoverable loop is one that continues until a timeout. In either case, the action directly proceeding a state loop and actions within the loop all receive a 0 for Stability. The action that breaks the loop receives a 1. All other actions receive a Stability value of 1. Both types of loops are treated the same way, because the length of the loop directly provides the number of times an action is labeled unstable. This, in turn, influences the ANN with repetition in the training set.

Policy Group Heuristics The policy group of MoEs, containing only the Legality MoE, is perhaps the most straight forward, in terms of heuristics. If a set of knobs and meters falls within the policy, as collected in the Setup Phase, the set receives a Legality value of 1. Otherwise, the set receives a value of 0. Legality is typically statically defined and therefore very simple to apply. Occasionally Legality varies with the situation or even data being passed. In these situations the CR must have the capability in its language to express these exceptions to static policy to utilize them. In this cause, additional language information will be necessary to decode when the CR appropriately violates static policy.

High Level Group Heuristics The high level MoE requires the most information and represents the most complex information. Both MoEs in this group, Security and Effectiveness, greatly benefit from mission information. Based on the influence of both the mission and the CR, these MoEs are very situation specific. The ensuing discussion provides a somewhat general approach to initialize these MoEs, but largely focuses on the mission discussed above. Note that, the mission information provides necessary context to decode the language of CR in order to examine these two aspects of performance.

The Security MoE captures the vulnerability of the CR to some malicious user. This is a very flexible MoE designed to handle any security concerns of a mission. Possible security concerns exhibit a great variety, and thus so do approaches for initializing this MoE. Security concerns that focus on staying covert are easier to quantify than others as penalties can be applied for detectable actions (high power, distinctive modulation/coding/signaling, etc.) Other security concerns can be harder to quantify, for example DSA radios that wish to avoid being herded into specific bands. In this scenario, initializing the Security MoE would likely involve penalizing changing frequencies too quickly, but several other methods are possible. Still other security threats are hard to detect in the first place, which further complicates quantifying security aspects of performance. This range of possibilities greatly complicates initializing the Security MoE. In this case, the mission used for evaluation does not have security concerns, and the Security MoE may be initialized to 0.5 for all knob and meters sets.

The Effectiveness MoE captures the impact of an action on the situation of the CR. As discussed previously, this MoE provides the naive performance of the CR. The direct connection between this MoE and the objective of the mission simplifies the initialization of this MoE. For example in the situation of this proof of concept the objective is minimization of the bandwidth, power, and BER. Thus the initial Effectiveness value is calculated as one minus the average of the normalized distance from the minimum possible value in these three areas, see Equation 4.5. Note, ideal Effectiveness need not represent an achievable state. This MoE is often very similar to the scoring function used with the CR to make decisions. Thus this MoE, while very specific to the situation, provides an estimate of how the CR viewed its situation at some point in the course of operation.

$$Effectiveness = 1 - \frac{1}{3}\left[\frac{bw - bw_{min}}{bw_{max} - bw_{min}} + \frac{pwr - pwr_{min}}{pwr_{max} - pwr_{min}} + \frac{ber - ber_{min}}{ber_{max} - ber_{min}}\right] \quad (4.5)$$

Once heuristics are used to proceed initial MoE values, these and their associated knob and meter form the target outputs and inputs, respectively, for training sets to be applied to the first ANN. The training algorithm used to apply the influence of these training sets will not be able to perfectly match these values. Rather, the training algorithm with have a smoothing affect on these discrete values, forming an interpolation of the first translation function. Smoothing the MoE values in this way helps to spread the weight of the heuristic based values more evenly. This enforces a level consistency for the MoEs and helps to reduce the impact of possible errors in the heuristic application. Note this interpolation is based on the MoE values, produced by the heuristics, that some pure measure for each MoE. Also note that because the ANN are trained over the values of the time series of knobs and meters, the training set encompasses all of the input test points. Such a methodology for training set generation ensures that the ANN will enjoy the benefits of operating in a interpolation capacity.

4.3.2 Modeling the MoP Function

The training for the second ANN is much more straight forward than that for the first ANN. The range of the MoEs is fixed to be zero to one and typically only a small number of MoPs are required. The small and fixed range of the MoEs allows training sets to simply be generated based on expect outcome, with out fear of entering the realm of extrapolation. Generating the training set for the second ANN consists of specifying a few rules to relate the MoPs to the MoE and creating training points by iterating through the possible MoE values while applying these rules.

In order to better understand this process, it is helpful to consider the properties of the MoEs and MoPs. First, note that there are a fixed number of MoEs that each provide an element of the performance information available from the CR. Thus, an alphabet is already available for describing the performance of the CR. Second, consider that the MoP is selected to capture the most interesting aspect of final performance in a mission. This means that the MoPs are selected on a mission-by-mission basis to provide desired information to the elevator. These two factors put the power in the hands of the evaluator, by enabling a set of performance descriptors and the freedom to use them in nearly anyway.

Note that the MoPs exist primarily to provide a succinct summary of the complete performance of the CR. The purpose the MoPs goes beyond simply expressing the naive performance of the CR and represents a more holistic view. This view includes every aspect enabled by the MoEs to show whether or not the CR adheres to all of the side requirements while attacking the main goal of the problem. Whether the MoP can most clearly express this as a scalar value or a vector does not alter this fundamental goal of providing a holistic understanding of mission performance.

To this end, the MoP used for this mission is simply a scalar value that relates this holistic view. The rules used to determine a training set then follow from the meanings of the MoE values. Actions that detract from the Survivability, Stability, or Security of the radio in operation clearly detract from holistic performance. For example if the only negative aspect of a certain training sequence was that Stability and Survivability were both 0.9, the MoP would be 0.8. Actions that are illegal are unusable, and therefore have no performance value. Thus if any MoE is below 0.5, the MoP is zero. Efficiency and Effectiveness both directly contribute to the holistic performance. Thus the average of these two MoEs provide the basis for the MoP. These rules are applied to randomly generated sets of MoE values to build a training set of 1000 points. This training set represents the model of a function for generating an appropriate, holistic performance score.

4.3.3 ANNs Details

With the training data available, the ANNs may be trained. Both ANNs have three layers. The first ANN has a layout of 5 input nodes, 11 hidden nodes, a bias node, and 6 output node, with sigmoidal activation functions in the hidden and output layers. The second ANN has 6 input nodes, 8 hidden nodes, a bias node, and 1 output node, with sigmoidal activation functions in the hidden and output layers. The first ANN is trained using the initial MoEs, as discussed above, for 1000 epochs of the back propagation algorithm. The second ANN is trained with randomized MoE inputs for 100 epochs.

4.4 Evaluation Phase

Once the ANNs have been trained they may be used to determine the MoEs and MoP. Note that the goal of the procedure is to produce a final value for the performance the CR thus after the training is complete the last value in the time series my be input into the first ANN to produce a set of overall MoEs for the test. This set represents the distillation of all of the language and action information from the CR during the test. This set of overall MoEs is then input to the second ANN to produce the final performance score.

The final score provides a relative measure of performance, that has been customized to account for the interest of the evaluator. This score is a consistent, quantitative measure of how the CR under test perform relative to similar CRs in the same scenario. The utilization of the environment and internal language information discussed in previous chapters would extend this score to enable comparisons across wider scenarios or sets of CRs. To display the functionality of the system presented here, two examples are presented below.

4.4.1 Examples of Performance Evaluation

For the purposes of comparison, two variations on the proof of concept evaluation, discussed above, have implemented and executed. The first variation is exactly as described above. The only difference in the second variation is that the FEC coding is fixed at the highest level of protection. This change reduces the search space of the the second CR, but also affects the overall performance. Clearly, the CR with the fixed FEC coding can not perform as well as the original. Thus the final performance score should be lower.

Table 4.1: Performance Scores for Tested CRs

	Free FEC	Fixed FEC
Survivability	0.9816	0.9750
Efficiency	0.8975	0.4467
Stability	0.9822	0.9749
Legality	0.9872	0.6005
Security	0.9820	0.9752
Effectiveness	0.9313	0.9303
Final Performance	0.8368	0.4357

Table 4.1 displays the MoEs and final performance scores for the two tested CRs. From this table it is clear that the CR with free FEC perform better than the CR with fixed FEC, as expected. Specifically, the evaluation reveals that the fixed FEC CR largely suffer from lower efficiency, again, as expected. Also the evaluation reveals that the optimization method employed attempted to use more illegal actions in fixed FEC CR. Thus, the performance system reveals a great deal about the relative performance of the two radios.

Chapter 5

Conclusion and Recommendations for Future Work

The conclusions of this work are clear. An evaluation system based on a holistic consideration of CR allows for a general purpose approach to CR performance evaluation. As part of this approach, consideration of the language used by the CR contributes a great deal to an evaluation system. This benefit arises from the parallelism of the actions and performance of a CR as values that are hidden from an evaluator. The approach to evaluation presented in this work offers clear advantages.

5.1 Summary

This work has identified the need for a widely applicable and consistent performance evaluation approach. Principles of performance evaluation have guided the development of such an approach. An implementation plan for the approach has been constructed. A examination of the approach has detailed the process of evaluation. Finally two example evaluations have been provided as a proof of concept for the evaluation approach. This work has provided the foundation of a much need system for the performance evaluation of a general purpose CR.

5.2 Accomplishments

This work has development an information centric approach for evaluation. The work presented here forms the core of a general purpose performance evaluation system for CR. This work has defined all of the elements of the system and how they fit together. A system for implementing this system has been fully examined. The specific details of the method have been spelled out for a slightly simplified evaluation system. An example evaluation has been stepped through to provide insight to the evaluation process. Finally, two examples were used to show the worth of the evaluation system.

The primary contribution of this work is the development of the core of a performance evaluation system for cognitive radio. This approach is based on a novel consideration of CR's language. Consideration of this language has allowed distilling key performance information from a record of a CR's actions. Applying this language based view to performance evaluation is one of the key factors enabling a general evaluation system. This work has fully outlined a complete performance evaluation system for CR.

5.3 Future Work

While this work has developed a complete core of a performance evaluation system, there is still work left to do. First, all of the details necessary for a full evaluation system have not yet been determined. These additional details would provide extended ability to the evaluation system in terms of the breadth of comparative ability. Second, further study of the heuristics used to provide initial MoE values would provide more insight into the process. Third, adapting the evaluation system to operate in an online mode could potentially provide additional utility to CR operation. These represent the most direct future extensions of the work presented here.

Appendix A

Knob and Meter Time Series

A.1 Free FEC Cognitive Radio

Table A.1: Free FEC Cognitive Radio Knob and Meter Time Series

Action Index	Modulation M	FEC Level	Power (dBm)	Bandwidth (MHz)	BER
1	4	2	-10	9.6899414	2.6562500e-01
2	4	1	-10	6.1962891	6.9335938e-02
3	4	1	-10	6.1962891	1.5820312e-01
4	4	1	-10	6.1962891	1.4550781e-01
5	8	1	0	4.1308594	3.9062500e-03
6	8	1	0	4.1308594	5.8593750e-03
7	8	1	-5	4.1308594	1.5820312e-01
8	8	1	0	4.1308594	1.1718750e-02
9	8	1	0	4.1308594	1.8554688e-02
10	8	1	0	4.1308594	5.3710938e-02
11	8	1	0	4.1308594	8.0078125e-02
12	8	1	0	4.1308594	3.3203125e-02
13	8	1	0	4.1308594	1.7578125e-02
14	8	1	0	4.1308594	3.3203125e-02
15	8	1	-10	4.1308594	2.3535156e-01
16	8	1	-5	4.1308594	1.8847656e-01
17	8	1	-5	4.1308594	6.8359375e-02
18	8	1	-5	4.1308594	2.5097656e-01
19	8	1	0	4.1308594	4.8828125e-03
20	8	1	0	4.1308594	0

A.2 Fixed FEC Cognitive Radio

Table A.2: Fixed FEC Cognitive Radio Knob and Meter Time Series

Action Index	Modulation M	Power (dBm)	Bandwidth (MHz)	BER
1	8	5	9.0087891	0
2	8	5	9.0087891	0
3	8	15	9.0087891	8.7890625e-03
4	8	5	9.0087891	0
5	8	10	9.0087891	0
6	8	-10	9.0087891	1.6601562e-02
7	8	15	9.0087891	0
8	8	10	9.0087891	8.7890625e-03
9	8	5	9.0087891	0
10	8	0	9.0087891	0

Appendix B

Initial MoE Values

B.1 Free FEC Cognitive Radio

Table B.1: Free FEC Cognitive Radio Initial MoE Values

Action Index	Survivability	Efficiency	Stability	Legality	Security	Effectiveness
1	1	0.415190225848	1	1	1	0.741984802834
2	1	0.649286835382	1	1	1	0.923706494736
3	1	0.649286835382	1	1	1	0.864461706736
4	1	0.649286835382	1	1	1	0.872925246736
5	1	0.973930745572	1	1	1	0.997395833333
6	1	0.973930745572	1	1	1	0.99609375
7	1	0.973930295572	1	1	1	0.894531253333
8	1	0.973930745572	1	1	1	0.9921875
9	1	0.973930745572	1	1	1	0.987630208
10	1	0.973930745572	1	1	1	0.964192708
11	1	0.973930745572	1	1	1	0.946614583333
12	1	0.973930745572	1	1	1	0.977864583333
13	1	0.973930745572	1	1	1	0.98828125
14	1	0.973930745572	1	1	1	0.977864583333
15	1	0.973930250572	1	1	1	0.84309896
16	1	0.973930295572	1	1	1	0.87434896
17	1	0.973930295572	1	1	1	0.954427083333
18	1	0.973930295572	1	1	1	0.832682293333
19	1	0.973930745572	1	1	1	0.996744791667
20	1	0.973930745572	1	1	1	1.0

B.2 Fixed FEC Cognitive Radio

Table B.2: Fixed FEC Cognitive Radio Initial MoE Values

Action Index	Survivability	Efficiency	Stability	Legality	Security	Effectiveness
1	1	0.446587650021	1	1	1	1.0
2	1	0.446587650021	1	1	1	1.0
3	1	0.447082525095	1	0	1	0.660807291667
4	1	0.446587650021	1	1	1	1.0
5	1	0.446632648783	1	0	1	1.0
6	1	0.446582655033	1	1	1	0.988932292
7	1	0.447082525095	1	0	1	0.666666666667
8	1	0.446632648783	1	0	1	0.994140625
9	1	0.446587650021	1	1	1	1.0
10	1	0.446583150033	1	1	1	1.0

Appendix C

Evaluation Code

C.1 Free FEC Cognitive Radio

```
#!/usr/bin/env python

from scipy.special import erfc

import random
from numpy import log10

from pybrain.structure import LinearLayer, SigmoidLayer
from pybrain.tools.shortcuts import buildNetwork
from pybrain.supervised.trainers import BackpropTrainer
from pybrain.structure import TanhLayer
from pybrain.datasets import SupervisedDataSet

def eval1():
    ds_moe = SupervisedDataSet(5,6)
    net1 = buildNetwork(5,11,6, bias=True, hiddenclass=SigmoidLayer,
                        outclass=SigmoidLayer)

    dat = init_moe1()
    for k in dat:
        ds_moe.appendLinked(k[0:5],k[5:11])

    trainer1 = BackpropTrainer(net1, ds_moe)
    trainer1.trainEpochs(1000)
```

```
    test = dat[-1][0:5]
    moe = net1.activate(test)

    ds_mop = SupervisedDataSet(6,1)
    net2 = buildNetwork(6,8,1, bias=True, hiddenclass=SigmoidLayer,
                        outclass=SigmoidLayer)

    dat = mop_train()
    for k in dat:
        ds_mop.appendLinked(k[0:-1],k[-1])

    trainer2 = BackpropTrainer(net2, ds_mop)
    trainer2.trainEpochs(100)

    mop = net2.activate(moe)

    print mop

def mop_train():
    n_train = 1000

    train =[]
    for i in range(n_train):
        tmp = []
        for k in range(3):
            tmp.append(random.random())
        tmp.insert(0,1)
        tmp.insert(2,1)
        tmp.insert(4,1)

        if tmp[3]>0.5:
            mop = 0.5*(tmp[1]+tmp[5])
        else:
            mop = 0
        tmp.append(mop)
        train.append(tmp)

    return train

def init_moe1():
    dat = get_data(1)
```

```
    bw_shan = 6.0e6/log10(31)

    bw_min = 4.1308594e+06
    bw_max = 27.027e6

    pwr_min = -10
    pwr_max = 15

    ber_min = 0
    ber_max = 0.5

    for k in range(len(dat)):
        bw = dat[k][3]
        pwr = dat[k][2]
        ber = dat[k][4]

        dat[k][6] = bw_shan/bw-(1-(1+(10**((pwr-30.0)/10.0))**2)**0.5)

        dat[k][7] = 1

        if pwr < 10:
            dat[k][8] = 1
        else:
            dat[k][8] = 0

        dat[k][9] = 1

        dat[k][10]=1-1.0/3.0*((bw-bw_min)/(bw_max-bw_min)
                              + (pwr-pwr_min)/(pwr_max-pwr_min)
                              + (ber-ber_min)/(ber_max-ber_min))

    return dat

def get_data(which):
    if which == 1:
        dat1 = [[4, 2, -10, 9.6899414e+06, 2.6562500e-01, 1, 2, 3, 4, 5, 6], #0
                [4, 1, -10, 6.1962891e+06, 6.9335938e-02, 1, 2, 3, 4, 5, 6], #1
                [4, 1, -10, 6.1962891e+06, 1.5820312e-01, 1, 2, 3, 4, 5, 6], #2
                [4, 1, -10, 6.1962891e+06, 1.4550781e-01, 1, 2, 3, 4, 5, 6], #3
                [8, 1, 0, 4.1308594e+06, 3.9062500e-03, 1, 2, 3, 4, 5, 6],   #4
                [8, 1, 0, 4.1308594e+06, 5.8593750e-03, 1, 2, 3, 4, 5, 6],   #5
                [8, 1, -5, 4.1308594e+06, 1.5820312e-01, 1, 2, 3, 4, 5, 6],  #6
```

```
                [8, 1, 0, 4.1308594e+06, 1.1718750e-02, 1, 2, 3, 4, 5, 6],   #7
                [8, 1, 0, 4.1308594e+06, 1.8554688e-02, 1, 2, 3, 4, 5, 6],   #8
                [8, 1, 0, 4.1308594e+06, 5.3710938e-02, 1, 2, 3, 4, 5, 6],   #9
                [8, 1, 0, 4.1308594e+06, 8.0078125e-02, 1, 2, 3, 4, 5, 6],   #10
                [8, 1, 0, 4.1308594e+06, 3.3203125e-02, 1, 2, 3, 4, 5, 6],   #11
                [8, 1, 0, 4.1308594e+06, 1.7578125e-02, 1, 2, 3, 4, 5, 6],   #12
                [8, 1, 0, 4.1308594e+06, 3.3203125e-02, 1, 2, 3, 4, 5, 6],   #13
                [8, 1, -10, 4.1308594e+06, 2.3535156e-01, 1, 2, 3, 4, 5, 6], #14
                [8, 1, -5, 4.1308594e+06, 1.8847656e-01, 1, 2, 3, 4, 5, 6],  #15
                [8, 1, -5, 4.1308594e+06, 6.8359375e-02, 1, 2, 3, 4, 5, 6],  #16
                [8, 1, -5, 4.1308594e+06, 2.5097656e-01, 1, 2, 3, 4, 5, 6],  #17
                [8, 1, 0, 4.1308594e+06, 4.8828125e-03, 1, 2, 3, 4, 5, 6],   #18
                [8, 1, 0, 4.1308594e+06, 0, 1, 2, 3, 4, 5, 6]]               #19

        return dat1
    elif which == 2:
        dat2 = [[8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],               #0
                [8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],               #1
                [8, 15, 9.0087891e+06, 8.7890625e-03, 1, 2, 3, 4, 5, 6],  #2
                [8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],               #3
                [8, 10, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],              #4
                [8, -10, 9.0087891e+06, 1.6601562e-02, 1, 2, 3, 4, 5, 6], #5
                [8, 15, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],              #6
                [8, 10, 9.0087891e+06, 8.7890625e-03, 1, 2, 3, 4, 5, 6],  #7
                [8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],               #8
                [8, 0, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6]]               #9

        return dat2

if __name__ == '__main__':
    eval1()
```

C.2 Fixed FEC Cognitive Radio

```
#!/usr/bin/env python

from scipy.special import erfc

import random
from numpy import log10
```

```
from pybrain.structure import LinearLayer, SigmoidLayer
from pybrain.tools.shortcuts import buildNetwork
from pybrain.supervised.trainers import BackpropTrainer
from pybrain.structure import TanhLayer
from pybrain.datasets import SupervisedDataSet

def eval2():
    ds_moe = SupervisedDataSet(4,6)
    net1 = buildNetwork(4,11,6, bias=True, hiddenclass=SigmoidLayer,
                        outclass=SigmoidLayer)

    dat = init_moe2()
    for k in dat:
        ds_moe.appendLinked(k[0:4],k[4:10])

    trainer1 = BackpropTrainer(net1, ds_moe)
    trainer1.trainEpochs(1000)

    test = dat[-1][0:4]
    moe = net1.activate(test)

    print moe

    ds_mop = SupervisedDataSet(6,1)
    net2 = buildNetwork(6,8,1, bias=True, hiddenclass=SigmoidLayer,
                        outclass=SigmoidLayer)

    dat = mop_train()
    for k in dat:
        ds_mop.appendLinked(k[0:-1],k[-1])

    trainer2 = BackpropTrainer(net2, ds_mop)
    trainer2.trainEpochs(100)

    mop = net2.activate(moe)

    print mop

def mop_train():
    n_train = 1000

    train =[]
```

```
    for i in range(n_train):
        tmp = []
        for k in range(3):
            tmp.append(random.random())
        tmp.insert(0,1)
        tmp.insert(2,1)
        tmp.insert(4,1)

        if tmp[3]>0.5:
            mop = 0.5*(tmp[1]+tmp[5])
        else:
            mop = 0
        tmp.append(mop)
        train.append(tmp)

    return train

def init_moe2():
    dat = get_data(2)

    bw_shan = 6.0e6/log10(31)

    bw_min = 9.0087891e+06
    bw_max = 27.027e6

    pwr_min = -10
    pwr_max = 15

    ber_min = 0
    ber_max = 0.5

    for k in range(len(dat)):
        bw = dat[k][2]
        pwr = dat[k][1]
        ber = dat[k][3]

        dat[k][5] = bw_shan/bw-(1-(1+(10**((pwr-30.0)/10.0))**2)**0.5)

        dat[k][6] = 1

        if pwr < 10:
            dat[k][7] = 1
```

```
        else:
            dat[k][7] = 0

        dat[k][8] = 1

        dat[k][9]=1-1.0/3.0*((bw-bw_min)/(bw_max-bw_min)
                              + (pwr-pwr_min)/(pwr_max-pwr_min)
                              + (ber-ber_min)/(ber_max-ber_min))

    return dat

def get_data(which):
    if which == 1:
        dat1 = [[4, 2, -10, 9.6899414e+06, 2.6562500e-01, 1, 2, 3, 4, 5, 6], #0
                [4, 1, -10, 6.1962891e+06, 6.9335938e-02, 1, 2, 3, 4, 5, 6], #1
                [4, 1, -10, 6.1962891e+06, 1.5820312e-01, 1, 2, 3, 4, 5, 6], #2
                [4, 1, -10, 6.1962891e+06, 1.4550781e-01, 1, 2, 3, 4, 5, 6], #3
                [8, 1, 0, 4.1308594e+06, 3.9062500e-03, 1, 2, 3, 4, 5, 6],   #4
                [8, 1, 0, 4.1308594e+06, 5.8593750e-03, 1, 2, 3, 4, 5, 6],   #5
                [8, 1, -5, 4.1308594e+06, 1.5820312e-01, 1, 2, 3, 4, 5, 6],  #6
                [8, 1, 0, 4.1308594e+06, 1.1718750e-02, 1, 2, 3, 4, 5, 6],   #7
                [8, 1, 0, 4.1308594e+06, 1.8554688e-02, 1, 2, 3, 4, 5, 6],   #8
                [8, 1, 0, 4.1308594e+06, 5.3710938e-02, 1, 2, 3, 4, 5, 6],   #9
                [8, 1, 0, 4.1308594e+06, 8.0078125e-02, 1, 2, 3, 4, 5, 6],   #10
                [8, 1, 0, 4.1308594e+06, 3.3203125e-02, 1, 2, 3, 4, 5, 6],   #11
                [8, 1, 0, 4.1308594e+06, 1.7578125e-02, 1, 2, 3, 4, 5, 6],   #12
                [8, 1, 0, 4.1308594e+06, 3.3203125e-02, 1, 2, 3, 4, 5, 6],   #13
                [8, 1, -10, 4.1308594e+06, 2.3535156e-01, 1, 2, 3, 4, 5, 6], #14
                [8, 1, -5, 4.1308594e+06, 1.8847656e-01, 1, 2, 3, 4, 5, 6],  #15
                [8, 1, -5, 4.1308594e+06, 6.8359375e-02, 1, 2, 3, 4, 5, 6],  #16
                [8, 1, -5, 4.1308594e+06, 2.5097656e-01, 1, 2, 3, 4, 5, 6],  #17
                [8, 1, 0, 4.1308594e+06, 4.8828125e-03, 1, 2, 3, 4, 5, 6],   #18
                [8, 1, 0, 4.1308594e+06, 0, 1, 2, 3, 4, 5, 6]]               #19

        return dat1
    elif which == 2:
        dat2 = [[8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],              #0
                [8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],              #1
                [8, 15, 9.0087891e+06, 8.7890625e-03, 1, 2, 3, 4, 5, 6],  #2
                [8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],              #3
                [8, 10, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],             #4
                [8, -10, 9.0087891e+06, 1.6601562e-02, 1, 2, 3, 4, 5, 6], #5
```

```
                [8, 15, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],                #6
                [8, 10, 9.0087891e+06, 8.7890625e-03, 1, 2, 3, 4, 5, 6],  #7
                [8, 5, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6],                 #8
                [8, 0, 9.0087891e+06, 0, 1, 2, 3, 4, 5, 6]]                 #9

        return dat2

if __name__ == '__main__':
    eval2()
```

Appendix D

Cognitive Radio Matlab Code

D.1 Free FEC Cognitive Radio

Controller

```
function csere1()

pop = opt();
n_pop = size(pop,1);

n_knobs = 3;

count_out = 50;
ber_thres = 0;
bw_thres = 10e6;
pwr_thres = 0;

cur_ber = 99;
cur_bw = 99;
cur_pwr = 99;

good = 0;
counter = 0;
while not(good);

  aug_pop = pop;
  for p=1:1:n_pop
    act = pop(p,:);
```

```
    a = size(act);
    ber = kb(act);
    aug_pop(p,n_knobs+1)=ber;
  end

  [max_score, best]=rank(aug_pop);

  [bw, ber] = radio(best);

  kb(best,ber);

  cur_ber = ber;
  cur_pwr = best(n_knobs);
  cur_bw = bw;

  record = [best bw ber];
  save('csere1.dat','record','-ASCII','-append')
  counter = counter +1;

  if cur_ber <= ber_thres
    if cur_pwr <= pwr_thres
      if cur_bw <= bw_thres
disp('answer found')
disp(counter)
good = 1;
      end
    end
  end

  if counter > count_out
    good =1;
    disp('count out')
  end

  if not(good)
    pop=opt(best);
  end

end
```

Optimizer

```
function pop=opt(seed)

if nargin<1
  seed = -1;
end

n_pop = 15;
n_knobs = 3;
knob_lev = [3 3 6];
knobs = [2 4 8 0 0 0;
 1 2 3 0 0 0;
 -10 -5 0 5 10 15];

pop = zeros(n_pop, n_knobs);

if seed == -1
  % Random Pop
  for p=1:1:n_pop
    for q=1:1:n_knobs
      top = knob_lev(q);
      idx = randi([1 top],1);
      pop(p,q) = knobs(q,idx);
    end
  end
else
  for p=1:1:n_pop
    direct = n_knobs-mod(p,n_knobs);
    for q=1:1:n_knobs
      if q==direct
pop(p,q)=seed(q);
      else
top = knob_lev(q);
idx = randi([1 top],1);
pop(p,q) = knobs(q,idx);
      end
    end
  end
end
```

Ranker

```
function [max_score, best]=rank(aug_pop)
```

```
max_score = -99;
max_idx = -1;

n_pop = size(aug_pop,1);

for p=1:1:n_pop
  mod_sc = aug_pop(p,1);
  rate_sc = aug_pop(p,2);
  pwr_sc = aug_pop(p,3);
  ber_sc = aug_pop(p,4);

  mod_sc = log2(mod_sc);
  rate_sc = -1*rate_sc+4;
  pwr_sc = 4 + pwr_sc/(-5);
  ber_sc = round(-11*ber_sc+6);
  score = sum([mod_sc rate_sc pwr_sc ber_sc]);
  if score > max_score
    max_score = score;
    max_idx = p;
  end
end

best = aug_pop(max_idx,1:3);
```

Radio

```
function  [bw, ber] = radio(act)

M = act(1);
rate = act(2);
pwr = act(3);

% Source
x = randi([0,1],1024,1);

% Encoder
switch rate
 case 1
  n = 15;
  k = 11;
 case 2
```

```
  n = 15;
  k = 7;
 case 3
  n = 15;
  k = 5;
end
len_x = size(x,1);
num_pad = k-mod(len_x,k);
code = [x;zeros(num_pad,1)];
enc = fec.bchenc(n,k);
code = encode(enc,code);

len_code = size(code,1);
eff = len_x/len_code;

% Modulation
mod_len = log2(M);
mod_pad = mod_len - mod(len_code,mod_len);
code=[code;zeros(mod_pad,1)];
h = modem.dpskmod('M',M, 'InputType','bit');
y = modulate(h,code);

% Channel
n_pwr = -15+2*randn(1);
snr = pwr - n_pwr;
rx = awgn(y,snr);

% Demodulation
h = modem.dpskdemod('M',M, 'OutputType','bit');
z = demodulate(h,rx);

z = z(1:len_code);

dec = fec.bchdec(n,k);
uncode = decode(dec,z);

uncode = uncode(1:len_x);

bw = 1/log2(M)*1/eff*6e6*(1.5);

bad_count = nnz(xor(x,uncode));
ber = bad_count/len_x;
```

Knowledge Base

```
function ber=kb(act,adber)

if nargin<2
  adber = -1;
end

filename = 'kb.dat';
act_size = size(act,2);
row = -1;
% Load Database
if exist(filename)
  dat_base=load(filename,'-ASCII');

  act_base = dat_base(:,1:act_size);
  base_size = size(act_base,1);
  % Find Row
  finder = repmat(act, base_size,1);
  finder = act_base-finder;
  finder = sum(finder,2);
  row = find(finder==0);
  if size(row,1)==0
    row = -1;
  end
else
  dat_base = -1;
end

if adber~=-1
  %In
  sav_act = act;
  sav_act(act_size+1)=adber;
  if dat_base ==-1;
    dat_base =[sav_act];
  else
    if row ==-1
      dat_base = [dat_base;sav_act];
    else
      old_adber = dat_base(row,act_size+1);
      new_adber = mean([old_adber,adber]);
      dat_base(row,act_size+1)=new_adber;
```

```
    end
  end
  save(filename,'dat_base', '-ASCII');
  ber =-1;
else
  %out
  if dat_base == -1
    ber = 0.25;
  else
    if row == -1
      ber = 0.25;
    else
      ber = dat_base(row,act_size+1);
    end
  end
end
```

D.2 Fixed FEC Cognitive Radio

Controller

```
function csere2()

pop = opt2();
n_pop = size(pop,1);

n_knobs = 2;

count_out = 50;
ber_thres = 0;
bw_thres = 10e6;
pwr_thres = 0;

cur_ber = 99;
cur_bw = 99;
cur_pwr = 99;

good = 0;
counter = 0;
while not(good);
```

```
  aug_pop = pop;
  for p=1:1:n_pop
    act = pop(p,:);
    a = size(act);
    ber = kb(act);
    aug_pop(p,n_knobs+1)=ber;
  end

  [max_score, best]=rank2(aug_pop);

  [bw, ber] = radio2(best);

  kb(best,ber);

  cur_ber = ber;
  cur_pwr = best(n_knobs);
  cur_bw = bw;

  record = [best bw ber];
  save('csere2.dat','record','-ASCII','-append')
  counter = counter +1;

  if cur_ber <= ber_thres
    if cur_pwr <= pwr_thres
      if cur_bw <= bw_thres
disp('answer found')
disp(counter)
good = 1;
      end
    end
  end

  if counter > count_out
    good =1;
    disp('count out')
  end

  if not(good)
    pop=opt2(best);
  end
end
```

Optimizer

```
function pop=opt2(seed)

if nargin<1
  seed = -1;
end

n_pop = 1;
n_knobs = 2;
knob_lev = [3 6];
knobs = [2 4 8 0 0 0;
 -10 -5 0 5 10 15];

pop = zeros(n_pop, n_knobs);

if seed == -1
  % Random Pop
  for p=1:1:n_pop
    for q=1:1:n_knobs
      top = knob_lev(q);
      idx = randi([1 top],1);
      pop(p,q) = knobs(q,idx);
    end
  end
else
  for p=1:1:n_pop
    direct = n_knobs-mod(p,n_knobs);
    for q=1:1:n_knobs
      if q==direct
pop(p,q)=seed(q);
      else
top = knob_lev(q);
idx = randi([1 top],1);
pop(p,q) = knobs(q,idx);
      end
    end
  end
end
```

Ranker

```
function [max_score, best]=rank2(aug_pop)

max_score = -99;
max_idx = -1;

n_pop = size(aug_pop,1);

for p=1:1:n_pop
  mod_sc = aug_pop(p,1);
  pwr_sc = aug_pop(p,2);
  ber_sc = aug_pop(p,3);

  mod_sc = 2*log2(mod_sc);
  pwr_sc = 4 + pwr_sc/(-5);
  ber_sc = round(-11*ber_sc+6);
  score = sum([mod_sc pwr_sc ber_sc]);
  if score > max_score
    max_score = score;
    max_idx = p;
  end
end

best = aug_pop(max_idx,1:3);
```

Radio

```
function  [bw, ber] = radio2(act)

M = act(1);
rate = 3;
pwr = act(3);

% Source
x = randi([0,1],1024,1);

% Encoder
switch rate
 case 1
  n = 15;
  k = 11;
 case 2
  n = 15;
```

```
  k = 7;
 case 3
  n = 15;
  k = 5;
end
len_x = size(x,1);
num_pad = k-mod(len_x,k);
code = [x;zeros(num_pad,1)];
enc = fec.bchenc(n,k);
code = encode(enc,code);

len_code = size(code,1);
eff = len_x/len_code;

% Modulation
mod_len = log2(M);
mod_pad = mod_len - mod(len_code,mod_len);
code=[code;zeros(mod_pad,1)];
h = modem.dpskmod('M',M, 'InputType','bit');
y = modulate(h,code);

% Channel
n_pwr = -15+2*randn(1);
snr = pwr - n_pwr;
rx = awgn(y,snr);

% Demodulation
h = modem.dpskdemod('M',M, 'OutputType','bit');
z = demodulate(h,rx);

z = z(1:len_code);

dec = fec.bchdec(n,k);
uncode = decode(dec,z);

uncode = uncode(1:len_x);

bw = 1/log2(M)*1/eff*6e6*(1.5);

bad_count = nnz(xor(x,uncode));
ber = bad_count/len_x;
```

Knowledge Base   Same as above

Bibliography

[1] J. Mitola, III, “Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio,” Ph.D. dissertation, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.

[2] C. J. Rieser *et al.*, “Cognitive Radio Engine Based on Genetic Algorithms in a Network,” U.S. Patent 7 289 972, Oct. 30, 2007.

[3] Y. Zhao *et al.*, “Performance Evaluation of Cognitive Radios: Metrics, Utility Functions, and Methodology,” *Proceedings of the IEEE*, vol. 97, no. 4, pp. 642 –659, Apr. 2009.

[4] M. Gadhiok *et al.*, “Metacognition: Enhancing the performance of a cognitive radio,” in *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, Feb. 2011, pp. 198 –203.

[5] A. Batra *et al.*, “Cognitive Radios for Unlicensed WANs,” in *BWRC Cognitive Radio Workshop*, 2004.

[6] Y. Zhao *et al.*, *Cognitive Radio Technology.* Academic Press, 2006, ch. Network Support: The Radio Envrionment Map.

[7] ——, “Performance Evaluation of Radio Environment Map-Enabled Cognitive Spectrum-Sharing Networks,” in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, Oct. 2007, pp. 1 –7.

[8] F. Bouali *et al.*, “Strengthening Radio Environment Maps with Primary-User Statistical Patterns for Enhancing Cognitive Radio Operation,” in *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2011 Sixth International ICST Conference on*, Jun. 2011, pp. 256 –260.

[9] Y. Zhao *et al.*, “Development of Radio Environment Map Enabled Case- and Knowledge-Based Learning Algorithms for IEEE 802.22 WRAN Cognitive Engines,” in *Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007. 2nd International Conference on*, Aug. 2007, pp. 44 –49.

[10] N. Chomsky, *Reflections on Language.* Pantheon, 1975.

[11] H. Hoijer, *The Sapir-Whorf Hypothesis.* University of Chicago Press, 1954, no. 1929, pp. 1–8.

[12] T. Willink and H. Rutagemwa, "Framework for Performance Evaluation of Cognitive Radio Networks in Heterogeneous Environments," in *Electrical and Computer Engineering, 2009. CCECE '09. Canadian Conference on*, May 2009, pp. 199 –203.

[13] N. Goldenfeld and L. P. Kadanoff, "Simple Lessons from Complexity," *Science*, vol. 284, no. 5411, pp. 87–89, 1999. [Online]. Available: http://www.sciencemag.org/content/284/5411/87.abstract

[14] N. Sproles, "Formulating Measures of Effectiveness," *Systems Engineering*, vol. 5, no. 4, pp. 253–263, 2002. [Online]. Available: http://dx.doi.org/10.1002/sys.10028

[15] W. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biology*, vol. 5, pp. 115–133, 1943, 10.1007/BF02478259. [Online]. Available: http://dx.doi.org/10.1007/BF02478259

[16] R. O. Duda *et al.*, *Pattern Classification.* John Wiley & Sons, Inc, 2001.

[17] A. He *et al.*, "A Survey of Artificial Intelligence for Cognitive Radios," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 4, pp. 1578 –1592, May 2010.

[18] H. Burke *et al.*, "Comparing Artificial Neural Networks to Other Statistical Methods for Medical Outcome Prediction," in *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*, vol. 4, Jun. 1994, pp. 2213 –2216 vol.4.

[19] A. Fehske *et al.*, "A New Approach to Signal Classification using Spectral Correlation and Neural Networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, Nov. 2005, pp. 144 –150.

[20] B. Ustundag and O. Orcay, "Pattern Based Encoding for Cognitive Communication," in *Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on*, May 2008, pp. 1 –6.

[21] S. Kang *et al.*, "Detecting Identity-Spoof Attack Based on BP Network in Cognitive Radio Network," in *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011*, vol. 2, Jul. 2011, pp. 1603 –1606.

[22] Z. Xiyun *et al.*, "The Application of Fuzzy Neural Network in Cognitive Radio Network," in *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, Aug. 2011, pp. 7310 –7316.

[23] V. Tumuluru *et al.*, "A Neural Network Based Spectrum Prediction Scheme for Cognitive Radio," in *Communications (ICC), 2010 IEEE International Conference on*, May 2010, pp. 1 –5.

[24] N. Baldo and M. Zorzi, "Learning and Adaptation in Cognitive Radios Using Neural Networks," in *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, Jan. 2008, pp. 998 –1003.

[25] Z. Zhang and X. Xie, "Intelligent Cognitive Radio: Research on Learning and Evaluation of CR Based on Neural Network," in *Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference on*, Dec. 2007, pp. 33 –37.

[26] M. Serra-Ricart *et al.*, "Multidimensional Interpolation Using Artifical Neural Networks: Application to an H_1 Cloud in Perseus," *The Astronomical Journal*, vol. 109, no. 1, pp. 312–318, Jan. 1995.

[27] E. Barnard and L. Wessels, "Extrapolation and Interpolation in Neural Network Classifiers," *Control Systems, IEEE*, vol. 12, no. 5, pp. 50 –53, Oct. 1992.

[28] T. Schaul *et al.*, "PyBrain," *J. Mach. Learn. Res.*, vol. 11, pp. 743–746, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756030

[29] P. Marshall, *Quantitative Analysis of Cognitive Radio and Network Performance.* Artech House, 2010.