Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking

by
Christian James Rieser

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

Doctor of Philosophy
in
Electrical Engineering

APPROVED:

Dr. C. W. Bostian, Chairman	Dr. S. F. Midkiff
Dr. T. L. Martin	Dr. G. E. Morgan
Dr. D. G. Sweeney	Dr. B. D. Woerner

August 2004
Blacksburg, Virginia

Keywords: Cognitive Radio, Biologically Inspired, Distributed, Genetic Algorithms, Secure, Robust, Communications, Networking, Rapidly Deployable, Disaster Response

Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking

Christian James Rieser

Abstract

This research focuses on developing a cognitive radio that could operate reliably in unforeseen communications environments like those faced by the disaster and emergency response communities. Cognitive radios may also offer the potential to open up secondary or complimentary spectrum markets, effectively easing the perceived spectrum crunch while providing new competitive wireless services to the consumer. A structure and process for embedding cognition in a radio is presented, including discussion of how the mechanism was derived from the human learning process and mapped to a mathematical formalism called the BioCR. Results from the implementation and testing of the model in a hardware test bed and simulation test bench are presented, with a focus on rapidly deployable disaster communications. Research contributions include developing a biologically inspired model of cognition in a radio architecture, proposing that genetic algorithm operations could be used to realize this model, developing an algorithmic framework to realize the cognition mechanism, developing a cognitive radio simulation toolset for evaluating the behavior the cognitive engine, and using this toolset to analyze the cognitive engine's performance in different operational scenarios. Specifically, this research proposes and details how the chaotic meta-knowledge search, optimization, and machine learning properties of distributed genetic algorithm operations could be used to map this model to a computable mathematical framework in conjunction with dynamic multi-stage distributed memories. The system formalism is contrasted with existing cognitive radio approaches, including traditionally brittle artificial intelligence approaches. The cognitive engine architecture and algorithmic framework is developed and introduced, including the Wireless Channel Genetic Algorithm (WCGA), Wireless

System Genetic Algorithm (WSGA), and Cognitive System Monitor (CSM). Experimental results show that the cognitive engine finds the best tradeoff between a host radio's operational parameters in changing wireless conditions, while the baseline adaptive controller only increases or decreases its data rate based on a threshold, often wasting usable bandwidth or excess power when it is not needed due its inability to learn. Limitations of this approach include some situations where the engine did not respond properly due to sensitivity in algorithm parameters, exhibiting ghosting of answers, bouncing back and forth between solutions. Future research could be pursued to probe the limits of the engine's operation and investigate opportunities for improvement, including how best to configure the genetic algorithms and engine mathematics to avoid engine solution errors. Future research also could include extending the cognitive engine to a cognitive radio network and investigating implications for secure communications.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The need for secure and robust communications is becoming more apparent every day. Wireless services are becoming largely ubiquitous throughout the nation, although still expensive. The explosion of IEEE 802.11 B/G/A wireless data and voice over IP networks, often called Wi-Fi for "Wireless Fidelity," has shown that for the "last mile" connection to a consumer, affordable broadband wireless is the preferred method of delivering bits from a fiber, cable, or satellite to your favorite digital computing pod, whether those bits represent voice, video, or data. The dependence on digital wireless communications technology has led to two major results: forecasts that the nation's wireless spectrum is dwindling and concerns that the wireless systems of today are not adequately robust and secure when emergency events like the September 11, 2001 attacks occur.

Virginia Tech has been pursing research and development of rapidly deployable broadband wireless systems for disaster response communications the past half a decade [1]. Following September 11, 2001 it became evident that there was a need for self healing wireless networks and radios that could autonomously and legally evolve in time to meet the needs of the nation's communications. At that time wireless technologies were emerging that allowed radios to adapt their behavior based on pre-calculated algorithms, a significant advance beyond the fixed radios of the past which had their operational parameters set at the time of manufacture [2]. Unfortunately when faced with unanticipated scenarios and electromagnetic environments, these adaptive radios often failed to function properly or experienced severe performance degradation [3]. As work progressed on highly flexible programmable radios based on software defined radio technology, the idea of a radio that could evolve its capabilities in time and space began to appear as a plausible concept [4]. A radio that would operate reliably in unforeseen communications environments and potentially open up secondary or complimentary spectrum markets could effectively ease the perceived spectrum crunch while providing new competitive wireless services to the consumer [5].

Such cognitive radios, a term first coined by Joseph Mitola III [5][6], have become a topic of great research interest in the past few years. Many cognitive radio researchers in government and industry have adopted the Oxford English Dictionary (OED) definition of "cognitive" as "pertaining to cognition, or to the action or process of knowing," and "cognition" is defined as "the action or faculty of knowing taken in its widest sense, including sensation, perception, conception, etc., as distinguished from feeling and volition". Given this definition, the process of sensing an existing wireless channel, evolving a radio's operation to accommodate the perceived wireless channel, and evaluating what happens is appropriately described as a cognitive process. This approach includes both awareness of the wireless channel and judgment of the best possible action to take given this knowledge. The convergence of research sponsored through the Defense Advanced Research Projects Agency (DARPA) NeXt Generation (XG) wireless program [7], governmental support of cognitive radio through the Federal Communications Commission's (FCC) Notice of Proposed Rulemaking (ET Docket No. 03-108) [8], and the upcoming National Science Foundation (NSF) Research in Networking Technology and Systems (NetS) programmable wireless networking program [9] point to an exciting next few years for cognitive radio researchers. This dissertation discusses these and other advances, including a vision for cognitive radio moving forward.

1.1 Summary of the Evolution of Cognitive Radios

It has been said that a picture is worth a thousand words. Figure 1.1 provides a brief timeline of advances in cognitive radio research and serves as a functional description of the enabling technology transitions that have occurred since Mitola introduced the term cognitive radio in 1999. Mitola's CR-1 cognitive radio prototype [10][11] modeled a context and location based cognition cycle at an application layer. His research pointed to the potential use of cognitive radio technology to enable spectrum rental applications and create secondary wireless access markets [12].

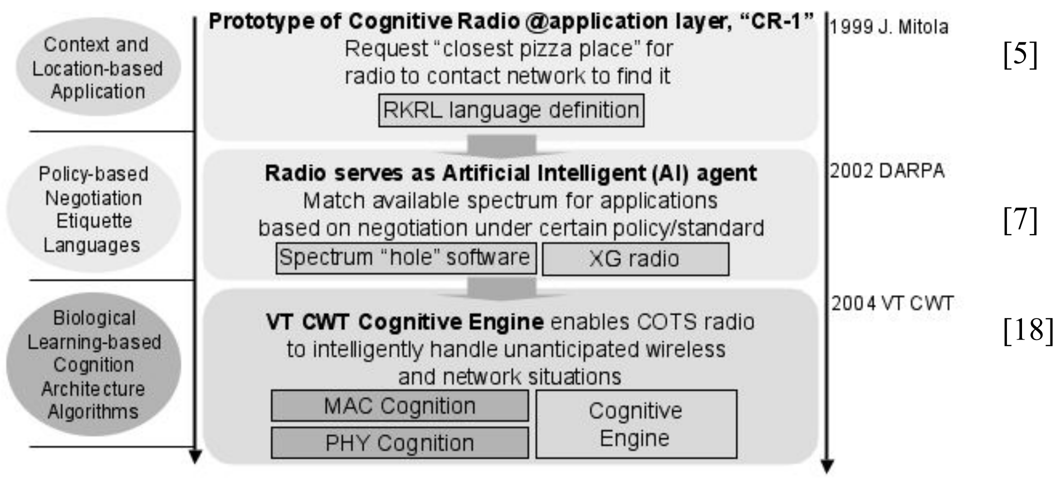


Figure 1.1: Cognitive radio roadmap and functional evolution

Recognizing that wireless systems underutilize spectrum, in 2002 DARPA funded the XG program [13] to create adaptive radios that sense and share use of the spectrum, with a focus on policy-based negotiation and radio etiquettes which leverage spectrum "holes" that open in space and time. These XG radios did not have cognitive learning and evolvable operation capabilities like those proposed by Mitola but could serve as potential hosts for cognitive wireless functions. The excitement around the XG program reinforced momentum building at the FCC, whose policy makers were completing a study that they felt showed the nation's wireless spectrum was underutilized in time and space [14].

The summer of 2003 proved to be a turning point for the cognitive radio concept. After the FCC's Spectrum Policy Task Force (SPTF) issued their fall 2002 report [15] and the FCC Office of Engineering Technology hosted a cognitive radio day, what followed sent shockwaves through the wireless industry. The FCC issued NPRM 03-108 on cognitive radio [16] that opened up a broad dialogue on what cognitive radio is and what, if any, rules the FCC should impose on the fledgling technology. The FCC's stated aim of the cognitive radio dialogue was to explore whether cognitive radio could open up competitive new wireless services through secondary or cooperative spectrum markets.

The FCC spoke of "low hanging fruit," or spectrum that could be re-assigned in time and space to open up new competitive wireless service offerings [17]. The FCC was particularly focused on providing broadband wireless to underserved markets. Interestingly, the FCC's definition of an underserved communications market was based on the observed spectrum utilization, not whether the region was urban or rural. This definition was clearly aimed at encouraging innovative use of spectrum that they felt lay fallow and underutilized.

I traveled to Washington, DC and attended numerous FCC hearings on cognitive radio as part of my research efforts. I heard definitions of cognitive radio that ranged from wireless systems that could switch between wireless profiles stored in a central database to talk of scanning receivers that interacted with elements of an artificial intelligence based expert system. These definitions mostly assumed a state of the art software defined radio as a host, implying that legacy data and voice broadband wireless systems used by the disaster communications community could not be made cognitive without expensive upgrades to infrastructure. My research the past five years has focused on rapidly deployable broadband wireless communications for disaster response, so with the encouragement and direction of my advisors, I focused my research interest in cognitive radio on creating a cognitive radio model and proof of concept for disaster communications systems.

In 2004 our cognitive radio research team at Virginia Tech demonstrated a biologically inspired cognitive engine based on genetic algorithms (GAs) that is capable of learning and intelligently evolving a radio's PHY and MAC behavior in the face of unanticipated wireless and network situations [18][19]. Our cognitive engine can be embedded in the XG agent and radio technology, with the policy and etiquette aware agents serving as wrappers that enable communication between cognitive radio communities and the adaptive and spectrum aware radios serving as host platforms.

1.2 Problem Statement, History, and Contributions

This dissertation summarizes the doctoral research I pursued that led to a cognitive radio engine model and implementation in a hardware test bed and simulation test bench, with a focus on rapidly deployable disaster communications. My specific research contributions included developing a biologically inspired model of cognition in a radio and proposing that the chaotic meta-knowledge processing and optimization properties of distributed genetic algorithms could be used to map this model to a computable mathematical framework which included multi-stage distributed memories. This dissertation presents that work and describes my contribution to the implementation of that formalism in the cognitive radio research toolset developed in conjunction with our research team.

A key research question that I set out to answer in my Ph.D. research was how to develop an appropriate structure and process for embedding cognition in a radio. What cognitive model should be used? Which host radio architecture? Which radio layer? The resulting architecture and algorithmic framework serve as the cornerstone of what I have labeled the "BioCR" formalism, with "Bio" standing for "biologically inspired" and "CR" standing for "cognitive radio."

My research included work developing and testing this formalism using a cognitive radio toolset. The toolset included a software simulation test bench and hardware test bed, both which could host the software implementation of the cognitive engine developed by our cognitive radio team that included myself, Dr. Charles Bostian, and graduate student colleagues Tim Gallagher and Tom Rondeau. Realizing these ideas in a real world test would not have been possible without each of their contributions.

Dr. Bostian encouraged our team to proceed with orthogonal, but interrelated problems. Tim Gallagher and my Ph.D. research were two sides of the same coin. Tim Gallagher's research focused on quantifying wireless "paths of opportunity" for emergency communication, in which he developed an algorithm that could convert an impulse response to a bit error rate (BER) curve. My research focused using such channel metrics to control and evolve a radio in unknown wireless channels, in which I developed the BioCR formalism which included an algorithmic framework and model architecture. Tom Rondeau's research focuses on extending the BioCR formalism to a cognitive wireless network, with results still to come.

My specific research contributions included proposal and development of the BioCR behavioral model, process framework, input/output architecture, procedural algorithmic framework, and experimental application simulator which served as the baseline my comparative experiments between the cognitive engine and traditional adaptive controller. Gallagher's specific contributions to this research included measurements of 28 GHz diffuse scattering, developing an algorithm that mapped a channel response to a BER curve, applying this technique to set equalizer taps, and participation in a research dialogue while I was developing the BioCR formalism. Rondeau's specific contributions to this research included implementation of the cognitive engine code and hardware test bed, design and implementation of the WSGA algorithm code, joint design and implementation of the WCGA algorithm code, implementation of the CSM algorithm code, and participation in a research dialogue while I was developing the BioCR formalism.

My thanks extend to each of them for their contributions to this research and to Dr. Bostian for his steady guidance and input – our research team made amazing process the past year or so.

Current wireless communication systems can be described as either *fixed* where the radio's technical characteristics are set at the time of manufacture, or *adaptive,* where the radio can respond to channel conditions that represent one of a finite set of anticipated events. Researchers like Mitola have postulated that cognitive radios could be used to enable intelligent wireless networks that evolve in time, but very few cognitive radios have been implemented. Due to my focus on disaster communications technology during my graduate research, I decided to concentrate my cognitive radio research on how to create a cognitive radio model and framework that could respond intelligently to an unanticipated series of events; i.e. learning to configure itself to optimize its operation in wireless channels that it has never encountered before.

My research addressed a number of specific problems. When I began this research back in 2001, very little information existed about actual real world cognitive radio architectures or host radio platforms. Mitola's work assumed a software radio platform, technology that was not yet available or affordable to the disaster communications community who were still using legacy radios with minimal programmability.

One key research question that I explored was whether a legacy radio could host a cognitive engine, and how the engine would interact with the system. I adopted the idea of treating the radio as a vector of parameters, with inputs labeled as "knobs" and outputs labeled as "meters." This concept was also emerging in the software radio community, championed by Friedrich K. Jondral of Universität Karlsruhe (TH) Germany in his talk describing "parameterized software radio."

A second major research question I explored was on which communications layer should I focus my cognitive radio engine model research? Existing software defined radio (SDR) research focused on the application (APP) layer, but I sensed that a new frontier lay ahead with agile radios which would require development of a cognitive radio engine that was focused on the physical (PHY) and medium access control (MAC) layers. Many researchers discussing cognitive radio had assumed a cognitive model based on an expert system accessing a database of radio profiles, a pure case-based system without learning capability. I decided that this central database concept would not be practical for disaster response applications, which inherently required a cognitive model that could autonomously learn without expert input or maintenance, since most of the cases that exist in a disaster event would be new and therefore may not be present in the existing memory of the expert system. Other researchers like Mitola viewed the cognitive engine as a mechanism that operated at the radio's application layer, serving as a task manager that could learn the users computing needs and then respond with the appropriate radio profile. Again, I felt that the disaster communications community needed cognitive radio communication links that could support such applications, but really the need was to develop broadband wireless links that were self-healing in the changing environments often observed in catastrophic situations.

A third major research question I explored was what model of cognition should be used and how could it be implemented in a computing environment. What systems in the world today are good examples of self-healing learning systems? What algorithms serve as the foundation for these systems? I was struck by how fragile computing systems were and how robust biologically systems were, including insects, animals, and humans. These organisms were capable of evolving to meet new challenges. After some interesting discussions with cognitive development researchers Dr. Cosby Rogers and Dr. Janet Sawyers, I discovered that the cognitive development process of children through creative play mimicked the self-healing learning behavior I wanted to embed in my cognitive radio model for disaster communications. Most of these mathematical models of play utilized neural networks as their basis, but these lacked evolutionary capabilities to learn to adapt to unforeseen scenarios.

I chose to focus my research on mapping this creative and chaotic learning mechanism to some form of mathematics. I discovered in the fall of 2002 through a dialogue with Dr. David De Wolf and Dr. Rogers that the properties of genetic algorithms and distributed multistage memories might serve as this mathematical glue. At the start of 2003, CWT spoke with DARPA about our interest in writing a research funding proposal for their cognitive systems research area. DARPA encouraged us to dig deeper and develop the mechanism that would be used to drive our proposed cognitive radio engine research. Tasked with developing this idea and with the encouragement of my advisors, I registered for and took a course on genetic algorithms taught by Dr. Walling Cyre. The first week in class Dr. Cyre introduced me to the concept of distributed meta-genetic algorithms in a theoretical paper that he and his students wrote about adaptive GAs. I decided to focus my Ph.D. research on extending this theoretical meta-learning algorithm to my biologically inspired cognitive radio model, proposing to pursue this work for my semester class project. I proceeded with this research.

About a month into the spring 2003 semester Dr. Bostian suggested that I chat with then senior undergraduate student Tom Rondeau, as Tom was auditing the genetic algorithm class. Tom expressed interested in the cognitive radio Ph.D. research I was pursuing and in joining our research team, so I briefed him on my research and current direction. With the encouragement of Dr. Bostian, Tom and I teamed up to further develop my proposal of applying meta-genetic algorithms to cognitive radio. Tom and I drafted a white paper that served as the basis for a patent disclosure requested by Virginia Tech. Due to time constraints, Tom and I decided to explore the possibility of applying a genetic algorithm to train a hidden Markov model of a wireless channels as the class semester project. This first proof of concept algorithm served as a starting point for our team's effort to implement distributed meta-genetic learning and radio adaptation algorithms. Tom decided to stay to pursue his M.S.E.E. at Virginia Tech and continue as part of the research team, and is now a direct-Ph.D. student leading a team of students building on this research.

The research collaboration between Dr. Bostian, Dr. Cyre, Tom, and Tim over the past year or so has been quite productive. This dissertation details the results of my contributions to that three year research effort.

1.3 Organization

This dissertation is organized into seven chapters and four appendixes. Chapter 1 introduces and motivates the cognitive radio research presented in this dissertation. Chapter 2 discusses the history of cognitive radio (CR) and provides system and mathematical foundations for cognitive radio. Chapter 3 introduces the bio-formalism as a vehicle for embodying the CR concept into a model, framework, architecture, and algorithms. Chapter 4 discusses the methodology used in the experimental study of the proposed BioCR model and framework. Chapter 5 presents and analyzes results from the CR simulation test bench experiments. Chapter 6 presents and analyzes results from the CR hardware test bed experiment. Chapter 7 summarizes the research and provides recommendations for future research.

Appendix A references the patent application Virginia Tech Intellectual Properties (VTIP) submitted covering the cognitive engine model presented in this dissertation. Appendix B is a glossary. Appendix C includes documentation of simulation test bench blocks and code used to test the BioCR engine. Appendix D includes detailed data dumps from the BioCR toolset simulation run. Appendix E documents my research progress made as part of the National Science Foundation (NSF) Integrative Graduate Education and Research Traineeship (IGERT) Integrated Research and Education in Advanced Networking (IREAN) research community. A bibliography and my vita are included at the conclusion of the dissertation.

1.4 Details of Research Contributions and Resulting Publications

My research contributions include creation of a biologically inspired model, framework, architecture, algorithms, and simulation application that realize a cognitive radio (CR). Table 1.1 below provides additional details about each contribution.

Table 1.1: My Research Contributions

MODEL (describes behavior)	Created Biologically-Inspired CR Engine Model based on Mitola's CR concept and cognitive development theories
FRAMEWORK (describes process)	Developed framework for CR Engine Model using cognitive development process and genetic algorithms

ARCHITECTURE (describes components)	Developed architecture for CR Engine Model, including structure and choice of components
ALGORITHM (describes procedure)	Developed the Cognitive System Monitor (CSM) algorithm for CR Engine Model cognitive process
SIMULATION (describes applications)	Designed and implemented CR simulation test bench using MATLAB-Simulink to test CR Engine Model. Presented method for creating HMMs representing wireless channel models using genetic algorithms instead of traditional expectation maximization (EM) techniques.
CR ANALYSIS	Interpreted CR Engine Model test results to provide recommendations for next generation of CR Engine Models
DISTRIBUTED CR	Proposed distributed CR Engine Model for CR Network
WSGA INPUT	Contributed to design and implementation of Wireless System Genetic Algorithm (WSGA) algorithm
WCGA DESIGN	Jointly designed and implemented Wireless Channel Genetic Algorithm (WCGA) algorithm
IMPLEMENTATION	Throughout the research process I helped researchers implement parts of CR Engine Model into CR engine and hardware test bed

Tables 1.2 and 1.3 list papers and conference presentations that I have made or contributed to about this research, including a patent application that Virginia Tech filed in June 2004 based on this dissertation. In addition to these publications, I helped a graduate student project team complete a report describing a cognitive wireless network

inspired by my research, the CRANIAsim (Cognitive Radio for Adapative Networking and Integrated Access Simulation). This project was done as part of the NSF IGERT sponsored Integrated Research and Education in Advanced Networking (IREAN) Simulation and Optimization course taught by Dr. Patrick Koelling. This research also served as the cornerstone for a grant proposal submitted to the NSF NetS program to build a cognitive wireless network utilizing the cognitive engine model presented in this dissertation.

Table 1.2: Related Research Publications

C. J. Rieser, T. W. Rondeau, C. W. Bostian, and T. M. Gallagher. “Cognitive Radio Test bed: Further Details and Testing of a Distributed Genetic Algorithm Based Cognitive Engine For Programmable Radios.” *IEEE MILCOM*, to appear October 2004.

C. J. Rieser. “Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking.” *Ph.D. Dissertation, Virginia Tech,* August 2004.

C. J. Rieser, T. W. Rondeau, C. W. Bostian, W. Cyre, and T. M. Gallagher. “Cognitive Radio Engine Based on Genetic Algorithms In A Network.” *VTIP Reference Number 03.056 - Patent Application Filed by Virginia Tech,* June 2004.

T. W. Rondeau, C. J. Rieser, and C. W. Bostian. “Cognitive Radios With Genetic Algorithms: Intelligent Control of Software Defined Radios.” *SDR Forum*, to appear November 2004.

T. W. Rondeau, C. J. Rieser, T. M. Gallagher, and C. W. Bostian, "Online Modeling of Wireless Channels with Hidden Markov Models and Channel Impulse Responses for Cognitive Radios, " *IEEE International Microwave Symposium*, June 2004.

C. W. Bostian, S. Midkiff, T. Gallagher, C. Rieser, T. Rondeau, M. Kurgan, L. Carstensen, G. Morgan, D. Sweeney, and J. Hood, “Test bed for High-Speed 'End-to-End' Communications in Support of Comprehensive Emergency Management,” *National Conference on Digital Government Research (dgo2004) Seattle, WA,* May 24-26, 2004.

Center for Wireless Telecommunications at Virginia Tech, “CANSAS (Cognition Across Networks for Sharing Access to Spectrum),” *NSF NetS grant proposal,* April 2004.

C. W. Bostian, S. F. Midkiff, T. M. Gallagher, C. J. Rieser, and T. W. Rondeau, "Rapidly Deployable Broadband Communications for Disaster Response, " *Proceedings of the International Symposium on Advanced Radio Technologies (ISART),* invited paper in Department of Homeland Security (DHS) SAFECOM session, Boulder, CO, March 2-4, 2004, NTIA Special Publication SP-04- 409, pp. 87-92.

C.W. Bostian, T.M. Gallagher, C. J. Rieser, T.W. Rondeau. “Cognitive Radio – A View from Virginia Tech,” *Software Defined Radio Forum,* invited paper in Cognitive Radio session, Orlando, FL, Nov. 17-19, 2003.

C. J. Rieser, T. W. Rondeau, and C. W. Bostian. “Cognitive Radio Architecture Based on Genetic Algorithms: A Proposed Architecture and Some Initial Results.” *Draft journal paper,* September 2003.

C. J. Rieser. “Design and Implementation of Sampling Swept Time Delay Short Pulse (SSTDSP) Channel Sounder for LMDS.” *M.S. Thesis,* July 2001.

J. H. Reed and C. J. Rieser. “Software Radio: Technical, Business, and Market Implications. *“World Markets Series Business Briefing: Wireless Technology 2001*, World Market Research Centre, October 2000, pp. 146-150.

Table 1.3: Related Research Presentations and Reports

C. J. Rieser. "Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking." *Ph.D. Final Defense, Virginia Tech,* August 2004.

C. J. Rieser, T. W. Rondeau, and C. W. Bostian. Cognitive Radio Research, *Briefing to CWT NSF NetS research team,* April 27, 2004.

C. J. Rieser, T. W. Rondeau, and C. W. Bostian. Genetic Algorithms and Cognitive Radio Research, *Presentation to graduate class, ISE 5984: Optimization and Simulation in Networks and Telecommunications NSF IGERT IREAN class taught by Dr. Patrick Koelling,* April 14, 2004.

C. J. Rieser, T. W. Rondeau, and C. W. Bostian. Genetic Algorithms and Cognitive Radio Research, *Presentation to undergraduate class, ECE 4510: Genetic Algorithms and Evolutionary Computing class taught by Dr. Walling Cyre,* April 8, 2004.

T. W. Rondeau, C. J. Rieser, C. W. Bostian, T. M. Gallagher. Cognitive Radios: An Overview Of A Cognitive Radio Engine and Channel Modeling Techniques, *Spring 2004 Virginia Tech ECE Communications Seminar,* March 19, 2004.

C.W. Bostian and C. J. Rieser. Rapidly Deployable Broadband Communications for Disaster Response, *invited ISART Department of Homeland Security SAFECOM panel speakers*, March 2004.

C. J. Rieser, T. W. Rondeau, C. W. Bostian, and T. M. Gallagher. Biologically Inspired Cognitive Radio Test bed Based on Genetic Algorithms, *NSF IREAN Research Workshop*, February 2004.

C. J. Rieser, T. W. Rondeau, and C. W. Bostian. Cognitive Radios Based on Biologically Inspired Techniques, *NSF Networking Technology and Systems (NetS) Forum,* February 2004.

C. J. Rieser. Biologically Inspired Cognitive Radio Architecture based on Genetic Algorithms. *NSF IREAN site visit*, January 2004.

C. W. Bostian, T. M. Gallagher, C. J. Rieser, and T. W. Rondeau. Invited Panel: Cognitive Radio – A View from Virginia Tech, *Software Defined Radio (SDR) Forum*, November 2003.

C. J. Rieser. Invited Panel: A Research Perspective on Cognitive Radio Technology. *CWT Wireless Opportunities Workshop (WOW)*, September 2003.

C. J. Rieser, T. W. Rondeau, C.W. Bostian, T. M Gallagher, and W. Cyre. Biologically Inspired Cognitive Wireless L12 Functionality. *NSF IREAN Research Workshop*, April 2003.

C. J. Rieser and C.W. Bostian. Cognitive Radio Models for Wireless Systems. *NSF IGERT IREAN Research Workshop,* May 2002.

C. J. Rieser. "Design and Implementation of Sampling Swept Time Delay Short Pulse (SSTDSP) Channel Sounder for LMDS." *M.S. Final Defense,* July 2001.

1.5 Summary

Chapter one presented a summary of the evolution of cognitive radios as well as the research problem statement, historical perspective, and a summary of individual contributions. The organization of the dissertation was documented along with additional details of my research contributions and resulting publications, presentations, and reports.

Chapter 2: History of Cognitive Radio - System and Mathematical Foundations

Cognitive Radio (CR) has received significant attention recently as a potentially disruptive technology. This section discusses the history and mathematical foundations of cognitive radio and mathematical foundations for biologically inspired models of cognition.

Emerging programmable radio technology like the frequency and waveform agile radios available as part of the Joint Tactical Radio System (JTRS) program [20] promise to open up new opportunities for robust and secure military communications. These software defined radios (SDR) will become even more powerful with the addition of electromagnetic environment sensing technologies that are being developed through the Defense Advanced Research Projects Agency (DARPA) NeXt Generation (XG) Communications research program. DARPA is developing the XG technology to allow multiple users to share use of the spectrum through adaptive mechanisms that distinguish users in terms of time, frequency, code, and other signal characteristics. DARPA's goals are to enable an increase of a factor of twenty in the usage of typical spectrum [21].

In [5], Mitola proposed that a cognitive radio could serve this purpose, allowing an adaptive radio to adjust its operation based on information captured from the environment as well as measurements of its own performance. Various “meters” that describe the current radio performance can capture information provided by the radio about its operation in a given wireless channel. Mitola’s cognitive cycle appears as a directed graph that includes various states such as Observe, Orient, Learn, Plan, Decide, and Act.

Most information processing research to date would interpret that cycle as a candidate for the "if-then-else" paradigm commonly found in artificial intelligence literature. The Mitola cognition cycle then translates the resulting decision logic output to settings for the various radio "knobs" that control the wireless system's behavior in a given wireless channel.

This approach requires extensive, branching logic and requires recalculation when the decision space changes in response to environmental shifts or changes in system capability. These changes may be either complementary in which new function or waveforms become available or catastrophic in the case that part of wireless network is destroyed.

This dissertation presents a biologically inspired model of cognition which is flexible and self evolving in the face of chaotic and fluctuating decision spaces, unlike the brittle nature of traditional artificial intelligence (AI) expert systems of the past.

2.1 Mitola's Cognitive Radio (CR) Concept

Joseph Mitola's cognitive radio concept sprung from his pioneering work on software radio. Mitola postulated about a decade ago that a major shift was occurring from hardware centric radio design and implementation to software centric design and implementation [22]. Mitola proposed that taken to the limit over time, traditional radio design would change from a mix of most radio functions being performed in fixed hardware subsystems with only some radio functions performed through software execution to the majority of radio functions performed through software execution with a minimum set of radio functions being performed in fixed hardware subsystems [23]. The evolution of analog cellular radio in the 1980's to the emergence of digital cellular systems and commercial Software Defined Radios (SDRs) post 2000 has shown Mitola's vision is becoming a reality.

SDRs are viewed as an interim step towards a full software radio architecture, in which certain reprogrammable radio functions are realized in software on a general purpose processor, but some functions like radio frequency (RF) mixing and filtering may still occur in hardware. Mainstream acceptance of software radio requires affordable wideband high speed analog to digital converters (ADC) and digital to analog converters (DAC).

Mitola created some of the first SDRs for the military in the early 1990s, serving as a consulting scientist with MITRE since 1993. Then he created a software radio architecture course in 1995 which he taught for four years [24]. The course material was turned into a book, *Software Radio Architecture: Object Oriented Approaches to Wireless Systems Engineering* [2]. Near the end of the decade Mitola's interests shifted from fundamental software radio architectural issues to researching hurdles to affordable software radio.

Mitola's Ph.D. dissertation [11] focused on how these next generation programmable software radio systems could be used. He surmised that given the flexibility inherent in software radios, a new "smart" radio could be developed that was capable of sensing the surrounding wireless environment and user communications and computing needs and acting to meet those needs. Mitola proposed an emerging topic within software radio, cognitive radio. By Mitola's definition a cognitive radio was a class of software radio that employed model-based reasoning and at least a chess-like level of sophistication in using, planning, and creating radio etiquettes. As such Mitola felt that a realization of a cognitive radio was easily five to ten years in the future.

This dissertation assumes a different definition for cognitive radio than Mitola. Rather than requiring a software radio as a baseline for cognitive radio functions, this research assumes that the cognitive functions that make a radio cognitive are hosted by any agile radio including either legacy radios or software defined radios. Software defined radios are treated as a radio with more "knobs" to turn and "meters" to observe than legacy radios. The more flexible the host radio, the more powerful the cognitive capabilities.

This position was assumed so that the cognitive engine presented in this dissertation could be used to enable cognitive radio functions on existing legacy disaster communications equipment, while allowing growth for up and coming programmable SDR technology.

Mitola's dissertation discusses the various operational levels of a software radio, but the cognitive radio formalism presented in his dissertation focuses almost solely on the application layer and higher. This research instead treats cognitive radio functions as inherent to physical (PHY) and medium access control (MAC) layer operations. The resulting "embedded cognition" requires a robust model and framework which is capability of operating within the computing constraints available in current wireless hardware platforms. In addition, traditional machine learning techniques require significant computational resources, which could limit the utility of a cognitive radio – who would want a smart radio which can intelligently learn but drains a battery and memory ten times faster than older technology without intelligence?

To synthesize a cognitive radio model suitable for a PHY and MAC layer, I pursued research that started with efficient biologically inspired models of cognition instead of existing computational focused cognition models that are widely known in the artificial intelligence community.

2.2 Biologically Inspired (Bio) versus Artifical Intelligence (AI) Cognitive Models

The limitations to current cognitive modeling are well known and documented [25]. Chess class supercomputers are regularly pitted against world class human subjects to see which can "outsmart" the other. Such AI approaches rely on pure computational horsepower and complexity to "outwit" the competition. Very little research has been pursued on the opposite extreme – what is the minimum amount of "intelligence" needed to make a computationally lightweight and self evolving cognitive model that can evolve its behavior with changing environmental inputs?

Examples of traditional cognitive approaches derived from AI computational techniques include rule based systems, expert systems, fuzzy logic, and neural networks [26][27]. Each of these approaches has severe limitations that diminish their operational value for on-line cognitive radio functions, especially in changing wireless environments. Rule based systems are limited to fixed capabilities designed into their rule set. Expert systems are notoriously brittle and dependent on an external expert that must be present when the view of the environmental system response changes. While fuzzy logic permits approximate solutions to be found in the face of uncertain inputs, the logic used to find the approximations does not have an inherent evolutionary ability that allows the logic to change in time as capabilities are required and environments are encountered. The most recognized AI technique for cognitive modeling, neural networks, is typically uncontrollable in that it may or may not play within a set of operational constraints, given the inherent "black-box" nature of its operation. Most neural networks require extensive training to replicate observed behaviors and usually behave in unexpected ways when presented with a totally new problem to solve.

The biologically inspired cognitive radio model presented in this dissertation was developed to address the traditional short comings of AI systems that lacked distributed self evolution and learning capabilities often observed in models of the human cognitive development process.

Traditional AI research has focused software implementations of cognition at the application (APP) layer. Current software radio approaches to cognition have been notably layer three or application (APP) centric due to the AI legacy. Unfortunately, assuming the presence of a workstation class application computational host can result in the acceptance of levels of complexity not accepted in PHY and MAC layer cognitive functions. These layers may be limited in power consumption, size, or digital architecture and processing complexity. This research assumes that the cognitive functions operating to control an agile radio may be resource or time limited. Such functions should be designed not search for "the solution" but instead "a solution" that meets the balance of needs as best as possible within the Quality of Service (QOS) and legal requirements presented to the cognitive engine.

This dissertation provides contributions that address the current lack of research focused on adding intelligence and evolution to physical "PHY" and medium access control "MAC" layers of a radio system.

2.3 Evolvable Hardware for Programmable Wireless

One of the requirements of a cognitive radio is the ability to evolve a host radio's operation when faced with a changing environment. As such, at a minimum a radio must be programmable in its "knob" values and at best must be able to add new "knobs" when needed. In the same way, a cognitive radio requires that cognitive functions can read existing radio "meters" and potentially request new "meters" or metrics from the host radio. Creating a "self aware" programmable wireless platform is the subject of current research.

I worked with the cognitive radio research team at the Center for Wireless Telecommunications (CWT) at Virginia Tech including Dr. Charles Bostian, Tim Gallagher, and Tom Rondeau to implement the cognitive engine model described in this dissertation in a hardware test bed. The test bed includes a software system, which, together with its associated hardware, is capable of modifying its behavior in response to conditions that change quickly and in unexpected ways. I also built a flexible software simulation test bench in MATLAB-Simulink that models an adaptive radio link capable of hosting the cognitive engine software as a co-simulation.

The cognitive engine can turn any radio transceiver with "meters" (outputs like data rate that indicate current performance) and "knobs" (inputs like channel frequency) into a cognitive radio (a radio that behaves like an intelligent being, sensing its environment and modifying its behavior to meet its goals). If multiple cognitive radios are combined in a network, the software allows them to share information and work cooperatively, creating a network which is itself cognitive and can organize its members to meet specified goals like minimizing the amount of radio spectrum occupied or maximizing the amount of information transmitted. Built in rules ensure that actions taken by the network are equitable and legal.

The National Science Foundation (NSF) recently announced a new program called NSF NetS program solicitation NSF 04-540 [9] that seeks to exploit the capabilities of programmable radios to make more effective use of the frequency spectrum and to improve wireless network connectivity. The cognitive engine model described in this dissertation served as the cornerstone for a proposal Virginia Tech submitted to NSF NetS describing a project called CANSAS (Cognition Across Networks for Sharing Access to Spectrum) whose objective is to build a cognitive network and study its behavior and the implications of that behavior for radio resource allocation and wireless system operation. The proposed network will consist of 10-30 cognitive transceivers operating in the 2 MHz – 2 GHz radio spectrum. The radios and the network will solve multidimensional problems of spectrum access and transmission efficiency - for example, how a wireless local area network can share an FM broadcast channel by hiding below the noise level of the FM receivers, modifying its behavior as needed to remain hidden. In networking terms, the radios will think across the PHY (physical) layer (the physical characteristics of the radios, like their transmitter power), the MAC (medium access control) layer (how the individual transceivers share the radio channel), and beyond.

2.4 Virginia Tech Broadband Wireless Channel Sounder

The cognitive engine model presented here relies on snapshots of the current wireless channel to evolve its behavior. In the future the task of sensing a wireless channel may be realized by fast spectrum sampling chipsets, like those being developed in the DARPA XG program. Scanning receivers that use high resolution and high speed spectrum sampling chipsets can provide channelized views of the wireless environment and permit cognitive functions to solve multimodal access optimization problems.

The current cognitive radio hardware test bed has been designed to use channel performance data from a wireless channel sounder shown in Figure 2.1. Virginia Tech created such a sounder for use in rapidly deployable wireless disaster response communications [28].

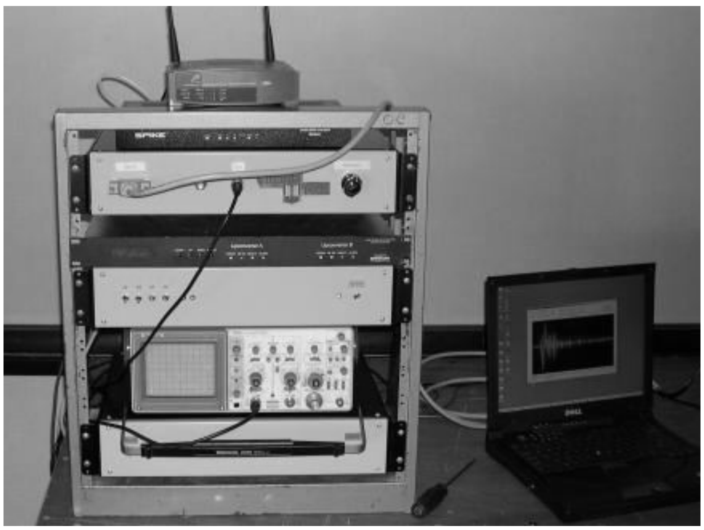


Figure 2.1: Virginia Tech broadband channel sounder

The Virginia Tech Sampling Swept Time Delay Short Pulse (SSTDSP) Broadband Wireless Vector Channel Sounder uses alternative processing methods to reduce the complexity and cost of implementing a wideband digital channel measurement system. The SSTDSP sounder transmits an impulse-like signal, or Ultra Wide Band (UWB) pulse shape, over the wireless channel of interest and uses the SSTDSP method to economically and efficiently digitize the received channel impulse response in the time domain. The SSTDSP sounder has been used by [29][30][31] to identify “paths of opportunity” for rapidly deployable broadband wireless disaster response communications. These paths may be non line of sight or reflected single "bounce paths" that extend the effective coverage of a wireless communications network.

The digital impulse response is used to assemble the power delay profile and calculate a number of key metrics that allow researchers to determine the sustainable bandwidth over that link. The sustainable bandwidth of a wireless link is a tradeoff between data rate, bit error rate, and throughput. By providing this information to Geographic Information Systems (GIS) applications operating in the field, an optimum network topology can be calculated. In addition, the status of the channel can be monitored and radio characteristics optimized to provide control of error correction coding, modulation, and power levels.

Recent research by Gallagher showed that wireless link performance can be directly estimated from a channel impulse response taken from the channel sounder [31]. Gallagher's algorithm calculates bit error rate (BER) performance of unanticipated fixed disaster response communications channels that may contain specular reflections and/or diffuse scattering. These observed channel statistics can then serve as inputs to the cognitive model proposed in this dissertation, allowing classification of the channel and resulting control of the agile radio platform to ensure robust and secure communications. Gallagher's research contributions serve as the front end to a cognitive radio, providing efficient channel performance characterization, while my research focused on what the radio would do with this online channel description.

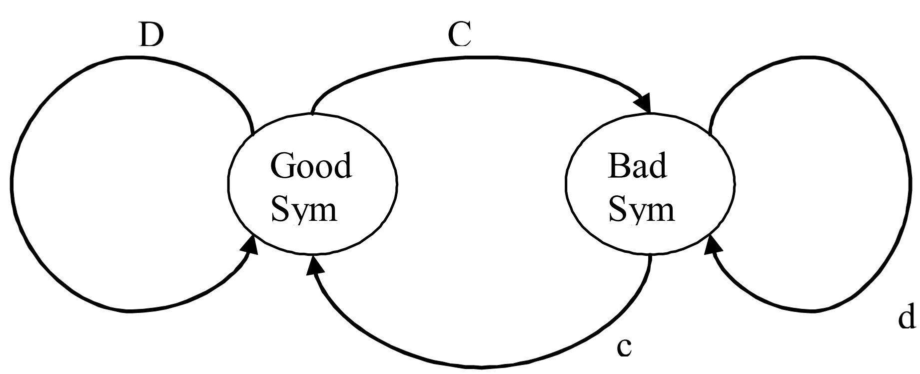
When the sounder is used for fixed broadband wireless systems it must periodically pass its channel measurements much less frequently than mobile devices would require. While the Virginia Tech sounder provides a current snapshot of the electromagnetic environment to the cognitive radio, since the cognitive engine is fully distributed not every cognitive radio requires a sounder. The sounder can characterize the channel and then pass those channel statistics to other neighboring cognitive radios, in a scanning receiver mode.

The sounder was designed for high speed wireless backbone applications and therefore is too large and expensive for small mobile radio applications. New high speed spectrum sampling application specific integrated circuits (ASIC) are being developed by vendors that may some day be transitioned to mobile devices.

2.5 Compact Channel Models at the Symbol/Waveform Level

This research was conducted prior to Gallagher developing his algorithm, so the algorithm was not available to quickly characterize the waveform level wireless channel and map channel changes to bit error behavior. A method of rapidly and compactly capturing and storing the symbol level error behavior of the channel was needed. Given that Hidden Markov Models (HMMs) have been used to generate error patterns in communications system simulation, they were chosen as a candidate for compactly describing the symbol level bit error behavior of a channel. I investigated the use of genetic algorithms to train HMMs, which served as a proof of concept exercise for the use of GAs in cognitive radio. A brief discussion of HMM channel modeling is provided in this section.

Gilbert first proposed a method for modeling burst error digital channels [32]. In his model shown in Figure 2.2, the channel has two states, a good state and a bad state. The transitions between these states are governed by a transition probability matrix A, where the variables c,d,C, and D are probabilities [33]. Rows of the matrix A must sum to one.



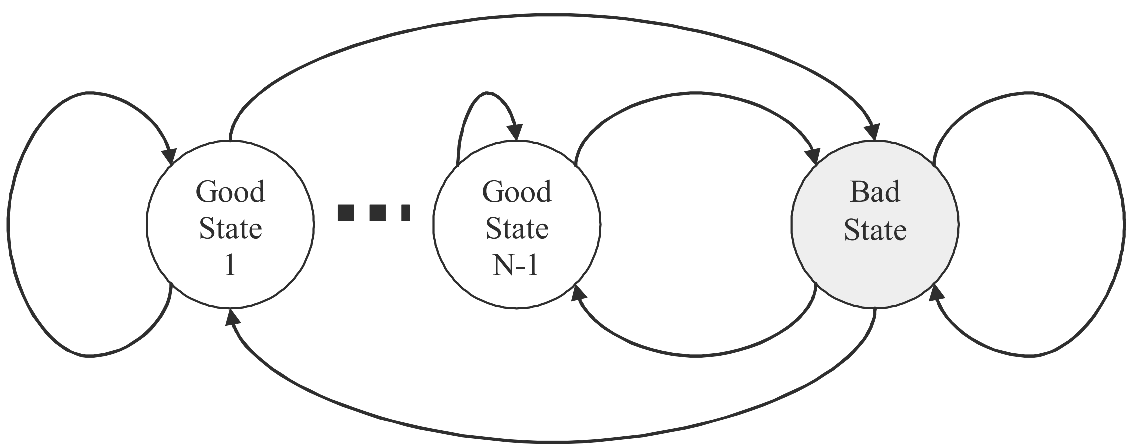
$$A = \begin{bmatrix} D & C \\ c & d \end{bmatrix}$$

Figure 2.2: Gilbert’s model

One of the primary limitations of Gilbert's model is that it lacks the ability to describe more complex bit error state transitions due to its simple geometric distribution.

Fritchman improved on Gilbert's model by proposing a state partitioned model in which there is more than one good state and errors occur only in one bad state with probability of one [35]. The burst length distribution is therefore polygeometric which is more realistic than Gilbert's model, but the Fritchman model assumes that the intervals between consecutive errors are independent and identically distributed, which is not always true of experimental data.

Fritchman's model is illustrated in Figure 2.3 [35].



f(Good state 1) = 0

f(Good state 2) = 0 **f(Bad state) = 1**

…

Figure 2.3: Fritchman's model

The limitations of Gilbert and Fritchman's models led some researchers to propose multi-state Markov models like HMMs. Since HMMs have more than one good state and bad state, they are able to characterize the dependence between successive error and error-free runs [36].

Per Rabiner's tutorial [37], Hidden Markov Models are described by the set $\lambda = (A, B, \pi)$, where A is a transition probability matrix, B is an observation probability matrix, and π is the initial state matrix. The transition matrix A governs what state the wireless channel switches to, while the B observation matrix determines which error symbol will be displayed for a given radio channel state. The initial state matrix π controls the initial state of the wireless channel modeled by the HMM.

The HMM of Figure 2.4 has N = 3 states and M = 2 possible outputs from any state.

A

A11	A12	A13
A21	A22	A23
A31	A32	A33

B

B11	B12
B21	B22
B31	B32

π

π 1	π 2	π 3

Figure 2.4: An example HMM

Typically, there are three problems of interest when applying HMMs to real world models:

1. EVALUATE: Given an observed sequence O and model λ, efficiently compute $P(O|\lambda)$ the probability of the observed sequence given the model
2. DECODE: Given an observed sequence O and model λ, choose the optimal state sequence Q that best explains the observed sequence
3. TRAIN: How can the HMM parameters $\lambda = (A, B, \pi)$ be adjusted to maximize $P(O|\lambda)$ the probability of the observed sequence given the model ?

Problem 1 "Evaluate" can be thought of as a scoring of how well an HMM model λ matches a given observation sequence, which allows us to pick among competing HMM models the model that "best fits" the observations. The observed sequence can then be processed to formulate statistical metrics like a probability density function (pdf) that can be compared to the pdf of the original data that was used to derive the HMM model λ.

Problem 2 "Decode" can be thought of as an attempt to find the "correct" state sequence that generated the observed sequence. Researchers note that except for degenerate models no "correct" state sequence can be found due to the statistical nature of the model because several reasonable optimality criteria can be imposed to determine the "correct" state sequence. Each time the model is run, a different state sequence can be found that matches the given model. Therefore, use of the state sequence is primarily to learn more about the structure of the model by observing average statistics of individual states. For the application of communications system error generation modeling, one is less concerned with the exact state sequence and more concerned with the statistics of the observed sequence produced by a given HMM.

Problem 3 "Train" is perhaps the most important problem as it allows us to optimally adapt HMM model parameters to observed training data thereby producing the most accurate models for real world signals.

Typically, iterative solutions are used instead of analytic methods because of the complexity of the problem. The Baum-Welch Algorithm (BWA) is the classical iterative method used to estimate and train an HMM model $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$. Most of the complexity of the BWA resides in the second step of the iteration, which is based on expectation maximization (EM) techniques.

1. Let the initial model be $\lambda 0$
2. Compute the new λ based on $\lambda 0$ and observation O

3. If the log $P(O|\lambda) - \log P(O|\lambda 0) < DELTA$ stop, where DELTA is some small difference
4. Else set $\lambda o = \lambda$ and go to step 2

The GA approach to training an HMM is also an iterative method but instead uses the GA's ability to achieve global optimization in a parallel manner to rapidly find the best HMM model for a channel.

HMMs have some similarities to neural nets. Just as the transitions between states in an HMM are hidden, neural nets have a hidden layer that converts input layer stimulus to output layer response. These hidden layers allow the computational graph to model complex error symbol behavior. The neurons in neural nets are weighted nodes with activation functions that operate on input layers to detect features hidden in the input layer. The detected features are then used by the output layer to present a network response. HMMs use the hidden statistical transitions between internal states in concert with output observation probability vector to model complex observed responses generated by the interaction between an input stimulus and the environment.

HMM characterizations of wireless channels have many applications. Researchers have shown that the performance of various decoding techniques depends on the bursty nature of the errors in the received data packet [38]. Since the specific nature of a mobile wireless channel often results in bursty received errors, the physical layer radio performance must be characterized by both the bit error rate (BER) and a mechanism to emulate the burst nature of error streams [39]. HMMs are well suited to this task and can be trained via statistically accurate data obtained from off-line simulations.

As an example, a simulation model implements every transmission element and can be used to derive the wireless channel behavior in terms of error distribution. The emulation model considers the system as a black box, which implies a loss of accuracy with respect to simulation models but is adequate to operate in real time. The results from using HMM models to emulate wireless channels indicate that the loss of accuracy using

HMMs is negligible, while providing significant reduction in time and resources when compared to real simulation of the system.

HMM characterizations may be validated by a two step process [40]:

(1) Validation of errors introduced by the channel
(2) Validation of the soft decision information associated with each bit.

The validation of errors introduced by the channel consists of analyzing the errors within the frame and comparing them with those of the real channel simulation. The analysis is done by means of the metric generated by the channel and the simulation metric.

Specifically, the following statistics may be computed:

(1) Histogram of number of errors per block or frame
(2) Histogram of the length of error bursts
(3) Histogram of free error intervals

The validation of the soft decision information associated to every bit requires computing the following statistics and comparing them to the simulations of the real channel:

(1) Histogram of block soft decision mean
(2) Histogram of dispersions around means with non-zero probability
(3) Histogram of soft decision levels for every non-zero mean

By using a known HMM of the wireless channel to generate the bit error behavior of wireless channels, a typical simulation time savings of two or more orders of magnitude can be observed when compared to traditional Monte Carlo simulations [36]. Libraries of HMM characterizations can be used to emulate and classify observed channels.

While they are compact and fast representations of symbol level error channels, HMMs of wireless channels do have limitations. If a wireless channel or radio under test changes, a new HMM must be created. This dissertation presents a method for creating HMMs using genetic algorithms instead of traditional Baum-Welch expectation-maximization (EM) techniques.

Because of these limitations, the current instantiation of the cognitive engine uses statistics from symbol level error streams and sounder waveforms to classify wireless channels. These compact channel models permit the cognitive engine to operate both at the symbol level and waveform level.

2.6 Overview of Genetic Algorithms

Genetic algorithms (GAs) are algorithms rooted in biological functions like reproduction and evolution, capable of rapidly searching a solution space. David Goldberg provides an excellent discussion of genetic algorithms for optimization and machine learning in his 1989 book "Genetic Algorithms: In search, Optimization, and Machine Learning" available through Addison-Wesley [65]. Goldberg's book provided the foundation from which the GAs developed in this dissertation evolved. These algorithms operate on chromosomes, which may be representations of a multi-dimensional solution search space. Chromosomes are comprised of numerous individual "genes" which represent problem variables, each of which may take on different "allele" values which represent the variable scope. Figure 2.5 shows an example radio chromosome with individual genes and alleles:

Genes →	Power	Frequency	Code Rate	Modulation
Chromosome 1 →	0 dBm	2 GHz	1/2	QPSK
Chromosome 2 →	6 dBm	3 GHz	3/4	BPSK

Figure 2.5: Example radio chromosome with alleles

Genetic algorithms take a population of chromosomes using a genetic operation called "selection" and mix the genes of its members through a genetic operation called "crossover" to produce offspring. These offspring solutions may be further randomly altered using a genetic operation called "mutation". Figure 2.6 shows an example of crossover and mutation.

Step 1: selection of chromosome 1 and 3 based on minimum power fitness function

Genes	Power	Frequency	Code Rate	Modulation	Fitness (min power)
Chromosome 1	0 dBm	2 GHz	½	QPSK	QPSK
Chromosome 2	9 dBm	4 GHz	2/3	64-QAM	64-QAM
Chromosome 3	6 dBm	3 GHz	¾	BPSK	BPSK
…					

Step 2: crossover at power gene, see light and dark interchange

Genes	Power	Frequency	Code Rate	Modulation
Chromosome 1	6 dBm	2 GHz	1/2	QPSK
Chromosome 3	0 dBm	3 GHz	3/4	BPSK

Figure 2.6: Example radio chromosome crossover and mutation

Each member of the entire population is then evaluated using a "fitness function" which represents how closely that chromosome solution solves the problem at hand; the most fit chromosomes survive and are "reproduced" and the rest are discarded. Figure 2.7 shows the final output of the GA, a radio with 0 dBm output power, 4 GHz center frequency, ½ code rate, and QPSK modulation. This information may then be passed to the radio through an operation called "expression" using an application programming interface (API) that translates the chromosome to operational radio commands.

Step 4: selection of chromosome 1, minimum power

Genes	Power	Frequency	Code Rate	Modulation
Chromosome 1	0 dBm	**4 GHz**	1/2	QPSK

Figure 2.7: Example radio chromosome selection

By keeping the most fit chromosomes, the population converges on an optimal solution by exploiting best practices among the population members. When a population member achieves the optimal solution, it is chosen as the solution. GAs are particularly well suited for applications like cognitive radio where the search space can be time varying and require constant evolution, because

1. GAs work with a representation of the parameter set, not the parameters themselves
2. GAs search from a population of points, not a single point
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge
4. GAs use probabilistic rules, not deterministic rules

In a simple GA like that above, several major steps occur:

1. Reproduction
2. Selection
3. Crossover
4. Mutation
5. Expression

In a cognitive radio framework based on GAs, crossover is viewed as a synthesis of best practices and mutation is viewed as a method for spontaneous inspiration and creativity. Since GAs operate on a coding of a parameter set and not the actual parameter set, they can be used in a number of applications and are well suited to evolving radios that have any number of knobs that can be turned. The flexible coding of chromosomes in GAs also allows both the addition and evolution of knobs to a radio. Chromosome structure can simultaneously embody the current radio parameter set, channel behavior, and evaluation function used to evolve the radio.

2.7 Summary

Chapter 2 discussed the foundations of cognitive radio, including exploration of the concept, model, key mathematical techniques, and supporting hardware platforms. These foundations serve as the ingredients for the cognitive radio recipe proposed and detailed in Chapter 3.

Chapter 3: Bio-formalism as Vehicle for Embodying the CR Concept

The previous two chapters provided an introduction, motivation, and history of cognitive radio research. Chapter 3 details my Ph.D. research proposal, including model, framework, architecture, and algorithms. In brief review, current wireless communication systems can be described as either *fixed* where the radio's technical characteristics are set at the time of manufacture, or *adaptive,* where the radio can respond to channel conditions that represent one of a finite set of anticipated events. Researchers have postulated that cognitive radios could be used to enable intelligent wireless networks that evolve in time, but very few cognitive radios have been implemented. Due to my focus on disaster communications technology during my graduate research, I decided to concentrate my cognitive radio research on how to create a cognitive radio model and framework that could respond intelligently to an unanticipated event; i.e. a channel that it has never encountered before.

3.1 Proposal: Bio-formalism as a Foundation for a Model of the CR Concept

As discussed in Chapter 2, Mitola's cognitive cycle appears as a directed graph that includes various states like Observe, Orient, Learn, Plan, Decide, and Act [5]. Most information processing research to date would interpret that cycle as a candidate for the "if-then-else" paradigm commonly found in the artificial intelligence literature. This approach requires extensive, branching logic and requires recalculation when the decision space changes in response to environmental shifts or changes in system capability. These changes may be either complementary in which new function or waveforms become available or catastrophic in the case that part of wireless network is destroyed.

I interpreted the cognitive cycle differently from a traditional branched logic or interconnected graph representation like a neural net. While most models of cognition based in artificial intelligence attempt to start with a computational model and then use that to model cognition, I chose instead to start with a biological framework of cognition based on the human cognitive development process, then mapped this framework to a computable model of the brain that was able to learn and evolve its operation using mathematical operators used by real biological systems to evolve their characteristics.

This distinction is important.

I have labeled the cognitive engine model proposed in this dissertation as biologically-inspired because it uses genetic operator-based computational techniques observed in real biological systems to model the ongoing parallel cognitive development process. This approach is contrasted with historical efforts to map generic mathematical operations to a multitude of activation functions that AI researchers use to represent the neurons in a physical brain [41]. Such brain emulations have difficulty scaling to large systems due to the memory and nodal communications requirements – neurons require rapid interconnection communications that can be created or destroyed at a moment's notice [42]. These nodal messaging requirements tax even the most powerful parallel computing architectures [43][44][45]. A fundamental feature of biological systems is their ability to evolve in response to external influences. Traditional AI approaches are unable to evolve their mathematical and functional structure to accommodate or assimilate new environments and tasks. This can be seen in the inability of one family of neural network activation functions to model all operations by the brain, further limiting a neuron's inherent ability to evolve without external influence. In fact, biological neurons do not work backward to adjust the strengths of their interconnections called synapses, so the "back propagation" process used to train neural networks most widely used to in neural network models is a biologically inaccurate process model [46].

The abstraction proposed in this dissertation was created to reduce the computational load of the model – radio operation is not modeled by thousands of neurons but instead by dynamic flows of operational commands that are contextual in nature and specify any information that is available about the electromagnetic environment and the adaptive radio system. Only information that needs to be dealt with or an abstract knowledge-base representation of past choices is maintained, along with the option of creating entirely new radio configuration solutions based on programmable choices available to the user.

I propose assuming a biologically inspired model of cognition derived from a model of the human cognitive development process which is fostered through creative play and interaction with the surrounding environment. Such a cognitive engine could avoid the many problems that plague brittle expert systems and other AI technologies. These rule based systems do not perform well in the face of unknown situations and often suffer from lack of scalability due their inability to efficiently learn new knowledge [26]. This performance degradation occurs because conventional artificial intelligence attempts to express human knowledge in symbolic terms, which requires rigid symbol manipulation and exact reasoning mechanisms, including forward and backward chaining. Follow on AI research has expanded to include artificial neural networks, genetic algorithms, and fuzzy set theory. These mechanisms are intentionally vague in their operation.

An emerging research area called "soft computing" includes mechanisms that leverage hybrid combinations of these techniques that can reason and learn in uncertain and imprecise environments [47]. My research focused on bringing soft computing techniques to cognitive radio by synthesizing a chaotic learning technique based on genetic algorithms with an abstracted distributed staged memory derived from case-based reasoning techniques to form a chaotic learning optimizer.

3.2 BioCR Model

Mitola's dissertation provides an extensive discussion of how artificial intelligence research might be extended to cognitive radio systems. His dissertation proposes that cognitive radio systems need a mechanism which synthesizes the best aspects of machine learning, case-based reasoning, and rule based systems in a radio engineering framework [11].

Mitola defines the machine learning process as the extraction of a concept description from examples and background knowledge. This process may produce an algorithm that can recognize additional instances of the learned concept. In his formalism, machine learning techniques may include conceptual clustering which derives predicate-calculus expressions given an unstructured database of cases, reinforcement learning which extracts rewards from the environment to structure the machine learning, and case based reasoning which retrieves and applies cases to new situations. Case based reasoning retains sets of "problems" with associated "solutions." Such case based systems may be data intensive, requiring retention of a large amount of original data points.

Contrasting approaches include statistical techniques that use feature-space information like the cluster center or covariance matrix for the data set instead of the original large data set. Case based systems that retain the original data attempt to retrieve the most relevant case or data point to apply a corresponding prior solution or associated class of data point to the current problem. The solution may then be revised to provide a better fit for the current situation. Successful solutions are archived with the associated problem, completing the "retrieve, reuse, revise, and retain" cycle.

Mitola hints at the use of case based reasoning as a way to include temporal and environmental information in the cognitive process. He defines rule based systems as using "if-then-else" logic which provides a structured decision making mechanism that tends to be "brittle," providing poor performance when presented with problems that are slightly different. Some research has been done to address brittleness of rule based systems by tracing rule schema, but this approach is still unable to evolve in completely new scenarios.

To be clear, my research aims to realize the vision of Mitola's cognitive radio concept in spirit, albeit in different ways. Rather than trying to model the brain with an if-then-else state machine operating on a large data base of case based scenarios, I assumed a cognitive model based on the chaotic learning process observed in the cognitive development process of young minds, extending several theories linking play in children to creativity and rapid learning [48].

Many AI methods like expert systems could be described by cognitive development researchers as information processing techniques which view human minds as computers that act on a flow of information represented by symbols without regard to stages of understanding. Information processing techniques do not produce effective cognitive models that include imagination and creativity because such techniques assume linear processing that is unable to evolve to assimilate or accommodate new solutions [4]. Such limitations often lead to poor performance when radios are required to operate in unfamiliar and unfavorable electromagnetic environments.

A contrasting model of cognition can be derived from Piaget's theory of cognitive development that postulates cognition in children is developed through active manipulation and exploration of the world that takes place in a continuum of stages [49]. Creativity researcher Chsikzenthmihalyi defined the concept of creative flow as being a balance between boredom and anxiety; people enter a flow state when they are fully absorbed in activity during which they lose their sense of time and have feelings of great satisfaction [50]. This concept of maintaining flow in a creative cognitive model provides motivation and bounds for a bounded chaotic solution exploration mechanism. While Piaget observed that children learn through play in new situations, Vygotsky observed that children learn through peers and educators, through a process called scaffolding that may not always be the same for each child due to differences in environment and interactions [51]. This human development research provided insight into how to develop the cognitive engine architecture and process flows, especially regarding how to scaffold a cognitive radio's understanding and how to act in an unknown wireless channel.

To meet the need of staged creative learning, I proposed a distributed two stage memory in the cognitive engine to enable stages of creative learning so that the engine can learn in conjunction with peers and as an individual. Short term memory serves as a working space that can operate on a larger knowledge base found in long term memory. The short term memory allows the cognitive engine to consider possible interactions with the environment while not destroying the more stable knowledge base.

Such constructivist theories state that cognitive functions operate to satisfy a dynamic equilibrium, with joint focus on function and structure [52][53]. Dynamic equilibria are observed and modeled in the world everyday, often using a Nash equilibrium [54]. John Nash postulated that if there is a set of strategies for a game with the property that no player can benefit by changing his strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute Nash equilibrium [55][56]. In layman's terms, a system may find an operational point which effectively balances the operational parameters of the system to balance user needs.

In this dissertation, the cognitive engine plays chaotic games or "what if" scenarios trying to learn how to find the optimal equilibrium of actions that meet the cognitive engine's operational goals for learned environment. The equilibrium may or may not appear to be Nash-like. The games the engine plays to balance the operational actions of the system are affected by time and relative weights of functions which are device-specific. What is interesting is that this model appears to permit multiple game solutions that satisfy a desired operational equilibrium, which would not appear to be Nash-like. Determining the best method of analyzing cognitive radio behavior is an active area of research.

Given this model of cognition based on the human learning, I developed a formalism for cognitive radio that mapped the rapid cognitive development of children to a functional structure and a mathematical architecture that could be implemented by wireless systems. A broad survey of potential candidates was investigated and genetic algorithms were chosen as a key component to the mathematical architecture of the distributed cognitive wireless engine presented in this paper.

Candidates that were rejected included rule based systems, expert systems, fuzzy logic, and neural nets. Each of these approaches has severe limitations that diminish their operational value for on-line cognitive radio functions, especially in changing wireless environments. Rule based systems are limited to fixed capabilities designed into their rule set. Expert systems are notoriously brittle and dependent on an external expert that must be present when the view of the environmental system response changes. While fuzzy logic permits approximate solutions to be found in the face of uncertain inputs, the logic used to find the approximations does not have an inherent evolutionary ability that allows the logic to change in time as capabilities are required and environments are encountered. The most recognized AI technique for cognitive modeling, neural nets, is typically uncontrollable in that it may or may not play within a set of operational constraints, given the inherent "black-boxed" nature of its operation. Most neural nets require extensive training to replicate observed behaviors and usually behave in unexpected ways when presented with a totally new problem to solve.

I chose genetic algorithms and operators in concert with a biological abstraction of the brain for a number of reasons, including the GA's ability to implement a number of the cognitive development theories simultaneously and in parallel [49][50][57][58][59][60]. Specifically I surmised that genetic algorithms and operators could serve as the mathematical glue to realize the human-based cognitive development process because their crossover operation permits synthesis of best practices, and mutation permits spontaneous creativity in the face of unknown scenarios that would break a traditional, brittle AI expert. The GA concept of populations of solutions permits data to be structured and avoids losing potential non-optimal solutions that could be evolved to provide a best fit. The GA fitness function permits rapid adaptive global solution search when combined with context and environment-based genetic selection using genetic tagging and templates, a mode that embraces both assimilation with a stable fitness function and accommodation in which the fitness function changes when a change is the environment is sensed.

The concept of generations of solutions and the chaotic nature of GAs lends them to very complex multimodal parallel play in search of best-fit solutions for ensuring robust transmission and reception of signals in unknown channels in changing environments. Such meta-GA functions allow the GA to serve as a smart inter layer and intra layer parameter optimizer and learning classifier. The evolving populations of GAs provide a mechanism for growth needed to respond to new environments and evolve new cognitive radio behaviors out of existing and created responses.

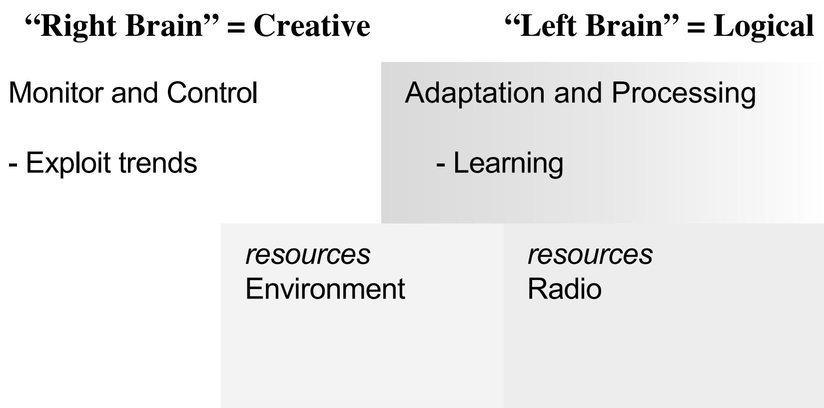


Figure 3.1: Concept-level block diagram of cognitive engine

The cognitive engine concept in Figure 3.1 assumes that the biological functions in the right brain maintain creative functions while the functions in the left brain maintain logical thought. The division of labor between cognitive setting of goals by the creative module and real time adaptation by the logical module ensures low overhead communications between the modules and scalability to large networks of cognitive radios. This delineation also recognizes that the cognitive engine needs to operate considering both real world data and simulated solutions.

3.3 BioCR Framework

This section describes the CR process. The resulting BioCR framework was inspired by a diverse set of research realms. The concept for the Wireless Channel Genetic Algorithm

(WCGA) was extended from existing research that used GAs to train HMMs that represent speech to HMMs that represent wireless channel [61][62][63]. The concept of the Wireless System Genetic Algorithm (WSGA) as a self-evolving genetic algorithm was based on a theoretical adaptive GA article written by Dr. Walling Cyre and his students [64]. For an excellent reference to the GA concepts presented in this dissertation please refer to Goldberg [65]. The concept of the Cognitive System Monitor (CSM) is based on learning classifiers and optimizers found in evolutionary computing research circles [65][66].

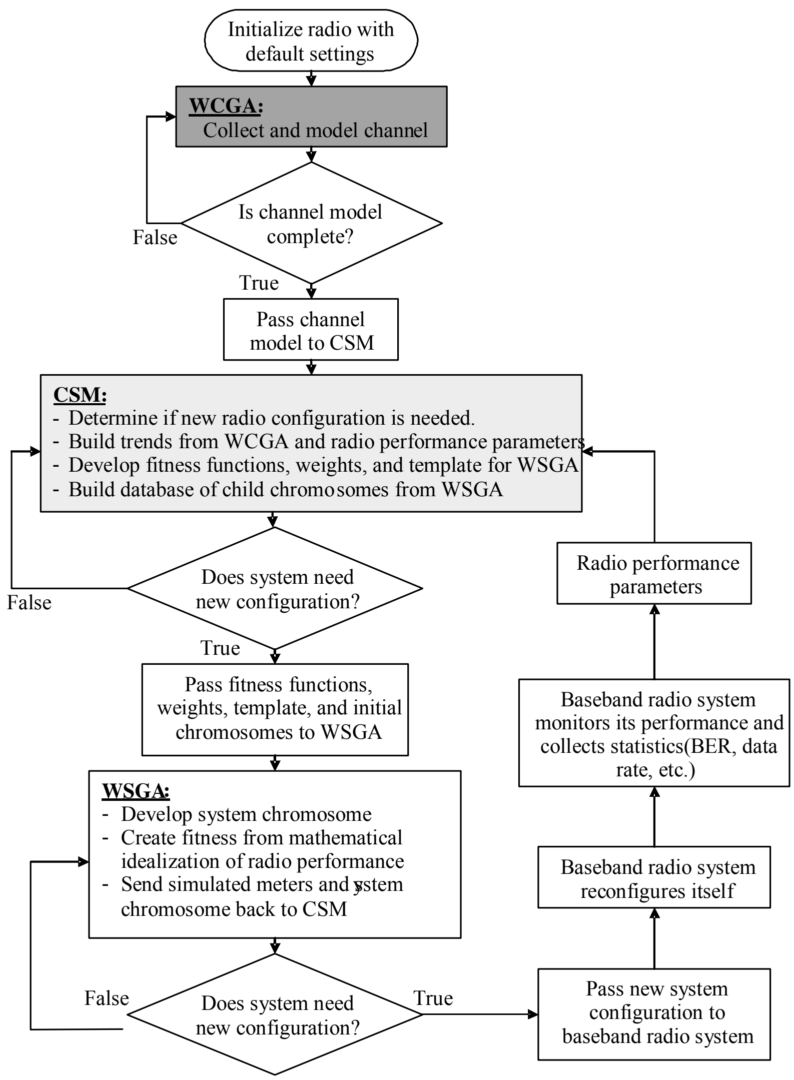


Figure 3.2: Biologically inspired cognitive engine framework

I propose that the biologically-inspired framework shown in Figure 3.2 could be used to realize cognitive wireless functionality in an adaptive radio using genetic algorithms.

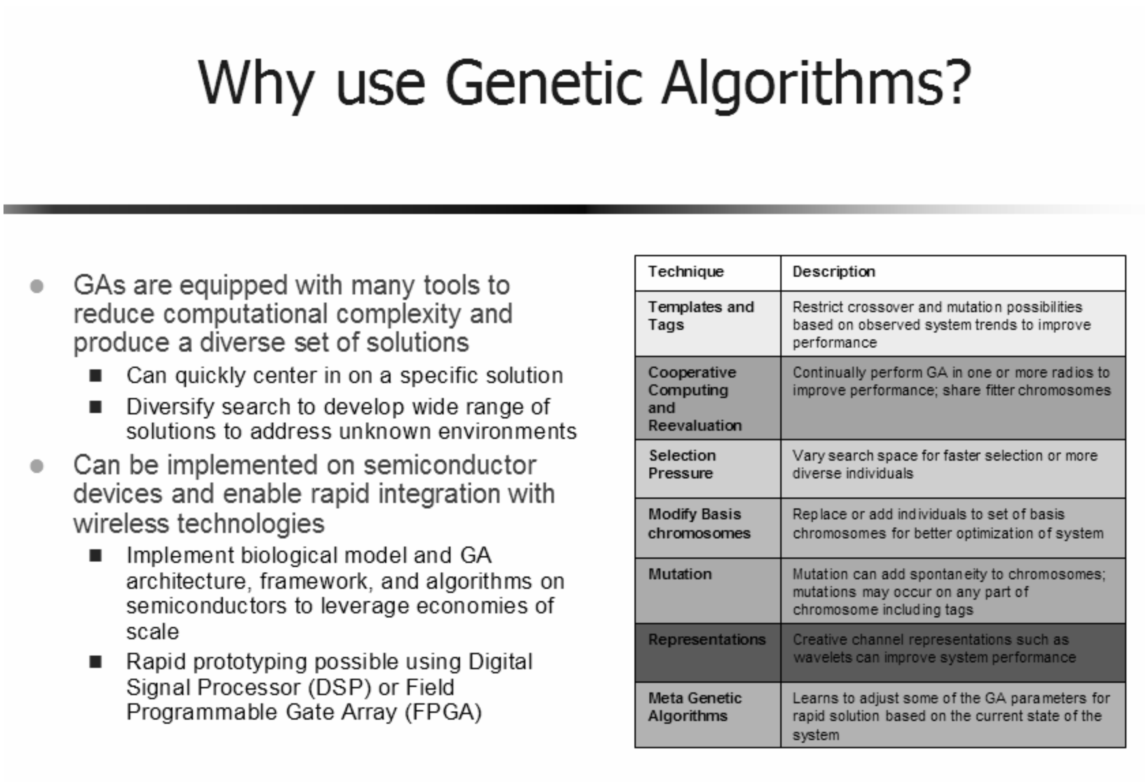


Figure 3.3: Advantages of using genetic algorithms in a cognitive radio

Figure 3.3 details the advantages of using genetic algorithms in a cognitive radio. Some advantages include their chaotic search capability and flexibility, as well as their ability to be implemented on a vector based co-processor. The biologically inspired cognitive radio framework includes a wireless channel genetic algorithm (WCGA) that senses the wireless environment, a wireless system genetic algorithm (WSGA) that evolves and adapts the radio, and a Cognitive System Monitor (CSM) based on a meta-genetic algorithm and adaptive control messaging that monitors and changes the behavior of the system. The process is cyclical and runs continuously.

The CSM synthesizes knowledge gained from sensing the current state of the wireless channel and the current radio parameters to direct the adaptation process. The CSM includes a meta-genetic algorithm (meta-GA) that adapts the WSGA and distributed short- and long-term memory. A meta-GA is a genetic algorithm that is used to evolve another genetic algorithm. The distributed memory is used to understand and utilize past trends of channel/goal pairs that impact radio system behavior. The CSM translates machine readable regulations specific to the current geographical location so that it can operate within the regulatory and physical environment constraints mandated at that locale. The CSM sends tags and templates to the WSGA indicating which parameters may legally be altered and what parameters should be left alone, based on knowledge gained from the past trends. The CSM is the creative side of the cognitive radio brain.

The WSGA receives operational goals from the CSM that are used to configure a GA used to optimize the radio configuration. The WSGA is the logical side of the cognitive radio brain. The WSGA tradeoff optimizer discovers the appropriate balance of radio parameters like transmitter power, frequency, bandwidth, modulation, and channel coding embedded in a chromosome. The WSGA also receives information like fitness functions as well as geo-specific tags and templates that impact how the algorithm attempts to evolve the radio's operational parameters. The fitness could be a measure of the minimum BER or of the maximum data rate. Depending on what information the CSM receives about the radio and environment, the fitness function may change to achieve a new goal. The WSGA then uses the CSM goals to synthesize an appropriate configuration. This configuration is passed to a programmable radio, which validates the veracity of the proposed configuration. The CSM monitors the radio's new performance and recommended a different configuration if the performance is not optimum.

The WCGA characterizes the electromagnetic environment, including relevant waveform level multi-path information and symbol level channel statistics. This information is passed to the CSM to classify the channel and estimate the performance of the link including maximum data rate. Geo-specific information is then used to validate the radio system configuration for a given locale and cast the performance results within the context of that channel.

3.4 BioCR Architecture

This section describes the CR engine component input and output. Figure 3.4 shows the data flow between the three cognitive engine modules. The radio hardware interprets radio frequency information about the wireless channel using the channel estimation process and passes this information on to the WCGA. The radio hardware also interprets information about the radio and data using the PHY and MAC layers of the radio baseband processor and passes this information on to the CSM. The WCGA takes the information about the estimated wireless channel and produces statistics and a compact model of the observed channel which is passed on to the CSM. The CSM uses its learning classifier functions to translate the observed channel model into appropriate goals which are passed to the WSGA in the form of fitness functions, initial chromosomes, and appropriate regulatory genetic tags and templates. The WSGA takes these goals and generates appropriate actions for the radio to take in the given wireless channel.

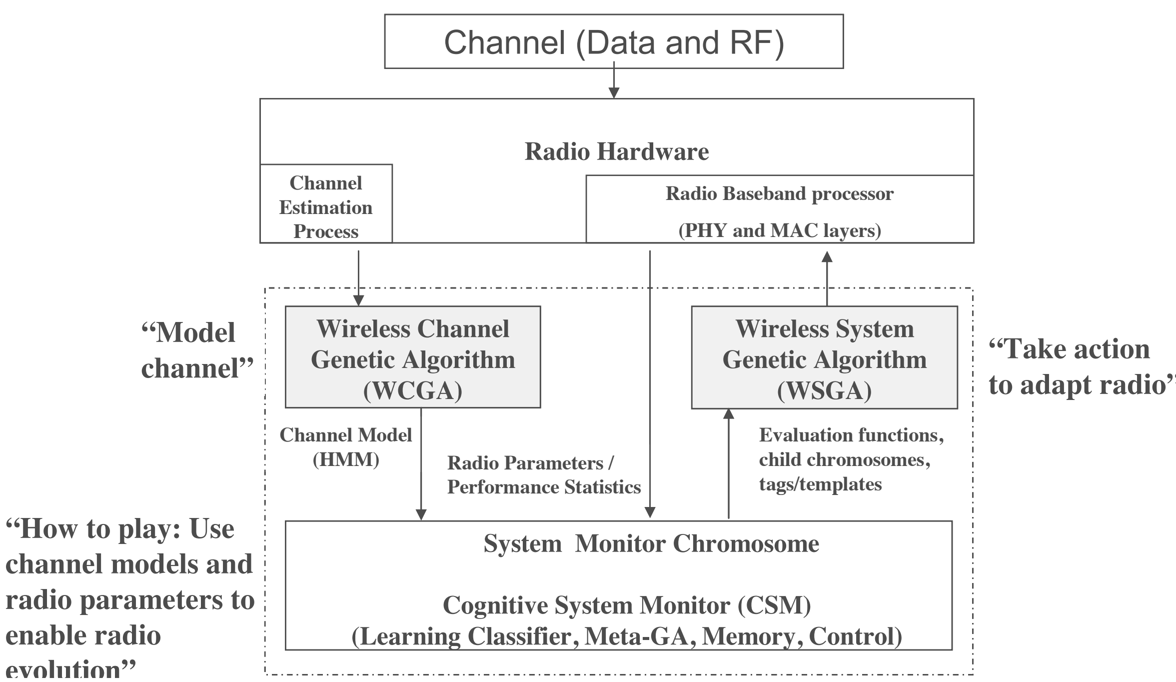


Figure 3.4: System-level block diagram of cognitive engine

The following information provides additional detail of the cognitive engine block inputs and outputs.

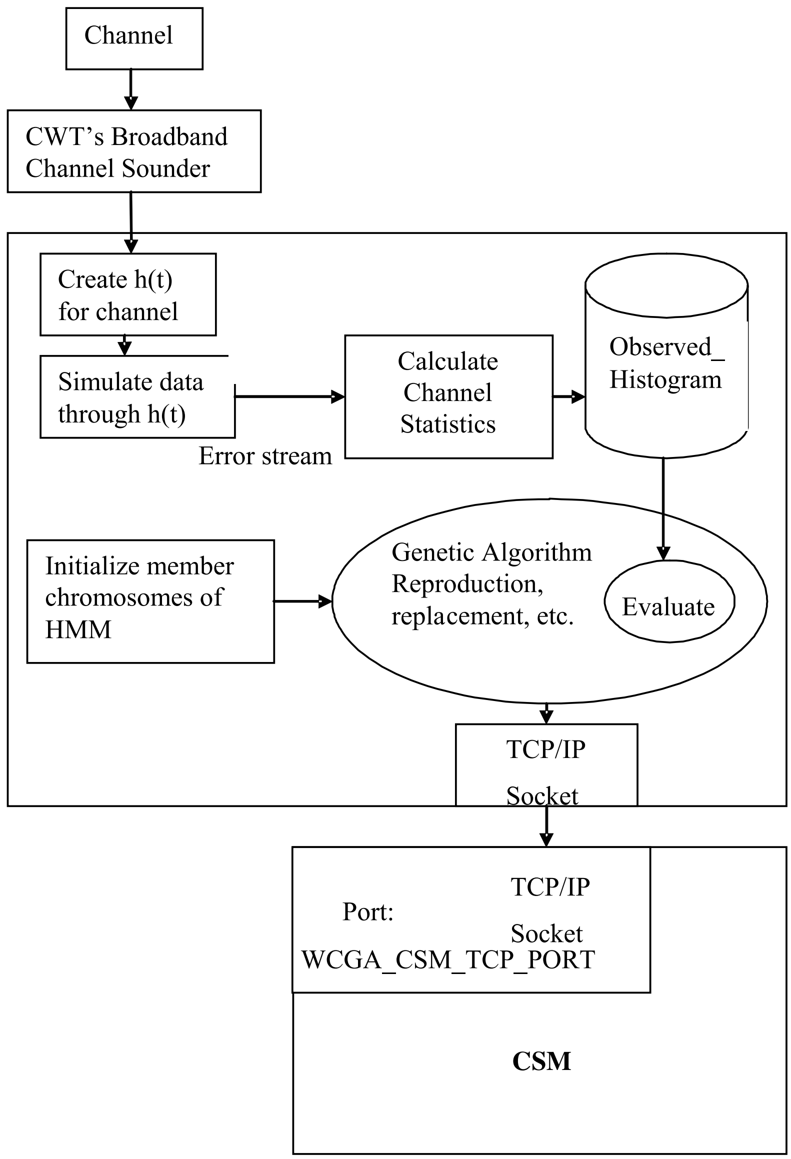


Figure 3.5: Wireless channel genetic algorithm (WCGA) block diagram

The Wireless Channel Genetic Algorithm (WCGA) shown in Figure 3.5 uses error stream data to train an HMM of the wireless channel and generates symbol level channel statistics. Error streams may be derived directly from a training sequence, a captured error symbol stream, or symbol level behavior that corresponds to an impulse response of a channel. The error streams are then used to calculate statistics of the channel, including burst error characteristics. The error statistics provided by the WCGA are then passed to the CSM via a TCP/IP socket to estimate channel performance metrics like minimum bit error rate (BER) or maximum data rate for the observed channel.

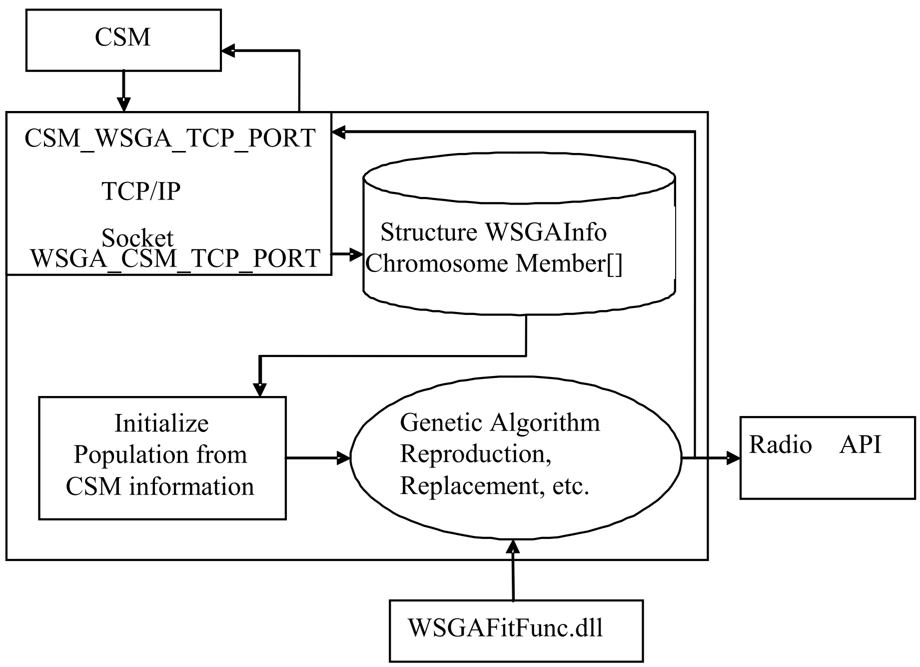


Figure 3.6: Wireless system genetic algorithm (WSGA) block diagram

The WSGA receives input from the CSM via a TCP/IP socket. As shown in Figure 3.6, information about the WSGA is stored in a structure called WSGAInfo. Member chromosomes of the genetic algorithm are stored in Member[]. The member chromosomes are then initialized. A genetic algorithm is used to determine the new radio system parameters, which links to a dynamic link library (DLL) to retrieve the mathematical fitness functions. The final solution from the genetic algorithm is transmitted to the radio via a radio-specific application program interface (API).

The WSGA chromosome structure is shown in the following table.

Table 3.1: WSGA Chromosome Parameters

Chromosome parameters	
0	Power
1	Carrier Frequency (Fc)
2	Bandwidth
3	Symbol Rate
4	Modulation
5	Forward Error Correction (FEC)
6	Payload/frame length
7	Automatic Repeat Request (ARQ)
8	Dynamic Range
9	Equalization
10	Encryption
11	Antenna Configuration
12	Voice
13	Noise Cancellation (limiting)
14	Interference Temperature
15	Time Division Duplex (TDD)
16	User defined host radio parameter 1
•	•

Chromosome parameters	
•	•
•	•
31	User defined host radio parameter N

The values of this table correspond to the genes of the chromosome used in the WSGA. Each gene is a specific knob, or parameter, of the radio. This table is viewed as a vector in the algorithm and operated upon as a chromosome through genetic algorithm crossover and mutation procedures. The values of these table parameters determine the fitness of the chromosome and the behavior of the radio.

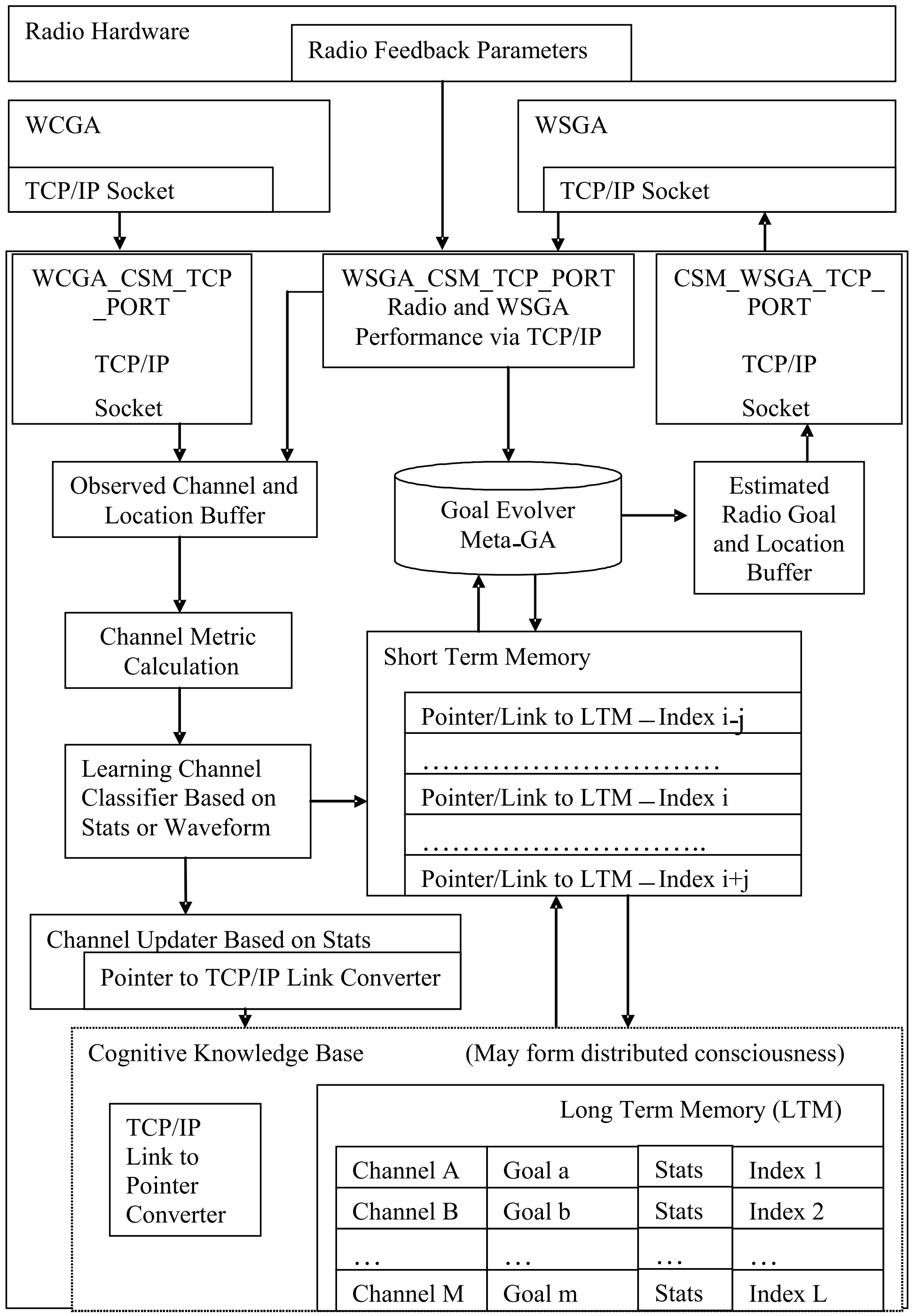


Figure 3.7: Cognitive system monitor (CSM) block diagram

The CSM shown in figure 3.7 receives input from both the WCGA via a TCP/IP socket and the host radio via the WSGA performance API. This information is stored in the observed channel and location buffer. The data in this buffer are then supplied to a channel statistics processor. The statistics computed by the processor are passed to a learning channel classifier which outputs the channel index match to short term memory and the channel updater. The channel updater passes a pointer for the channel match to the TCP/IP link converter attached to long term memory (LTM) so that LTM can be updated. The LTM interfaces with the short term memory (STM) so that the goal evolver can operate on the closest channel matches when synthesizing the appropriate goals for the observed channel. The goal evolver also receives input from the radio and WSGA performance API. This information is used by the goal evolver to provide an output to the estimated radio goal and location buffer. The data in this buffer is provided to the WSGA.

3.5 BioCR Algorithms

This section describes the various CR procedures, including the WCGA, WSGA, and CSM.

WCGA

The WCGA module in Figure 3.8 illustrates a recent implementation of the sensing and modeling blocks within the cognitive engine [67]. The WCGA receives channel information from the CWT's Broadband Sounder. The sounder is used to capture the channel impulse response. The mathematical representation $h(t)$ of the channel impulse response is developed and used to generate an error symbol stream. The channel statistics are calculated from this error stream and a histogram is stored describing the symbol burst errors. A population of HMM chromosomes is randomly initialized. A genetic algorithm loop is then initiated and run until a specified stopping criteria like reaching a maximum number of generations (the currently used method), or a certain minimum desired fitness value. Any GA selection process may be used to choose parents for mating, including tournament or roulette wheel selection. In a tournament selection, each parent is chosen in sets of N players at random and then the best individual out of that set is chosen to be a parent based on their fitness value. In a roulette selection, each individual is assigned an area on a wheel corresponding to that individual's probability. A random number is generated and used to select one of regions on the wheel which corresponds to the individual. The parents are then genetically manipulated through crossover and mutation to create a new set of offspring. The offspring are then evaluated. The worst members of the current generation are replaced by more fit offspring, and the entire population is evaluated based on their fitness values. The percent of worst members replaced may be calculated from values in the GA parameters file (number of adults replaced each generation / population_size * 100). The WCGA was configured to replace 75 % of its population each generation, 15 out of every 20 population members. The best fit member of the population can then be used to determine if the stopping criteria are met. If the stopping criteria are met, the GA exits and the best fit member of the population is the channel model transmitted to the CSM, which exits the WCGA routine.

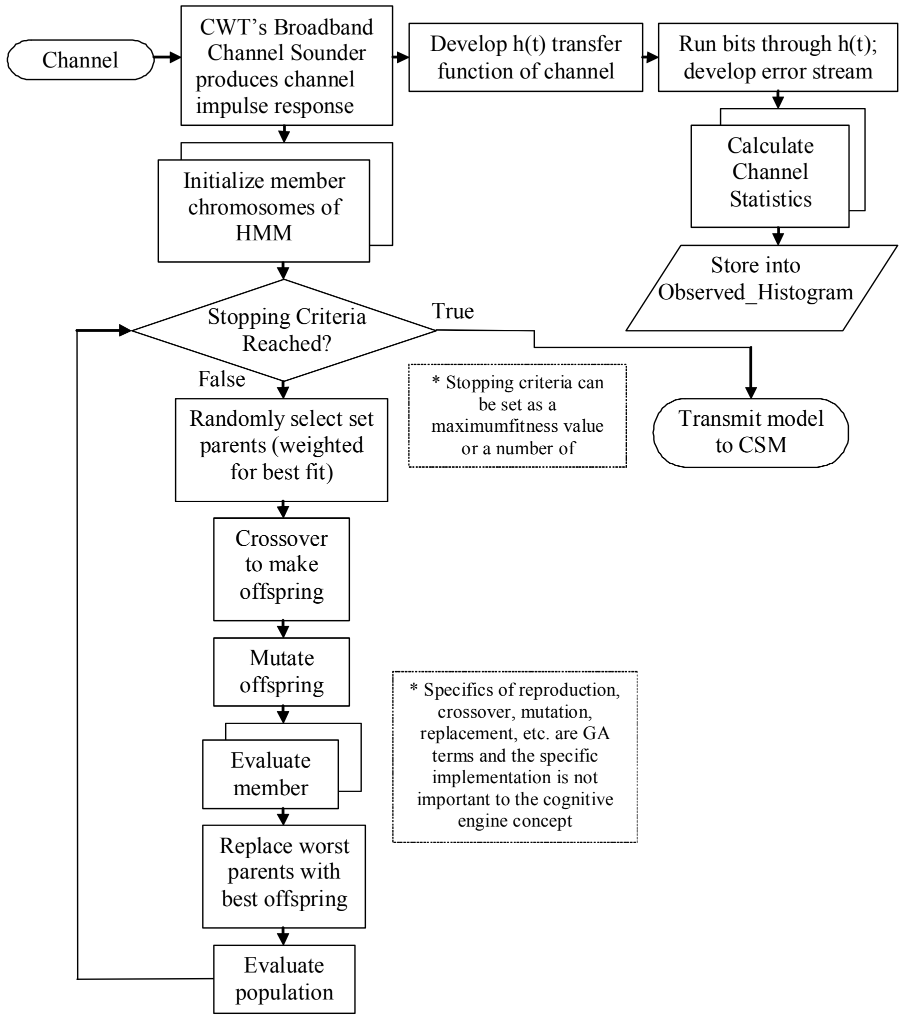


Figure 3.8: Wireless channel genetic algorithm (WCGA) flowchart

WSGA

The WSGA module in Figure 3.9 implements the adaptation block within the cognitive engine. The WSGA receives a packet from the CSM containing the WSGA goals which is temporarily stored. The population of chromosomes is then initialized. A decision block which controls the genetic algorithm loop then checks for stopping criteria and exits the loop upon finding one, which could be a certain number of generations or after a decrease in performance gain per generation is detected (that is, the fitness of the current generation did not differ significantly from the previous generation). While the loop is running, parent chromosomes are selected that will be used to generate offspring chromosomes to replace the population the next generation. The WSGA proceeds to perform standard genetic algorithm techniques of crossover and mutation in an effort to optimize radio parameters for a given set of goals. The fitness values for each chromosome are evaluated for both parent and offspring based on a relative fitness evaluation method which determines which members of the population to replace. The percent of worst members replaced may be calculated from values in the GA parameters file (number of adults replaced each generation / population_size * 100). The WSGA was configured to replace 85 % of its population each generation, 17 out of every 20 population members. Once the genetic algorithm loop has exited, the system parameters contained within the best fit chromosome of the final generation are transmitted to the radio via an API. The best fit chromosome is also transmitted along with the simulated fitness values to the CSM so the CSM can compare the simulated fitness values to the real fitness values read from the radio after the new radio settings have been set.

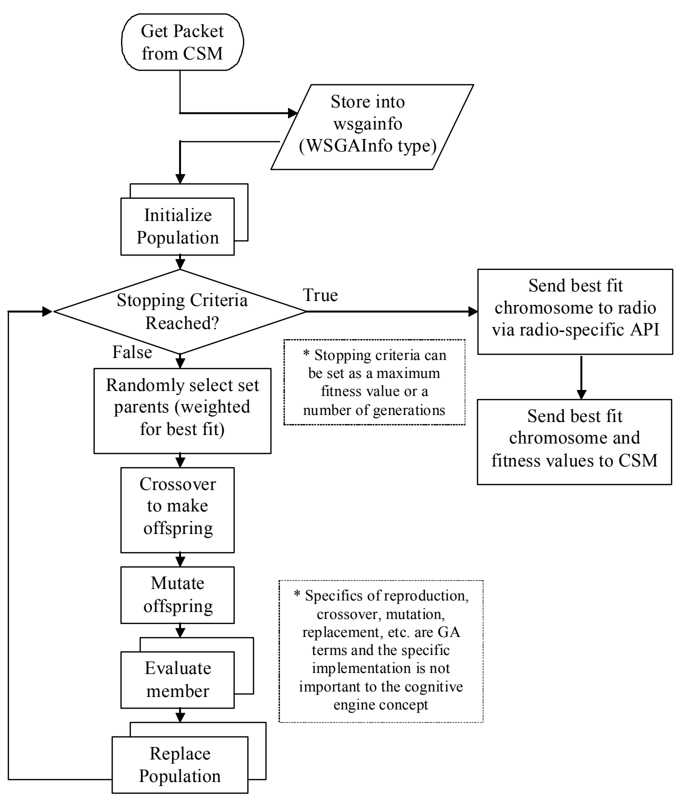


Figure 3.9: Wireless system genetic algorithm (WSGA) flowchart

CSM

The CSM module in Figure 3.10 implements the cognition block within the cognitive engine. The CSM receives a channel model and statistics from the WCGA and stores this in the observed channel and location buffer. The channel statistics processor then calculates the statistics of the observed channel and passes that information to the learning channel classifier which classifies the observed wireless channel by either statistics or waveform. The learning classifier then finds the closest match in Long Term Memory (LTM) by GA channel index scan or a binary search and updates the LTM, letting the Goal Evolver know that a change has been observed in the wireless channel. The Short Term Memory (STM) is populated with chromosomes from the LTM containing similar channels compared by statistics or waveform.

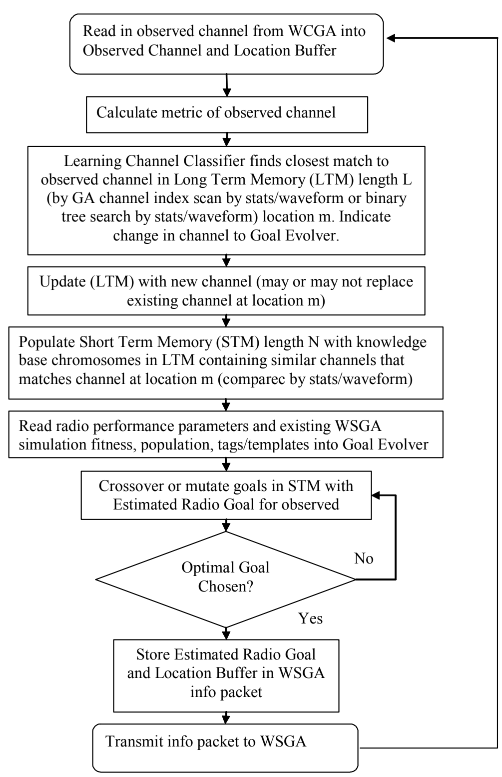


Figure 3.10: Cognitive system monitor (CSM) flowchart

The radio performance parameters are read with the existing WSGA simulation fitness function, population, tags, and templates into the Goal Evolver. The Goal Evolver then uses crossover and mutation of goals in the STM to synthesize the estimated radio goal for the observed channel and calculate an estimated goal value based on the estimated statistics calculated for the observed channel. The percent of worst members replaced may be calculated from values in the GA parameters file (number of adults replaced each generation / population_size * 100). The CSM was configured to replace 85 % of its population each generation, 17 out of every 20 population members.

The resulting goal vector is stored in a buffer and transmitted to the WSGA for radio evolution and optimization to begin until another change in the wireless channel is observed. The LTM knowledge base may function as a distributed consciousness, providing a flexible structure for developing location and temporal specific data. The flexibility of the GA gene functions and chromosome structure allows the formalism to adapt to any host radio system, even legacy radios with minimal adaptability. More flexible programmable wireless systems like software radios showcase the power of the distributed cognitive engine.

3.6 Summary

This chapter presented the details my Ph.D. research proposal, including model, framework, architecture, and algorithms. The WCGA, WSGA, and CSM were introduced and the system design for the cognitive engine was discussed. Chapter 4 presents the methodology I used for my experiments.

Chapter 4: Methodology for experiments

As an initial proof of concept application, I assumed that the cognitive radio engine would be used in a rapidly deployable broadband wireless disaster communications environment. This chapter describes the methodology used to set up hardware and software simulation experiments which test the cognitive engine in electromagnetic environments that might exist during a disaster. This is a brief chapter, serving only to set the stage for Chapters 5 and 6 which provide extensive details about the individual experiments and how they work.

4.1 Methodology for Experimental Study

My research used simulation as the primary methodology for the experimental study of the BioCR engine model. I created a set of experiments that test the behavior of the engine in various unanticipated wireless environments created in the CR simulation test bench. In addition, some functionality of the BioCR engine model was tested on a radio host platform with limited adaptive capabilities.

Chapter 5 presents a scenario and simulation platform that shows how the cognitive engine could evolve the radio's operation in the face of unanticipated wireless channels, like those found in rapidly deployable emergency communications situations. I architected a symbol level simulation test bench shown in Figure 4.1 to emulate an adaptive radio that could serve as the host to the cognitive engine.

The test bench consists of a co-simulation that enables a C/C++ compiled implementation of the BioCR Engine code to run inside of a simulated adaptive radio host implemented in MATLAB-Simulink.

The simulation was designed with a set of wireless channels that could be activated throughout the simulation to mimic changing wireless channels. The cognitive engine used an API to configure the adaptive radio in MATLAB-Simulink and read performance metrics from MATLAB-Simulink into the cognitive engine. The API permitted the same core cognitive engine code to be embedded in both the simulation and hardware platforms. Required software included MATLAB R13 version 6.5.1, Simulink, Communications Blockset, Communications Toolbox, DSP Blockset, and DSP Toolbox.

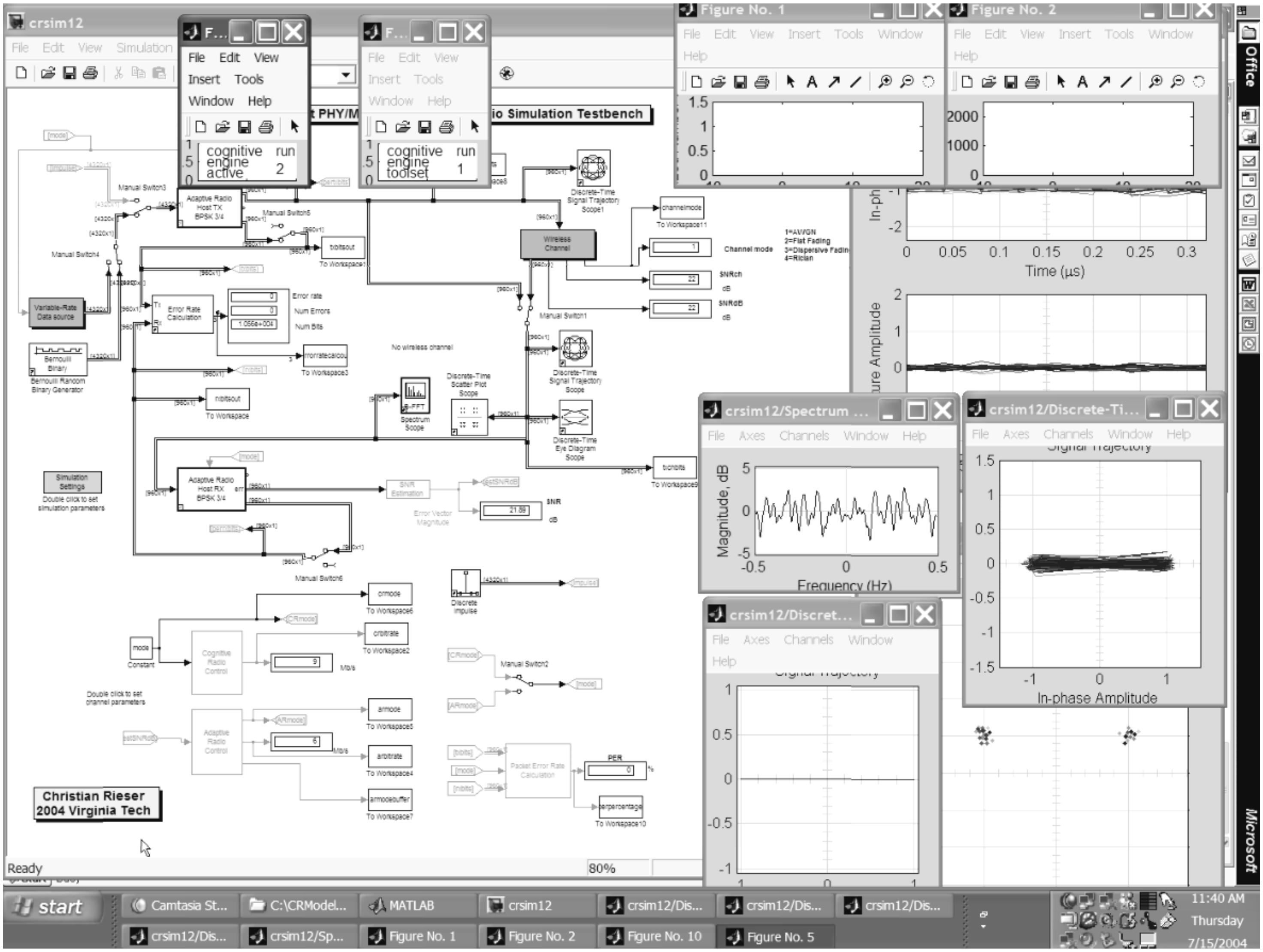


Figure 4.1: Photo of simulation test bench design

Chapter 6 discusses a test bed developed to explore the behavior of a cognitive engine in an actual disaster response communications system. This test bed was developed prior to the CSM code base completion, so the experiment served as a test of the WSGA. Our cognitive radio team built a cognitive radio hardware test bed shown in Figure 4.2 based on legacy broadband wireless communications equipment used by the disaster response community. Virginia Tech chose the fixed broadband wireless Proxim Tsunami radio [68] as host for the cognitive algorithms because the disaster communications community expressed interest in deploying that vendor hardware solution with our value-add research components.

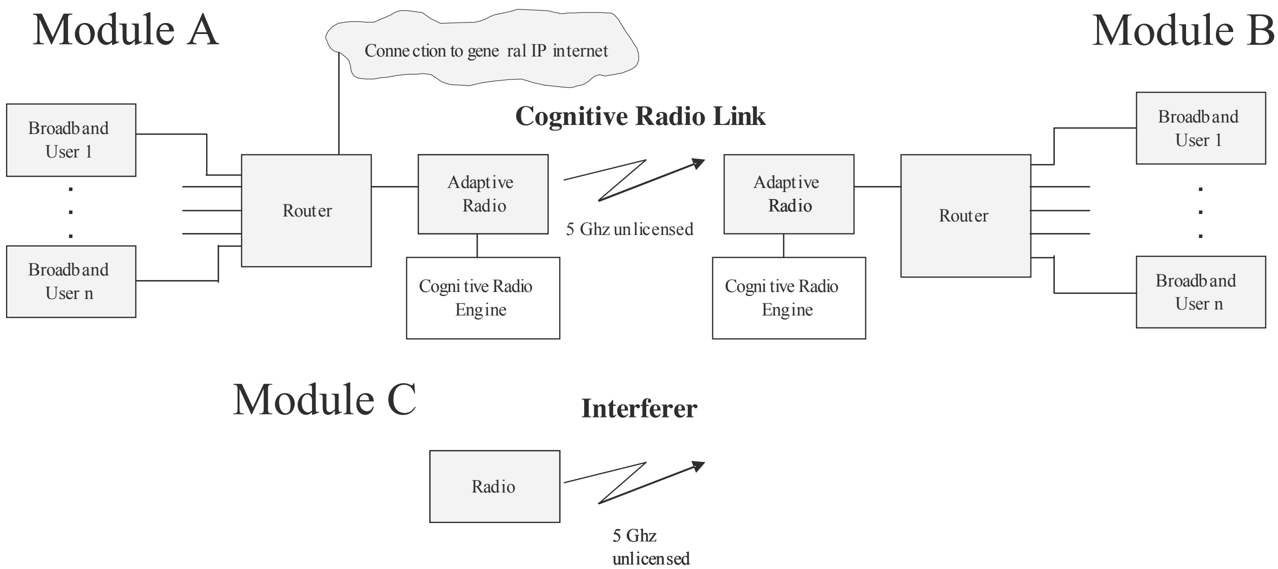


Figure 4.2: Photo of hardware test bed design

These radios had limited programmable features and lacked access to key low level performance information. Even so, the test was valuable in that allowed us to explore how the cognitive engine performed in a legacy system. An experiment was devised by our team to test how the cognitive engine reacted on a hardware platform that was subject to intense interference and signal jamming. Such a caustic wireless channel could be due to the destruction and resulting malfunction of infrastructure that might occur in a natural disaster or attack on the homeland.

These radios have a limited number of knobs and meters compared to the simulated CR test bench environment, but this demo shows how the cognitive engine behaves in a real world system.

4.2 Modeling of Channel Variations in the Simulator

My research used two different ways of modeling how wireless channel variations resulted in symbol or packet errors: statistic symbol error estimating and hidden Markov models (HMMs).

Early in my research I used Hidden Markov Models that modeled wireless channels at the symbol level to rapidly simulate how the cognitive engine responded to symbol errors introduced by different wireless channels [69]. For more information on symbol error channel modeling using HMMs please refer to Chapter 2. Due to MATLAB-Simulink's ability to directly capture and analyze symbol level information from a simulation, I chose not to use the HMM modeling technique for my final experiments.

The BioCR Toolset simulation used statistical distributions of Additive White Gaussian Noise (AWGN), flat fading, dispersive fading, and Rician channels to reflect the impact those wireless channels could have on the decision statistic of a symbol [70][71][72].

AWGN wireless channels are defined as channels that contain noise whose frequency spectrum is continuous and uniform over a specified frequency band. A flat fading wireless channel may be observed when frequency components of a received radio signal vary in the same proportion simultaneously. A dispersive fading wireless channel may be observed when transmitted energy arrives at the receiver at different times, superimposed on other symbols. Both flat and dispersive fading channels are modeled in this simulation using Rayleigh fading channels. Rayleigh fading wireless channels may be observed when phase-interference fading occurs caused by multipath. The resulting channel behavior may be approximated by the Rayleigh distribution. Rician fading occurs when a Rayleigh fading channel exists with a strong line of sight component. The resulting channel is said to have a Rician distribution.

I decided to use the channel models that shipped with the MATLAB-Simulink Communications Blockset. The purpose of the experiment was to test the cognitive engine's performance in changing and unknown channels, so any of the preprogrammed channels would be satisfactory. I was able to delineate between channels that were known and unknown to the radio by "priming the pump" with several statistical descriptions of available wireless channels that were inserted into the engines long term memory (LTM) on boot up. To generate a scenario where the engine encountered an unknown channel, I simply switched in a wireless channel with statistical properties that were not in LTM. This was accomplished by leaving LTM empty on engine initialization. In this case every channel was unknown until a channel was encountered for a second time. This methodology was used in simulator. The following channels were encountered: AWGN, Flat fading, Dispersive fading, Rician. Then the simulation experiment was programmed to encounter another known channel, in this case an AWGN channel.

4.3 Summary

This chapter provided a brief overview of the methodology used for the software and hardware experiments detailed in the following chapters. A description is provided of how the channel variations are generated, including definitions of the various wireless channel models. The next two chapters provide detailed analysis of the software simulation and hardware experiments.

Chapter 5: Results from Virginia Tech CR Simulation Test Bench Experiments

This chapter presents the results of a simulation test bench I created that demonstrates how a cognitive engine could evolve a radio's operation in the face of unanticipated wireless channels, like those found in rapidly deployable emergency communications situations. Appendix E contains detailed logs of BioCR toolset simulation runs, including captured data from both the host radio and cognitive engine. An explanation of each trend step of the simulation run is discussed. A "trend step" is a simulation mechanism which freezes time and shows what is going on under the hood of the cognitive engine at that instant. The simulation toolset facilitates this analysis through extensive time stamped data logging of the simulated adaptive radio and cognitive radio engine output.

5.1 Simulation of CR Engine Model versus Traditional Adaptive Radio Controller

One of the goals of this research was to create a cognitive radio model that could operate in unanticipated wireless channels. This chapter presents the results of a simulation I created to test the implementation of the cognitive radio engine model presented in this dissertation and compare it to a traditional adaptive radio controller available in MATLAB-Simulink [73].

In the case of this experiment, a traditional adaptive controller changes the data rate of the system based on the measured signal to noise ratio (SNR) using a delta search method. The traditional adaptive controller uses a state-machine controller by thresholds. The controller starts in state one, the lowest data rate and corresponding modulation. If the measured SNR exceeds the SNR needed to support error free data demodulation for the given modulation, the adaptive controller increases the data rate by one modulation index. Figure 5.1 shows the low SNR thresholds required to support error free transmission of data using a given modulation in the simulation. The traditional adaptive controller ratchets the data rate up and down with changing channel SNR. Due to the adaptive controller's delta search and slow response time it may not be able to leverage available spectrum so as to provide "instantaneous bandwidth" on demand. The cognitive engine uses its learning process to deduce from the unknown channel that the radio can tap into available wireless spectrum at higher data rates.

Adaptive Controller

LOW SNR THRESHOLDS

[SNR	Data Rate	Modulation	Index]
[-	6250000	BPSK 1/2	1]
[10	9375000	BPSK 3/4	2]
[11	12500000	QPSK 1/2	3]
[14	18750000	QPSK 3/4	4]
[18	25000000	16-QAM 1/2	5]
[22	37500000	16-QAM 3/4	6]
[26	50000000	64-QAM 2/3	7]
[28	56250000	64-QAM 3/4	8]

Figure 5.1: Table of thresholds used by adaptive controller

In contrast, the cognitive engine is capable of evolving the operation of an adaptive radio host. Figure 5.2 illustrates this mechanism. A radio transmitter and receiver communicate by transmitting and receiving symbols over a wireless channel. The simulation collects statistics about the symbol errors that occur in this link and passes those statistics to the cognitive engine. The cognitive engine reads the current radio settings called "old knobs" along with appropriate radio performance metrics called "old meters." This information used by the cognitive engine to establish operational goals for the radio which are used to generate new optimal radio parameters settings, "new knobs." The radio is configured with these new operational parameters and the process begins anew.

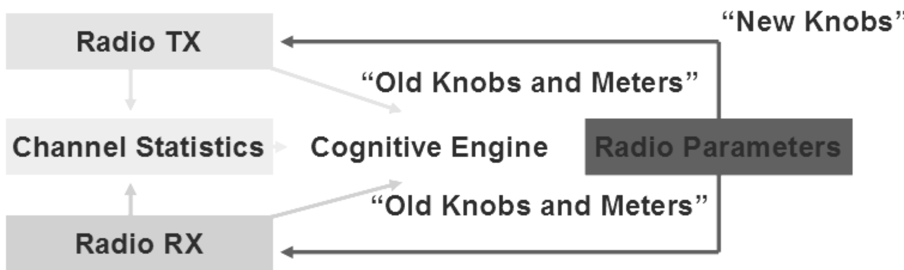


Figure 5.2: Basic explanation of cognitive engine operation

Figure 5.3 provides a more detailed example of this process. In this fictional example a binary symbol error stream is captured by the simulation and passed to the WCGA, which generates channel statistics in the form of a burst error histogram, [.7 .2 .05 .05 0]. This fictional example histogram may be interpreted that 70% of the time the channel had errors of burst length one, 20% of the time the channel had errors of burst length two, 5% of the time the channel had errors of burst length three, 5% of the time the channel had burst length four, and 0% of the time the channel had errors of burst length five or longer. This tells the researcher that most errors are single errors, information that may be useful in configuring the radio for reliable performance in this channel. The radio then reports current knobs and meters. In this case the radio is currently using 64 QAM modulation and a transmit power of 9 dBm with a BER of 10^{-4}.

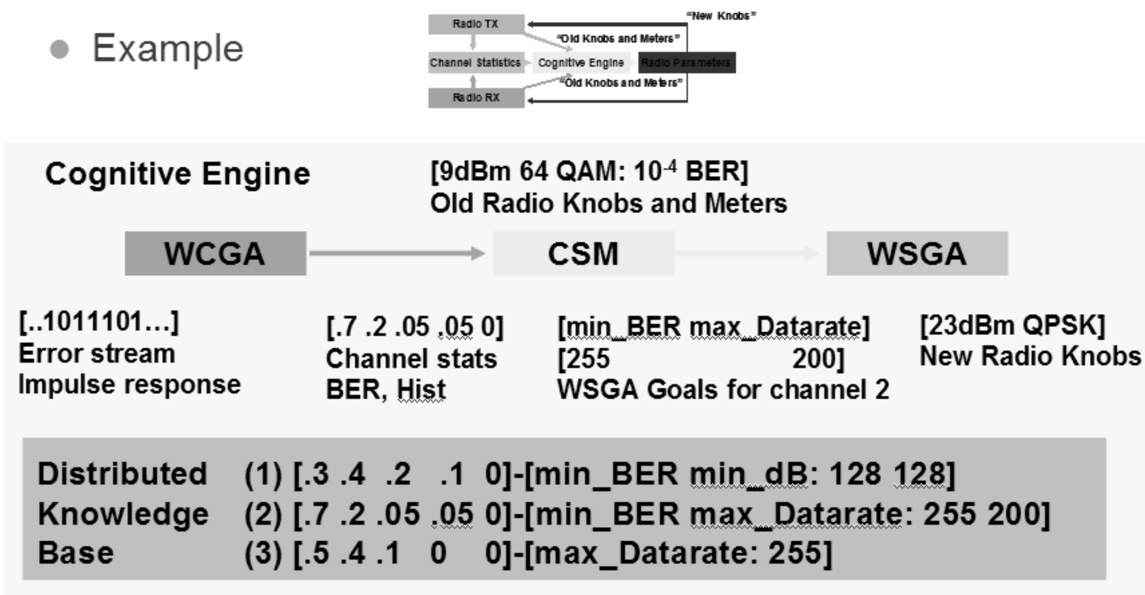


Figure 5.3: Basic explanation of cognitive engine process

The cognitive engine then classifies the observed channel to see if there is a match in the long term memory (LTM) which serves as a distributed knowledge base. The LTM in this example consists of a pair of channel statistics and WSGA goal entries. The CSM also maintains control data not shown in this basic example. In this example, the CSM classifier function finds an exact match in LTM index location two. The corresponding WSGA goal information is read from memory; in this case the CSM tells the WSGA to optimize the radio parameters so that bit error rate (BER) is minimized and data rate is maximized. These fitness functions and priorities may change if the CSM decides to evolve the recommended goals passed to the WSGA to increase system performance. The goal evolver mechanism was proposed to enable the learning optimizer mode of the engine. In order to validate the engine's basic learning capability, the goal evolver module was not permitted to autonomously change channel-goals pairs on the fly because of issues related to tracking the dynamic changes the experiment made to itself. Future dynamic experiments could be architected in a way to observe the goal evolver's actions with the autonomous learning process operational. Note that the engine instructs the WSGA to prioritize minimizing BER by assigning that fitness function a larger relation weight of 255 compared to the task of maximizing the data rate, which has a relative fitness function weight of 200. With these fitness functions and priorities, the WSGA generates a new set of radio knobs, increasing the power to 23 dBm and decreasing the modulation index to QPSK. Note that WSGA met the goals given to it by the CSM of minimizing BER, its main priority, while attempting to maximize its data rate, its secondary priority. The engine performs the tradeoff analysis on it's own to balance the goals of the system. When tasked with top priority of minimizing BER, most adaptive radios would reduce the modulation to BPSK as this modulation performs well in challenging channels, thereby limiting data throughput in a channel that the cognitive engine had learned could support higher data rates. The cognitive engine instead was able to learn to generate radio parameters that balanced the needs of the system. What is interesting is that this learning mechanism is immediately applicable to changing system needs based on changing channels, a key challenge in disaster communications systems.

The cognitive radio operation and process detailed in Figures 5.2 and 5.3 were used to design the experimental simulation shown in Figures 5.4 and 5.5, which provide an overview and details of the cognitive radio simulation test bench.

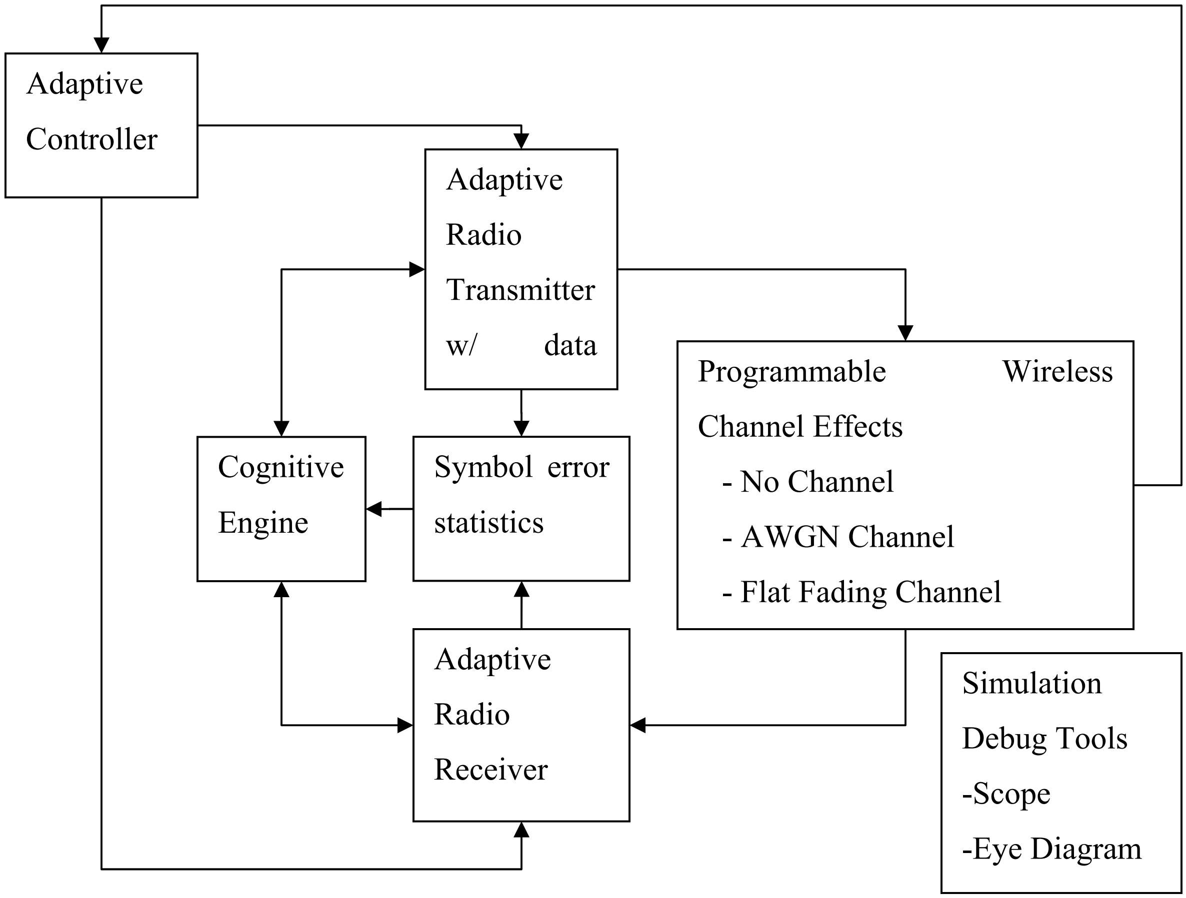


Figure 5.4: Overview of adaptive radio host simulation in Simulink

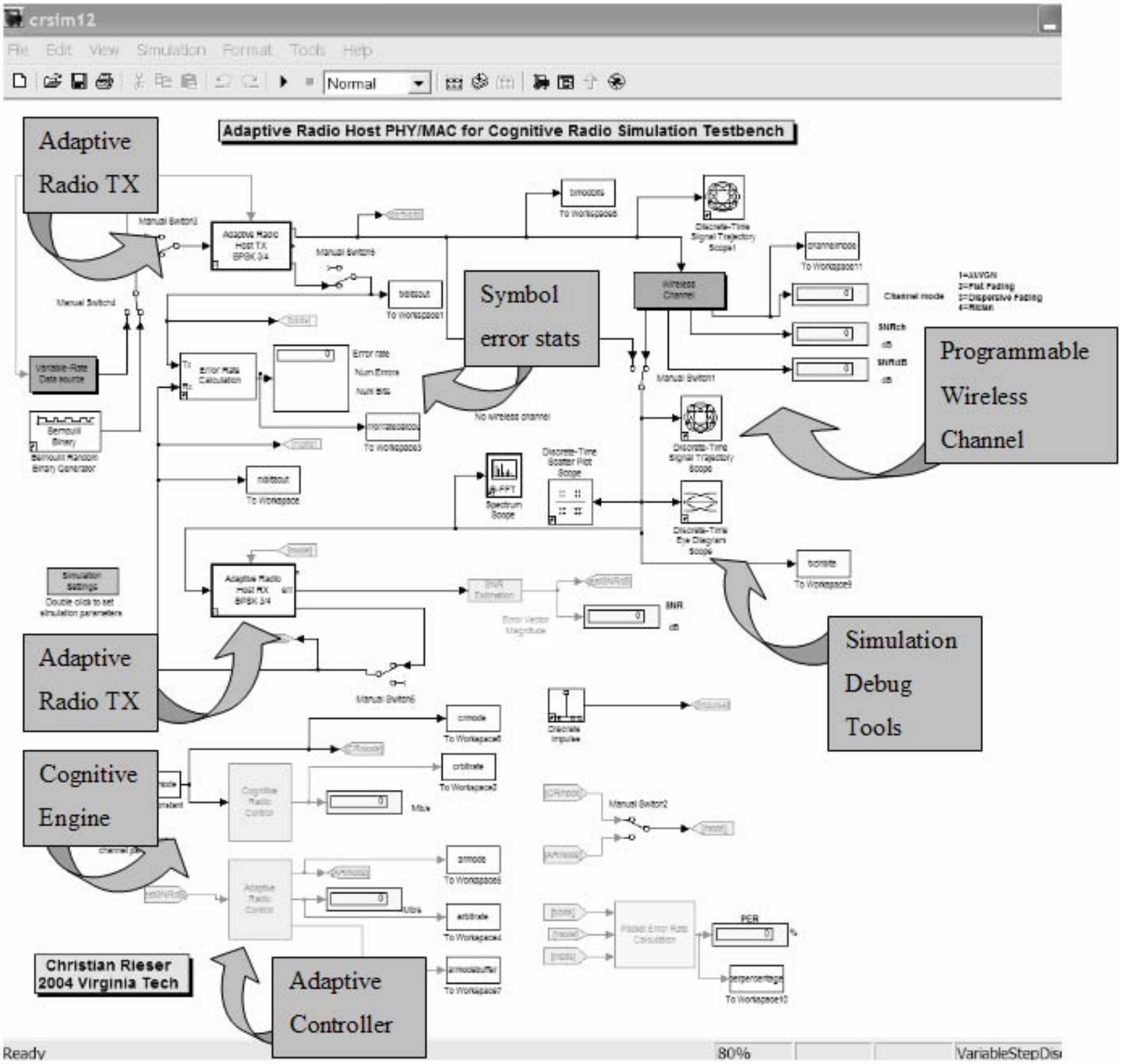


Figure 5.5: Adaptive radio host simulation in Simulink

Figure 5.5 shows a simulation I created in MATLAB-Simulink of an adaptive radio host operating in several different wireless channels [74][75][76]. The adaptive radio host may be configured to be controlled by the cognitive engine or a traditional adaptive controller.

The simulation consists of a data source, adaptive radio transmitter, various wireless channels, an adaptive radio receiver, and a block that calculates the radio performance metrics like symbol error rate and histograms plotting the distribution of burst error lengths for an observed wireless channel. The cognitive engine operates inside of the simulation, providing control of the adaptive radio host by automatically configuring the radio in response to changing wireless channels. A traditional adaptive controller is included in the simulation for reference. MATLAB code is included in the CR Toolset to provide post simulation analysis and data archival functions.

The following data are time stamped and recorded in the "berdata" directory of the CR toolset directory every time the cognitive engine completes an iteration in the simulation:

(1) MATLAB workspace .MAT file with simulation variables and values
(2) Radio performance plots in .JPG form that show the absolute and relative error performance of the cognitive radio. These error histograms show how the cognitive radio responds to errors introduced by the wireless channel.
(3) Absolute and normalized error distributions in .CSV format that correspond to the .JPG plots
(4) WCGAinput.csv file produced by the adaptive radio to serve as input to cognitive engine
(5) Snapshot of the cognitive engine's current long term memory (LTM), ltmstat.csv
(6) SystemKnobs.csv file produced by the cognitive engine to control the adaptive radio simulation test bench
(7) SystemMeters.csv file produced by the adaptive radio to serve as input to the cognitive engine
(8) WSGAActions.csv file read by the cognitive engine that initializes the values of LTM in the cognitive engine with channel/goal pairs, including WSGA fitness functions and weights
(9) WSGAFinalOutput.csv file produced by the cognitive engine to control the adaptive radio hardware test bed
(10) Error stream sequence captured by simulation
(11) CSM Final Output that shows details of the cognitive engine operation
(12) CSM parameters file that controls the CSM operation
(13) WSGA parameters file that controls the WSGA operation

(14) WCGA parameters file that controls the WCGA operation

(15) System chromosome produced by the cognitive engine. This is interpreted by the cognitive engine into radio parameter settings

Numerous demonstrations of the cognitive radio have been conducted that illustrate the cognitive radio engine model controlling a simulated adaptive radio host in changing wireless channels that are both unknown and known, like those found in disaster communication scenarios (over 500 MB of data and 25,000 archive files). The alpha release toolset has been run for long simulations (over 12 hours) to collect data on the cognitive engine's ability to learn and self evolve its behavior in changing wireless channels within the legal constraints of which it is aware.

A demonstration was put together to compare the cognitive radio engine model and framework I propose to a traditional adaptive controller model and framework. This simulation was created to explore how the cognitive engine behaved in known and unknown wireless channels, and how this behavior compared to a traditional adaptive controller. The adaptive controller was compared to the cognitive radio by observing which mechanism allowed the radio to maximize performance in changing wireless channels, specifically for a given SNR which mechanism provided a higher throughput for the user? How quickly could each mechanism leverage changes in the channel? These comparisons were made for each trend step, since each trend step corresponded to a potential change in the channel and SNR of the system.

To simulate a dynamic wireless environment, the simulation was designed to have the opportunity to switch between AWGN, Flat Fading, Dispersive Fading, and Rician channels at the beginning of each trend step. Both the adaptive controller mechanism and cognitive engine were subjected to these known and unknown wireless channels that switched in time and system performance was logged.

The results of the demonstration show that the cognitive engine finds the best tradeoff between a host radio's operational parameters in changing wireless conditions, while the baseline adaptive controller only increases or decreases its data rate based on a threshold, often wasting usable bandwidth or excess power when it is not needed due its inability to learn.

At this time the demonstration is a point-to-point experiment, however with the baseline cognitive radio engine code operational; future research could be pursued to expand the toolset to explore the behavior of the engine in a cognitive radio network. This future research is the subject of a recent grant proposal by Virginia Tech. Since the engine was proposed and implemented as a fully distributed model and algorithmic framework, in the future our research team intends to extend the cognitive radio simulation test bench and hardware test bed to create and study cognitive wireless networks as follow on work to my research. The current implementation of the cognitive engine already has hooks in it to allow networked peer to peer communications using the same code framework that the different modules currently use to coordinate their operation.

The remainder of this chapter is dedicated to presenting and analyzing that demonstration. The reader will have an opportunity to look “under the hood” of the cognitive engine and see its learning process in action and how it behaves in unanticipated wireless environments. The demonstration is available via website [77] with sample data dumps from the toolset and a performance trace showing a behavior profile for a variable wireless environment.

5.2 CR Engine Performance in an Unknown Channel

This section details how the cognitive engine responds to unknown channels. Figure 5.6 shows a trace of the cognitive engine operating in four different channels. In this case all four channels were unknown to the engine when it began its operation.

(1) Additive White Gaussian Noise (AWGN) wireless channel

(2) Flat Fading Rayleigh wireless channel

(3) Dispersive Fading Rayleigh wireless channel

(4) Rician wireless channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1 = AWGN	1 = BPSK 1/2 rate	6250000	1	0.057202
2	1 = AWGN	2 = BPSK 3/4 rate	9375000	22	0
3	1 = AWGN	4 = QPSK 3/4 rate	18750000	19	0
4	1 = AWGN	2 = BPSK 1/2 rate	9375000	9	0
5	2 = Flat fading	8 = 64-QAM 3/4 rate	56250000	25	0.000174
6	2 = Flat fading	4 = QPSK 3/4 rate	18750000	26	0
7	2 = Flat fading	5 = 16-QAM 1/2 rate	25000000	19	0.000434
8	3 = Dispersive fading	2 = BPSK 3/4 rate	9375000	22	0
9	3 = Dispersive fading	6 = 16-QAM 3/4	37500000	9	0.10041
10	3 = Dispersive fading	4 = QPSK 3/4 rate	18750000	26	0.13042
11	3 = Dispersive fading	4 = QPSK 3/4 rate	18750000	26	0.13042
12	3 = Dispersive fading	4 = QPSK 3/4 rate	18750000	26	0.13042
13	4 = Rician	4 = QPSK 3/4 rate	18750000	26	0
14	4 = Rician	8 = 64-QAM 3/4 rate	56250000	23	0
15	4 = Rician	6 = 16-QAM 3/4	37500000	9	0.006688
16	4 = Rician	4 = QPSK 3/4 rate	18750000	9	8.68E-05

Figure 5.6: CR toolset trace showing cognitive engine reacting to unknown channel

Figure 5.6 provides a trace of the simulation demonstration, showing how the cognitive engine responded to unknown wireless channels in trend steps one through sixteen. In trends 1-4 we see the radios adapting to an AWGN channel. At the end of trend 4 the simulation switched to a Flat fading channel and the radios adapted to that channel in trends 5-7. At the end of trend step 7, the simulation then switched to a dispersive channel and the radios adapted to that channel in trend steps 8-12. At the end of trend step 12, the simulation switched to a Rician channel and the radios adapted to that channel. Various performance metrics are listed like data rate for the given SNR and respective bit error rate (BER) for the channel. This information can provide insight into how well the engine responds to unanticipated wireless channels. Combining with data captured from the cognitive engine's memory, this presents a complete picture of the "thought process" of the cognitive engine for analysis. This section provides that analysis.

The engine operated in an AWGN channel the first four steps of the demonstration (called "trend steps"), then encountered a flat fading channel from trend step five to seven. These trend steps can be thought of as snapshots in time and would normally be part of a fluid real-world demonstration. However, for simulation purposes, freezing time and analyzing the data can be quite helpful. On trend step eight the engine encountered a dispersive fading channel. The engine operated in a Rician channel the remaining four trend steps. Section 5.4 will compare these results to a traditional adaptive controller.

The meanings of the WSGA fitness function codes are listed in Table 5.1. WSGA fitness functions weights may range from 0 to 255.

Table 5.1: WSGA Fitness Functions Used in Simulation

1 =	Minimize AWGN BER
2 =	Minimize Rayleigh BER
3 =	Minimize Rician BER
10000 =	Minimize Power Consumption
20000 =	Maximize Data Rate

The following section provides step by step analysis of the cognitive engine's operation in various unknown channels.

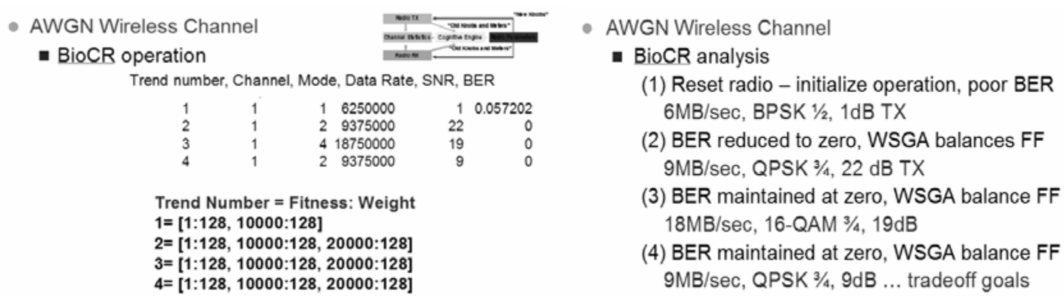


Figure 5.7: Summary of cognitive engine behavior in AWGN channel

Figure 5.7 provides a consolidated view of the first few trend steps while the radio is operating in an AWGN channel. Figures are provided later in this section describing the engine's behavior as it transitions from the AWGN channel to a flat fading channel, then to a dispersive fading channel, on to a Rician channel, and back to an AWGN channel.

A great deal of information is in Figure 5.7 and the companion figures for the other wireless channels the engine encounters. The list of trend steps in the figure documents how the engine responded to the channel it encountered, including the mode of operation which includes modulation and code rate, the data rate, and the transmit power/SNR. The radio's BER performance for that set of radio parameter values is documented. The engine data capture "Trend Number = Fitness : Weight" includes information on how the engine made decisions during each trend step. The WSGA goal information, including fitness functions and priority weights, are listed for each trend step. The goal information is used to provide analysis of the engine's behavior. The information from inside the engine teamed with information about the engine's performance sheds light on the learning process that drives the engine.

As an example of how to go from trend information to conclusions, consider the trend steps in the AWGN channel. After the first initialization step, the engine memory indicates that the engine initially sought to balance goals of minimizing BER and transmit power, while maximizing data rate. Trend step 2 increased power and modulation index, thereby decreasing BER and increasing data rate, at the cost of excess power consumption. Based on the engine's memory, it maintained the goal to balance fitness functions, so it decreased the transmit power, and increased the data rate, while maintaining low BER. This indicated that the engine had discovered additional capacity for the link to increase speed while decreasing power. Trend step three further decreased transmit power and increased speed. Since the goals of the engine were equally weighted, it is really up to the random decision of the engine as to what priority takes precedence. In trend step four the engine decided that since all things were equally it would significantly decrease the power, however such a move required lowering the modulation index so as to maintain a low BER value. The right portion of figure 5.7 provides a visual summary of the results of running the engine in each wireless channel.

As a review, I decided to use the channel models that shipped with the MATLAB-Simulink Communications Blockset because they were readily available and were already tested for simulation purposes. Since the purpose of the experiment was to test the cognitive engine's performance in changing and unknown channels, any of the preprogrammed channels would be satisfactory. Channels that were known or unknown to the radio were delineated by "priming the pump" with several statistical descriptions of available wireless channels that were inserted into the engines long term memory (LTM) on boot up. To generate a scenario where the engine encountered an unknown channel, I simply switched in a wireless channel with statistical properties that were not in LTM. This was accomplished by leaving LTM empty on engine initialization. In this case every channel was unknown until a channel was encountered for a second time. This methodology was used in simulator. The following channels were encountered: AWGN, Flat fading, Dispersive fading, Rician. Then the simulation experiment was programmed to encounter another known channel, in this case an AWGN channel.

In trend step 1 shown in Figure 5.7, the engine initialized its operation at data rate 6 Megabits per second (Mbps), radio mode 1, BPSK ½ rate forward error correction (FEC) code with a transmit power of 1 dBm, where dBm is decibel referenced to 1 milliwatt; 0 dBm equals one milliwatt. Note that to simplify the Simulink simulation the radio frequency noise floor was assumed to be 0 dBm, so the transmit power in dBm is also the signal to noise ratio (SNR = Signal Power / Noise Power dB = Signal Power dBm – Noise Power dBm). The bit error rate (BER) performance was poor because of this initial reset of the radio. At this time no learning had occurred.

In trend step 2 shown in Figure 5.7, the engine increased its data rate to 9 Mbps, changed modulation and coding to radio mode 2 QPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) performance was reduced to zero due to the engine's changes in radio parameters. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate.

In trend step 3 shown in Figure 5.7, the engine increased its data rate to 18 Mbps, changed modulation and coding to radio mode 4 16-QAM ¾ rate, and decreased its power to 19 dB. The bit error rate (BER) performance was maintained at zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate.

In trend step 4 shown in Figure 5.7, the engine decreased its data rate to 9 Mbps, changed modulation and coding back to radio mode 2 QPSK ¾ rate, and decreased its power to 9 dB. The bit error rate (BER) performance was maintained at zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate. In this case it traded off lower data rate for lower power since each of its goals was equally weighted.

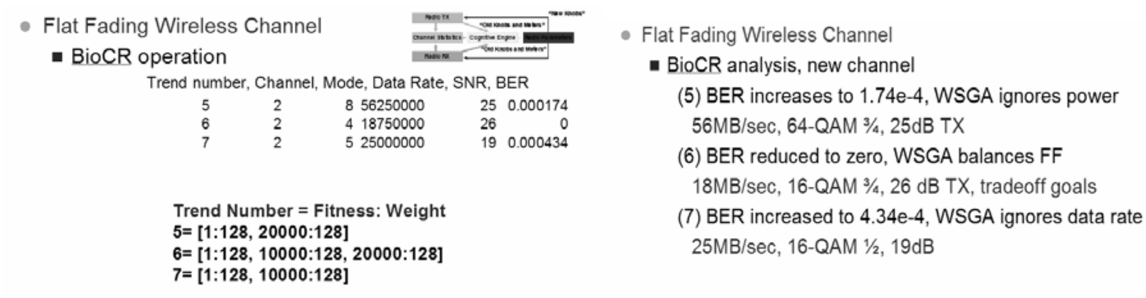


Figure 5.8: Summary of cognitive engine behavior in flat fading channel

In trend step 5 shown in Figure 5.8, the engine encountered a flat fading channel, increasing its data rate to 56 Mbps, changed modulation and coding to radio mode 8 64-QAM ¾ rate, and increased its power to 25 dB. The bit error rate (BER) increased to 1.74×10^{-4} due the more challenging channel. Note that the CSM instructed the WSGA to minimize BER and maximize data rate, in this case ignoring the need to conserve power.

In trend step 6 shown in Figure 5.8, the engine decreased its data rate back to 18 Mbps, changed modulation and coding to radio mode 4 16-QAM ¾ rate, and increased its power to 26 dB. The bit error rate (BER) performance in the flat fading channel was reduced to zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate. In this case the engine traded off lower data rate for better BER performance.

In trend step 7 shown in Figure 5.8, the engine increased its data rate to 25 Mbps, changed modulation and coding to radio mode 5 16-QAM ½ rate, and decreased its power to 19 dB. The bit error rate (BER) increased to 4.34×10^{-4}. Note that the CSM instructed the WSGA to minimize BER and minimize power, ignoring the data rate.

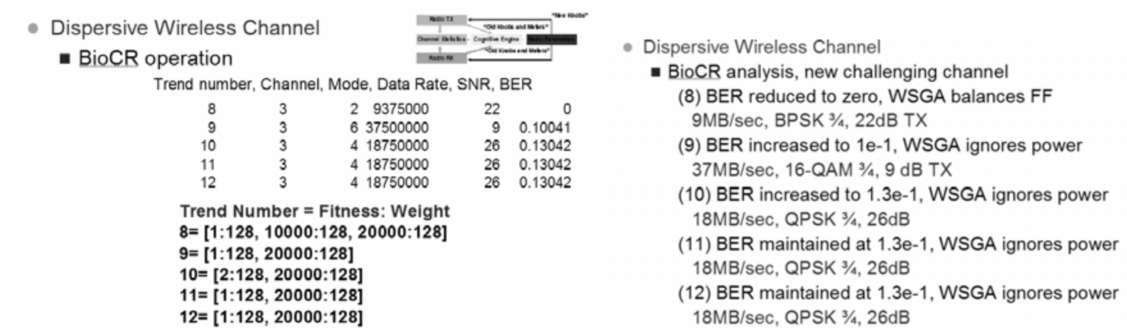


Figure 5.9: Summary of cognitive engine behavior in dispersive fading channel

In trend step 8 shown in Figure 5.9, the engine encountered a dispersive fading channel, decreasing its data rate to 9 Mbps, changed modulation and coding to radio mode 2 BPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) was reduced to zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate.

Note that a problem occurred when simulating the dispersive channel. No software phase lock loop was implemented, so phase angle drift corrections were calculated manually and applied to each channel and modulation. This approach did not work when the dispersive channel generated inter-symbol interference, so the error rate values for the dispersive channels are inaccurate and therefore were disregarded.

In trend step 9 shown in Figure 5.9, the engine increased its data rate to 37 Mbps, changed modulation and coding to radio mode 6 16-QAM ¾ rate, and decreased its power to 9 dB. The bit error rate (BER) increased to $1x10^{-1}$ due to the engine attempting to maximize data rate in a very challenging wireless channel. Note that the CSM instructed the WSGA to minimize BER and maximize data rate, ignoring power.

In trend step 10 shown in Figure 5.9, the engine decreased its data rate to 18 Mbps, changed modulation and coding to radio mode 4 QPSK ¾ rate, and increased its power to 26 dB. The bit error rate (BER) increased to $1.3x10^{-1}$ due to the engine attempting to maximize data rate in a very challenging wireless channel. Note that the CSM recognized the channel as a Rayleigh channel and instructed the WSGA to minimize BER using fitness function 2, maximize data rate, and ignore power.

In trend step 11 shown in Figure 5.9, the engine maintained its data rate at 18 Mbps, modulation and coding to radio mode 4 QPSK ¾ rate, and power at 26 dB. The bit error rate (BER) stayed at 1.3×10^{-1}. Note that the CSM instructed the WSGA to minimize BER, maximize data rate, and ignore power.

In trend step 12 shown in Figure 5.9, the engine maintained its data rate at 18 Mbps, modulation and coding to radio mode 4 QPSK ¾ rate, and power at 26 dB. The bit error rate (BER) stayed at 1.3×10^{-1}. Note that the CSM instructed the WSGA to minimize BER, minimize power, and maximize data rate.

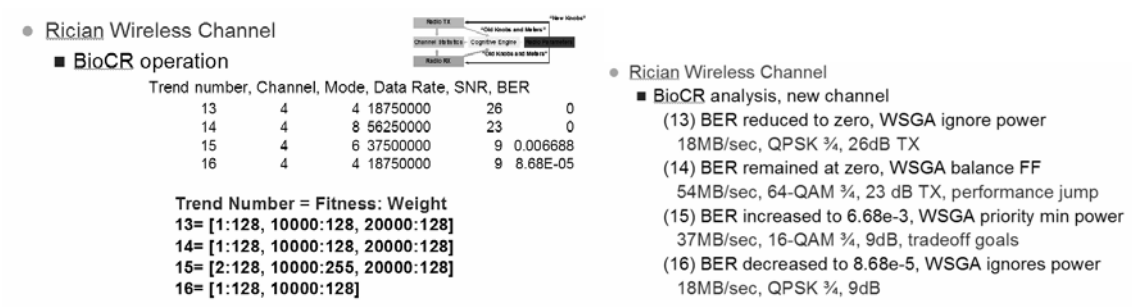


Figure 5.10: Summary of cognitive engine behavior in Rician channel

In trend step 13 shown in Figure 5.10, the engine maintained its data rate at 18 Mbps, modulation and coding to radio mode 4 QPSK ¾ rate, and power at 26 dB. The bit error rate (BER) was reduced to zero because the engine encountered a Rician channel. Note that the CSM instructed the WSGA to minimize BER and maximize data rate, ignoring power.

In trend step 14 shown in Figure 5.10, the engine increased its data rate to 54 Mbps, changed its modulation and coding to radio mode 8 64-QAM ¾ rate, and decreased its power to 23 dB. The bit error rate (BER) remained at zero. Note that the CSM instructed the WSGA to balance fitness functions equally, minimize BER, minimize power, and maximize data rate.

In trend step 15 shown in Figure 5.10, the engine decreased its data rate to 37 Mbps, changed its modulation and coding to radio mode 6 16-QAM ¾ rate, and decreased its power to 9 dB. The bit error rate (BER) increased to 6.68×10^{-3}. Note that the CSM instructed the WSGA to give highest priority to minimizing power, while equally weighting the need to minimize BER and maximize data rate.

In trend step 16 shown in Figure 5.10, the engine decreased its data rate to 18 Mbps, changed its modulation and coding to radio mode 4 QPSK ¾ rate, and maintained its power at 9 dB. The bit error rate (BER) decreased to 8.68×10^{-5}. Note that the CSM achieved this better error rate performance by instructing the WSGA to minimize BER and minimize power, ignoring data rate.

This section demonstrated that the cognitive engine is capable of operating in unknown wireless channels, balancing radio goals based on its ability to learn about the wireless environment it is operating in.

5.3 CR Engine Performance in a Known Channel

The previous section analyzed the cognitive engine's ability to operate in unanticipated and unknown wireless channels. This section analyzes the engine as it encounters a wireless channel that it has seen before, in this case an AWGN wireless channel. Figure 5.11 provides a complete trace of the entire simulation demonstration, showing how the cognitive engine responded to unknown wireless channels in trend steps one through sixteen and a known wireless channel in trend steps seventeen through twenty. Various performance metrics are listed like data rate for the given SNR and respective BER for the channel. This information can provide insight into how well the engine responds to unanticipated wireless channels. The trend steps highlighted in grey in Figure 5.11 show the AWGN channel that the engine encounters following the channels of the previous section. Section 5.4 will compare these results to a traditional adaptive controller.

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1 = AWGN	1 = BPSK 1/2 rate	6250000	1	0.057202
2	1 = AWGN	2 = BPSK 3/4 rate	9375000	22	0
3	1 = AWGN	4 = QPSK 3/4 rate	18750000	19	0
4	1 = AWGN	2 = BPSK 1/2 rate	9375000	9	0
5	2 = Flat fading	8 = 64-QAM 3/4 rate	56250000	25	0.000174
6	2 = Flat fading	4 = QPSK 3/4 rate	18750000	26	0
7	2 = Flat fading	5 = 16-QAM 1/2 rate	25000000	19	0.000434
8	3 = Dispersive fading	2 = BPSK 3/4 rate	9375000	22	0
9	3 = Dispersive fading	6 = 16-QAM 3/4	37500000	9	0.10041
10	3 = Dispersive fading	4 = QPSK 3/4 rate	18750000	26	0.13042
11	3 = Dispersive fading	4 = QPSK 3/4 rate	18750000	26	0.13042
12	3 = Dispersive fading	4 = QPSK 3/4 rate	18750000	26	0.13042
13	4 = Rician	4 = QPSK 3/4 rate	18750000	26	0
14	4 = Rician	8 = 64-QAM 3/4 rate	56250000	23	0
15	4 = Rician	6 = 16-QAM 3/4	37500000	9	0.006688
16	4 = Rician	4 = QPSK 3/4 rate	18750000	9	8.68E-05
17	1 = AWGN	2 = BPSK 3/4 rate	9375000	22	0
18	1 = AWGN	7 = 64-QAM 2/3 rate	50000000	6	0.018333
19	1 = AWGN	8 = 64-QAM 3/4 rate	56250000	19	0.000869
20	1 = AWGN	2 = BPSK 3/4 rate	9375000	22	0

Figure 5.11: CR toolset trace showing cognitive engine reacting to known channel

In trend step 17 shown in Figure 5.12, the engine encounters a channel that it has seen before, an AWGN channel. It decreased its data rate to 9 Mbps, changed its modulation and coding to radio mode 2 BPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) decreased to zero. Note that the CSM achieved this better error rate performance by instructing the WSGA to balance fitness functions equally, minimize BER, minimize power, and maximize data rate.

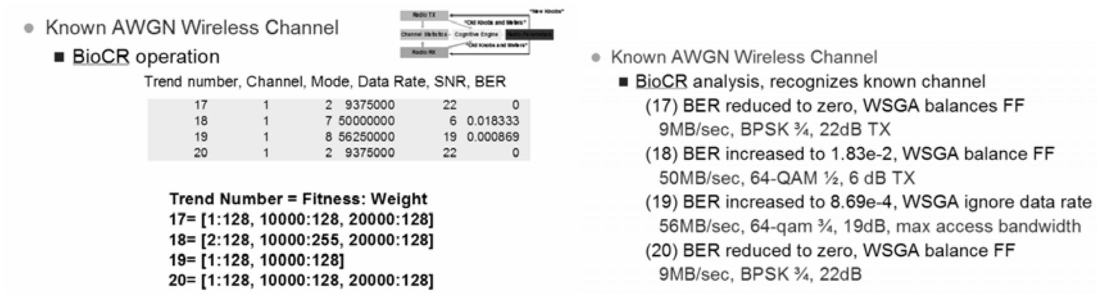


Figure 5.12: Summary of cognitive engine behavior in known AWGN channel

In trend step 18 shown in Figure 5.12, the engine increases its data rate to 50 Mbps, changed its modulation and coding to radio mode 7 64-QAM ½ rate, and decreased its power to 6 dB. The bit error rate (BER) increased to 1.83×10^{-2}. Note that the CSM instructed the WSGA to give highest priority to minimizing power, while equally weighting the need to minimize BER and maximize data rate.

In trend step 19 shown in Figure 5.12, the engine increases its data rate to 56 Mbps, changed its modulation and coding to radio mode 8 64-QAM ¾ rate, and increased its power to 19 dB. The bit error rate (BER) decreased to 8.69×10^{-4}. Note that the CSM instructed the WSGA equally weighting the need to minimize BER and minimize power, ignoring data rate. In this case since engine weighted the goals equally, it traded its goal of minimizing power for better BER performance and achieved better data rate in the process. Note that since the engine had already learned about the AWGN channel it was able to recognize that the observed wireless channel was an AWGN channel and quickly achieve higher data rates with decent BER much quicker than in trend steps one through four.

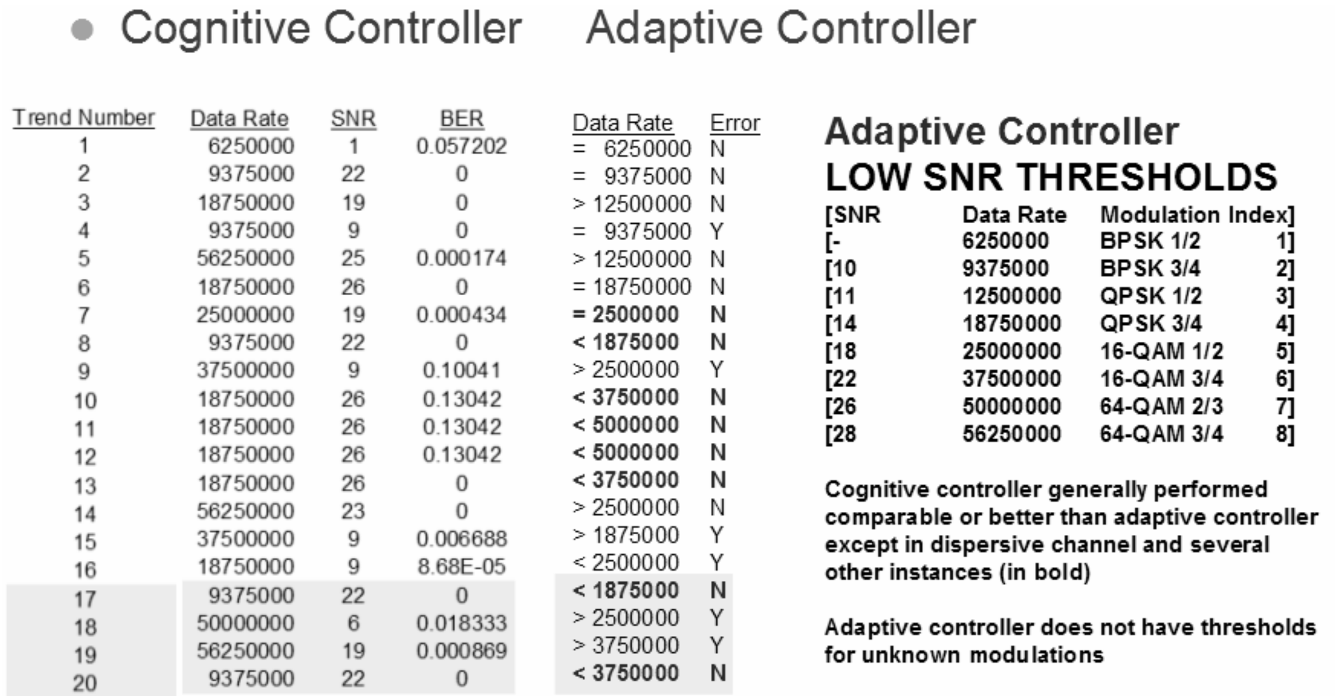
In trend step 20 shown in Figure 5.12, the engine decreases its data rate to 9 Mbps, changed its modulation and coding to radio mode 2 BPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) decreased to 0. Note that the CSM achieved this better error rate performance by instructing the WSGA to balance fitness functions equally, minimize BER, minimize power, and maximize data rate.

This section illustrated the cognitive engine's ability to recognize a known channel and quickly find a balance of radio parameters that meet the goals for that channel.

Through the experimental process I did discover some situations where the engine did not respond properly, exhibiting ghosting of answers, bouncing back and forth between solutions. This phenomenon was observed in trend steps seventeen through twenty. In these scenarios the engine was presented with data it had seen before but did not immediately settle on a solution due to sensitivity in the algorithm implementations. Future research could be pursued to further probe the limits of the engine's operation and investigate opportunities for improvement, including how best to configure the genetic algorithms and engine mathematics to avoid engine solution errors.

5.4 Comparison To Traditional Adaptive Controller

This section discusses how the cognitive engine compares to the traditional adaptive controller presented earlier in the chapter. Figure 5.13 provides a succinct description of how the two different mechanisms faired in the changing known and unknown channels and Figure 5.14 provides a visual summary of the results of this investigation. The part of the figure titled "Low SNR Thresholds" was repeated here for convenience.



● Cognitive Controller Adaptive Controller

Trend Number	Data Rate	SNR	BER	Data Rate	Error
1	6250000	1	0.057202	= 6250000	N
2	9375000	22	0	= 9375000	N
3	18750000	19	0	> 12500000	N
4	9375000	9	0	= 9375000	Y
5	56250000	25	0.000174	> 12500000	N
6	18750000	26	0	= 18750000	N
7	25000000	19	0.000434	**= 25000000**	**N**
8	9375000	22	0	**< 18750000**	**N**
9	37500000	9	0.10041	> 25000000	Y
10	18750000	26	0.13042	**< 37500000**	**N**
11	18750000	26	0.13042	**< 50000000**	**N**
12	18750000	26	0.13042	**< 50000000**	**N**
13	18750000	26	0	**< 37500000**	**N**
14	56250000	23	0	> 25000000	N
15	37500000	9	0.006688	> 18750000	Y
16	18750000	9	8.68E-05	< 25000000	Y
17	9375000	22	0	**< 18750000**	**N**
18	50000000	6	0.018333	> 25000000	Y
19	56250000	19	0.000869	> 37500000	Y
20	9375000	22	0	**< 37500000**	**N**

Adaptive Controller
LOW SNR THRESHOLDS

[SNR	Data Rate	Modulation	Index]
[-	6250000	BPSK 1/2	1]
[10	9375000	BPSK 3/4	2]
[11	12500000	QPSK 1/2	3]
[14	18750000	QPSK 3/4	4]
[18	25000000	16-QAM 1/2	5]
[22	37500000	16-QAM 3/4	6]
[26	50000000	64-QAM 2/3	7]
[28	56250000	64-QAM 3/4	8]

Cognitive controller generally performed comparable or better than adaptive controller except in dispersive channel and several other instances (in bold)

Adaptive controller does not have thresholds for unknown modulations

Figure 5.13: Comparison of adaptive controller behavior to cognitive engine behavior

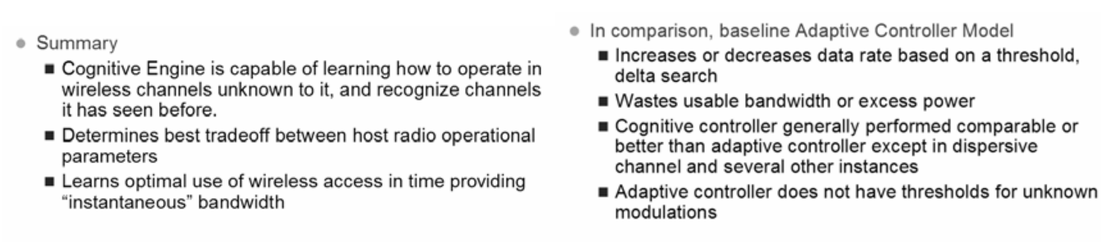


Figure 5.14: Summary of cognitive engine behavior in AWGN channel

Several columns in Figure 5.13 were created excerpting the information about the cognitive engine's data rate, SNR, and BER for each trend step. The data rate and error status achieved by the adaptive controller at each trend step are documented. A "N" in the error column means the adaptive controller observed no errors, while a "Y" meant that the adaptive controller did not observe enough SNR in the wireless channel for the modulation/data rate selected by the adaptive controller and was observing symbol errors. This occured when the channel quality suddenly decreased or increased and the adaptive controller either was not able to close the link and lost data communications or was wasting bandwidth by running the modulation index and data rate too low. The “<”, “=”, or “>” indicates whether the cognitive engine or adaptive controller achieved the highest data rate. “56250000 > 3750000” means that the engine performed better in trend step nineteen. As noted earlier in the chapter, since a software phase lock loop was not included in the simulation, the data from trend steps eight through twelve in the dispersive fading channel appeared to be inaccurate due to incorrect manual de-rotation of the phase and was ignored in this analysis. The cognitive engine had as good or better performance than the adaptive controller for eleven of the fifteen trend steps analyzed.

This analysis showed that the cognitive engine is capable of learning how to operate in wireless channels unknown to it and recognizing channels it has seen before. It determined the best tradeoff between host radio operational parameters and learns the optimal use of wireless access in time, providing “instantaneous” bandwidth. In comparison, the baseline adaptive controller model is only capable of increasing or decreasing data rate based on a threshold using a delta search method, wasting usable bandwidth or excess power. The cognitive controller generally performed comparable or better than the adaptive controller, except in dispersive channel and several other instances. While the cognitive engine as a mechanism to evolve in time and learn how to operate with unknown modulations, the adaptive controller is limited in that it does not have thresholds for unknown modulations and therefore may not be well suited for the dynamic environment where rapidly deployable communications systems are used which often requires interoperating between user communities that have unique radio configurations.

5.5 Summary

Chapter 5 that showed how the cognitive engine could evolve the radio’s operation in the face of unanticipated wireless channels, like those found in rapidly deployable emergency communications situations. The cognitive radio simulation toolset was presented and the concept of using trend steps was discussed for freezing simulation time and providing analysis of the cognitive learning process. Simulation results were presented that detailed the cognitive engine's behavior while operating in a set of channels that were both known and unknown to it. A traditional adaptive controller was introduced and used as a baseline for analyzing the behavior of a cognitive engine in changing electromagnetic environments. Limitations of the mechanism were observed and recommendations for future improvements were made. The following chapter demonstrates some functionality of the cognitive engine operating in a real world adaptive radio hardware test bed.

Chapter 6: Results from Virginia Tech CR Hardware Test Bed Experiment

This chapter discusses a test bed developed to explore the behavior of a cognitive engine in an actual disaster response communications system. The experimental tests reflected this environment, including a test of the cognitive engine on a hardware platform that was subject to intense interference and signal jamming. Such a caustic wireless channel could be due to the destruction and resulting malfunction of infrastructure that might occur in a natural disaster or attack on the homeland.

6.1 CR Engine Telemedicine Demonstration - Jamming Channel

Our group implemented a cognitive radio hardware test bed demonstration platform shown and presented the results during the Department of Homeland Security (DHS) SAFECOM session as a supplement to [5] at the 2004 International Symposium on Advanced Radio Technologies (ISART). The WSGA demonstration illustrated adaptive radio controller functionality given a set of simulated goals from the CSM, permitting the cognitive radio test bed to operate in the presence of an interferer without switching frequency. Various radio "knobs" including power, modulation, coding, and TDMA schemes were adjusted by the WSGA given the goals simulated from the CSM. Simulated CSM action was required for this hardware demonstration because the CSM application was not yet implemented. At the time we did the experiment, we were not able to test the CSM because the Proxim radios lacked the needed flexibility. Since this demonstration, the final cognitive engine code base was completed, including the CSM. The resulting cognitive radio engine alpha code release sets the stage for the cognitive engine to operate autonomously once the hardware platform is expanded to a fully programmable software radio.

6.2 WSGA Experiment for Maintaining QOS in the Presence of a Jammer

Figure 6.1 shows a network layout used in a winter 2004 test of the WSGA functionality in the cognitive radio engine. In this demonstration a single user on either end of a broadband wireless radio link established a video conference using Apple iChat AV to simulate a telemedicine operation over a cognitive radio link that was under attack.

Proxim Tsunami radios operating in the 5.8 GHz Unlicensed National Information Infrastructure (UNII) band were used as the host adaptive radio platform. PC's at the base station and subscriber units were used to execute the WSGA program on the base station side and accept the WSGA update parameters for both the base station and subscriber units. A third Tsunami base station was used as an interferer/jammer.

The radios were capable of changing the following knobs:

(1) Transmitter power
    (6 – 17 dBm in 1 dBm steps at the base station)
(2) Modulation
    (QPSK, QAM8, and QAM16)
(3) FEC
    (only adjustable for QPSK -> 1/2 or 3/4 rate;
    QAM8 always uses 2/3 rate and QAM16 always uses 3/4 rate)
(4) Uplink/downlink timeslot ratio
(5) Center frequency (5.740-5.806 GHz)

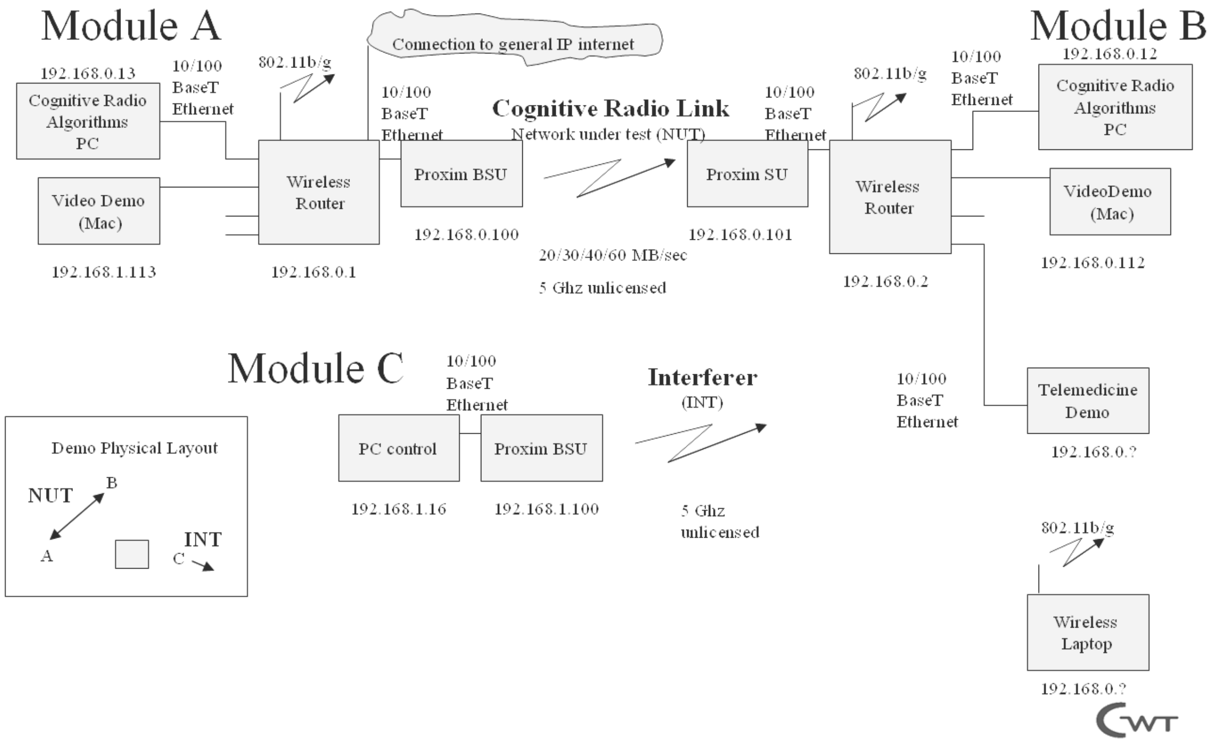


Figure 6.1: Winter 2004 cognitive engine test setup

To start the demo, the interfering radio was turned off, and the broadband video connection was established to show error-free video quality. The network under test (NUT) had the initial radio parameters: transmit power = 6 dBm, QAM 16, First Inbound Slot = 22 (of 24), 3/4 FEC, Frequency = 5740.40 MHz.

The interferer was then turned on with these parameters: transmit power = 11 dBm, QAM 16, First Inbound Slot = 12 (of 24), 3/4 FEC, Frequency = 5757.69 MHz.

With the interferer on, the video link quality substantially degraded. The WSGA was run with set goals and initial population members as a simulated CSM action. As part of the experiment the radios were prevented from switching frequency. This forced the engine to evolve the radio's operation while coexisting with the in band frequency jammer, a significant challenge. This scenario simulated a worst case situation in which a cognitive radio had to make the best of a bad wireless channel environment and still maintain the quality of service required to support an emergency telemedicine operation. Rather than switch frequencies when channel degradation occurs which many carrier sense technologies do, the engine learned the best combination of radio parameters to allow it to operate in the frequency band being jammed.

As the frequencies indicate, the NUT and interferer operated on slightly different center frequencies, but were co-channel interferers due to their overlapping 20.75 MHz bandwidths. If both radios were set to the same center frequency, the link quality degraded beyond the point that the radios could communicate at all, so a small offset in the frequency was used to allow communications that were severely degraded without completely interrupting the link.

The WSGA was sent goals to maximize the power and the coding gain, which implied reducing the data rate by lowering the modulation index and increasing the redundancy in the data encoding. When the WSGA finished, the NUT's radio parameters were: transmit power =16 dBm QPSK, First Inbound Slot 4 (of 8), 1/2 FEC, Frequency = 5740.40 MHz.

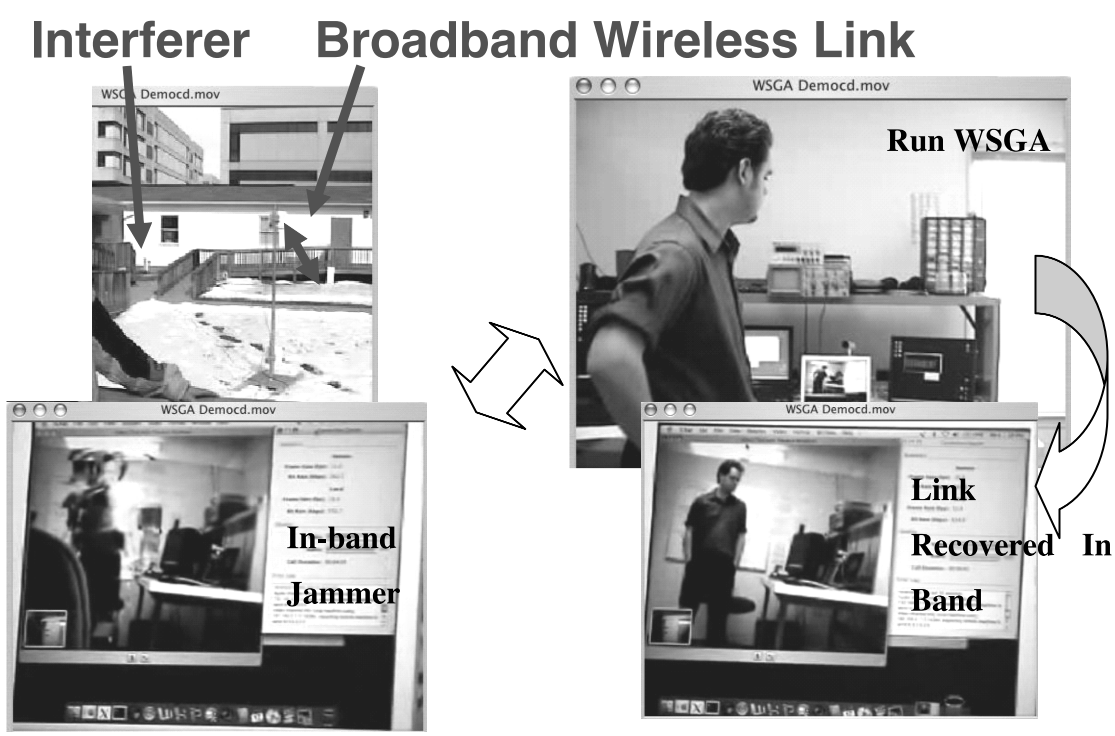


Figure 6.2: Photographs of cognitive engine control of adaptive radio network

With the new parameter settings, the video quality improved to that of a near flawless signal while the interferer was still on. The progression of the demonstration can be seen in Figure 6.2. The link quality was initially high, degrades significantly before the WSGA is run, and is then returns to high quality after the WSGA was run. This experiment was a quantitative evaluation of the WSGA because at the time no mechanism existed to collect data regarding the percent of packets lost.

6.3 Summary

This chapter discussed a test bed developed to explore the behavior of a cognitive engine in disaster response communications system. The experimental tests reflected this scenario, including a test of the cognitive engine on a hardware platform that was subject to intense interference and signal jamming. Such a caustic wireless channel could be due to the destruction and resulting malfunction of infrastructure that might occur in a natural disaster or attack on the homeland. In the experiment presented in chapter 6 the link quality was initially high until it was attacked by the jammer, at which point the video quality degraded significantly before the WSGA was run. Following reconfiguration of the radio by the WSGA with fixed goals from the simulated CSM the video link returns to high quality. This experiment served as a test of the WSGA's ability to reconfigure the hardware radio test bed platform. Future research may be pursued to test the CSM in the hardware test bed. The next chapter provides conclusions and recommendations for follow on research.

Chapter 7: Conclusions and Recommendations

This chapter provides a summary of the research I present in this dissertation, details my research contributions, and provides recommendations for future research directions.

7.1 Summary of Research Results

This research focused on developing a cognitive radio that could operate reliably in unforeseen communications environments like those faced by the disaster and emergency response communities. Cognitive radios may also offer the potential to open up secondary or complimentary spectrum markets, effectively easing the perceived spectrum crunch while providing new competitive wireless services to the consumer. A structure and process for embedding cognition in a radio was presented, including discussion of how the mechanism was derived from the human learning process and mapped to a mathematical formalism called the BioCR. Results from the implementation and testing of the model in a hardware test bed and simulation test bench were presented, with a focus on rapidly deployable disaster communications.

This dissertation presented a number of key results. The simulation in Chapter 5 showed that the cognitive engine finds the best tradeoff between a host radio's operational parameters in changing wireless conditions, while the baseline adaptive controller only increases or decreases its data rate based on a threshold, often wasting usable bandwidth or excess power when it is not needed due its inability to learn. The hardware test bed in Chapter 6 showed that the cognitive engine can learn how to configure the adaptive host radio to operate in the same band as a jammer. Future work needs to be done to investigate the sensitivity of the engine's algorithms, including how to address conditions when the solution does not settle, but bounces back and forth. This is a topic for future research, as well as investigating mechanisms for accelerating the engine's solution settle time. A fraction of the approaches explored in this research have been implemented due to time and resource constraints. Of particular interest will be how to leverage dynamic domain specific genetic tags and templates in a fully distributed system.

7.2 Summary of Contributions

This dissertation made a number of research contributions. I developed a biologically inspired model of cognition in a radio architecture. This contribution was important because as of the start of this research in late 2001 very little, if any, literature existed regarding how to build a "real world" cognitive radio. My research addressed that gap. I proposed that genetic algorithm operations could be used to realize the biologically inspired cognitive radio model. This contribution was important because when I began developing the mathematical formalism for the model in late 2002, most researchers were assuming an expert system as the central brain of the cognitive radio. My research set a stake in the sand and recommended extending mathematics to the cognitive radio research realm that could inherently evolve with changing times and needs. Specifically, this research proposes and details how the chaotic meta-knowledge search, optimization, and machine learning properties of distributed genetic algorithm operations could be used to map this model to a computable mathematical framework in conjunction with dynamic multi-stage distributed memories.

I developed an algorithmic framework to realize the cognition mechanism which was modeled after the functional operation of the brain, but not the physical design. This contribution was significant because most cognitive system designers immediately assume a neural network of nodes was needed to learn. The model I proposed shows that the system could be abstracted to contextual flows of information corresponding to the surrounding environment and current system status. The system formalism was contrasted with existing cognitive radio approaches, including traditionally brittle artificial intelligence approaches. The cognitive engine architecture and algorithmic framework is developed and introduced, including the Wireless Channel Genetic Algorithm (WCGA), Wireless System Genetic Algorithm (WSGA), and Cognitive System Monitor (CSM).

I developed a cognitive radio simulation toolset for evaluating the behavior the cognitive engine. This contribution was necessary to explore my research proposal. This work answered the question, could a cognitive engine using genetic algorithms identify and operate in both known and unknown wireless channels? The simulation indicated yes, however work was needed to extend the concept to a cognitive radio network. Finally, I used the toolset to analyze the cognitive engine's performance in different operational scenarios. This contribution provided the methodology for examining the behavior of the engine, including my development of the "trend step" which could freeze time and show what was going on under the hood of the cognitive engine at any given time. The toolset I developed facilitated this analysis through extensive time stamped data logging of the simulated adaptive radio and cognitive radio engine output.

7.3 Future Research and Recommendations

This research lays the foundation for a number of opportunities for future academic exploration. I recommend that future researchers implement a fully distributed CR engine model simulation. This research advance could be used to explore how to structure future field trials of the CR engine model within a network. These field trials could be pursued using real world channel measurements and a programmable wireless network platform, including an integrated channel sensing capability like the sounder when it becomes available. The simulation could then be transitioned to a cognitive wireless network test bed.

I think that it would interesting to extend this research and investigate waveform level CSM channel classification using the algorithm developed by Tim Gallagher. Gallagher's algorithm converts an impulse response to error statistics, skipping symbol level HMM channel modeling with error statistics. Completing this research would allow the cognitive engine to classify channels using statistical features other than symbol error behavior, potentially accelerating and refining the cognitive functions in the radio.

I recommend fully implementing the Goal Evolver CSM block with peer to peer communications. This advance would allow the cognitive wireless network to operate autonomously, which would test the scalability of the engine in a real world setting.

As part of the cognitive network research, I recommend developing a version of the cognitive engine for cross-layer co-simulation with the CRANIAsim OPNET project. The CRANIAsim OPNET project was an effort to model cognitive radios in a network, but lacked physical and MAC layer resolution. By "black-boxing" the cognitive engine and providing an "BioCR engine-to-OPNET" translation API which translates symbol level statistics to packet level statistics and makes relevant cognitive engine control data available to the researcher, networking researchers could interact with the cognitive engine and recommend MAC and network layer engine processing additions. This cross layer experiment could investigate appropriate interlayer communications in the cognitive engine.

Finally, I recommend extending the focus of this research from robust disaster communications and networking to secure military communications and networking. This research advance would allow researchers to extend the creative learning BioCR mechanism to a number of unique applications which could be used to safeguard our nation.

Bibliography

[1] C. W. Bostian, S. F. Midkiff, T. M. Gallagher, C. J. Rieser, and T. W. Rondeau, "Rapidly Deployable Broadband Communications for Disaster Response, " Proceedings of the International Symposium on Advanced Radio Technologies (ISART), invited paper in Department of Homeland Security (DHS) SAFECOM session, Boulder, CO, March 2-4, 2004, NTIA Special Publication SP-04- 409, pp. 87-92.

[2] J. Mitola. Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. New York: John Wiley and Sons, 2000.

[3] S. F. Midkiff and C. W. Bostian, "Rapidly Deployable Broadband Wireless Communications for Emergency Management", presented at the National Digital Government Research Conference (dg.o 2001), May 21-23, 2001, Redondo Beach, CA.

[4] J. Mitola and Z. Zvonar. Software Radio Technologies: Selected Readings. New York: the Institute of Electrical and Electronics Engineers, Inc., 2001.

[5] J. Mitola and G. Maguire, Jr., "Cognitive Radio: Making Software Radios More Personal," IEEE Personal Communications Magazine, Vol. 6, No. 6, pp. 13-18, August 1999.

[6] J. Mitola III, "Cognitive Radio," Licentiate proposal, KTH, Stockholm, Sweden, Dec 1998.

[7] DARPA XG program: http://www.darpa.mil/ato/programs/xg/

[8] FCC Cognitive Radio website: http://ftp.fcc.gov/oet/cognitiveradio/

[9] NSF Research in Networking Technology and Systems (NetS) program 04-540: http://www.nsf.gov/pubs/2004/nsf04540/nsf04540.htm

[10] J. Mitola KTH research website: http://www.it.kth.se/~jmitola/

[11] J. Mitola doctoral dissertation, KTH, 2000: http://www.it.kth.se/~jmitola/cognitiveRadio.ppt

[12] J. Mitola dissertation defense, KTH, 2000: http://www.it.kth.se/~jmitola/Mitola_Dissertation8_Integrated.pdf

[13] DARPA XG RFCS: http://www.darpa.mil/ato/programs/xg/rfcs.htm

[14] FCC Spectrum Policy Task Force (SPTF) website: http://www.fcc.gov/sptf/

[15] FCC SPTF report: http://www.fcc.gov/sptf/reports.html

[16] FCC Cognitive Radio NPRM 03-108: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-03-322A1.doc

[17] FCC Cognitive Radio Workshop Video: http://www.fcc.gov/realaudio/mt051903.ram

[18] C. W. Bostian, S. F. Midkiff, T. M. Gallagher, C. J. Rieser, and T. W. Rondeau, "Rapidly Deployable Broadband Communications for Disaster Response, " Proceedings of the International Symposium on Advanced Radio Technologies (ISART), invited paper in Department of Homeland Security (DHS) SAFECOM session, Boulder, CO, March 2-4, 2004, NTIA Special Publication SP-04- 409, pp. 87-92.

[19] C. J. Rieser, T. W. Rondeau, C. W. Bostian, and T. M. Gallagher. "Cognitive Radio Test bed: Further Details and Testing of a Distributed Genetic Algorithm Based Cognitive Engine For Programmable Radios." IEEE MILCOM, to appear October 2004.

[20] Army Joint Tactical Radio System (JTRS) website: http://jtrs.army.mil/

[21] DARPA XG Solicitation: http://www.eps.gov/spg/USAF/AFMC/AFRLRRS/Reference-Number-PRDA-02-01-IFKPA/Modification%2003.html

[22] J. Mitola software radio website: http://ourworld.compuserve.com/homepages/jmitola/

[23] J. Mitola software radio definition: http://ourworld.compuserve.com/homepages/jmitola/whatisas.htm

[24] J. Mitola software radio course website: http://ourworld.compuserve.com/homepages/jmitola/swr97.htm

[25] G. Luger. Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 4th edition. Addison-Wesley / Pearson Education Limited, 2002.

[26] M. Negnevitsky. Artifical Intelligence: A Guide to Intelligent Systems. New York: Addison-Wesley / Pearson Education Limited, 2002.

[27] T. Mitchell. Machine Learning. New York: McGraw-Hill Series in Computer Science, 1997.

[28] C. J. Rieser. "Design and Implementation of Sampling Swept Time Delay Short Pulse (SSTDSP) Channel Sounder for LMDS." M.S. Thesis Virginia Tech, Blacksburg, Virginia, July 2001.

[29] C. L. Dillard. "A Study of Rough Surface Scattering Phenomena in the LMDS Band (28 GHz)" M.S.E.E Thesis Virginia Tech, Blacksburg, Virginia, February 2003.

[30] M. Miniuk. “Channel Impulse Response and Its Relationship to Bit Error Rate at 28 GHz” M.S.E.E Thesis Virginia Tech, Blacksburg, Virginia, December 2003.

[31] T. M. Gallagher. “Characterization and Evaluation of Non-Line-of-Sight Paths for Fixed Broadband Wireless Communications.” Ph.D. Dissertation Virginia Tech, Blacksburg, Virginia, June 2004.

[32] E. N. Gilbert, “Capcity of a burst-noise channel,” Bell Syst. Tech. J., Vol. 39, pp. 1253-1266, Sept 1960.

[33] W. Turin and M. M. Sondhi, “Modeling Error Sources in digital channels,” Selected Areas in Communications, IEEE Journal on, Vol. 11, Issue 3, pp. 340-347, April 1993.

[34] B. D. Fritchman, “A binary channel characterization using partitioned Markov chains.” IEEE Trans. Inform. Theory, Vol. IT-13, pp. 221-227, April 1967.

[35] S. Sivaprakasam, and K.S. Shanmugan, “An Equivalent Markov Model for Burst Errors in Digital Channels”, IEEE Trans. On Communications, vol. 43, pp. 1347-1355, Feb. 1995.

[36] A. Beverly and K.S. Shanmugan, “Hidden Markov Models for Burst Errors in GSM and DECT Channels”, International Conference on Communications (ICC), pp. 3692-3698, 1998.

[37] L.R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proceedings of the IEEE, vol. 77, pp. 257 – 286, Feb. 1989.

[38] W. Turin, “Simulation of error sources in digital channels,” IEEE J. Selected Areas Commun., Vol. 6, no. 1, pp. 85-93, Jan. 1988.

[39] A. Umbert and P. Diaz, “ A radio channel emulator for WCDMA, based on hidden Markov model (HMM),” IEEE 52nd Vehicular Technology Conference (VTS-Fall VTC 2000), Vol. 5, pp. 2173-2179, Sept 2000.

[40] A. Umbert and P. Diaz, “A generic radio channel emulator to evaluate higher layer protocols in a CDMA system,” Proc. on 11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2000), pp. 401-405, Sept. 2000.

[41] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” Bulletin of Mathematical Biophysics, Vol. 5, pp. 115-137, 1943.

[42] G. M. Shepherd and C. Koch, “Introduction to synaptic circuits,” The Synaptic Organization of the Brain, G. M. Shepherd, ed., New York: Oxford University Press, pp. 3-31, 1990.

[43] M. Caudill, “Neural network training tips and techniques,” AI Expert, pp. 56-61, January 1991.

[44] R. A. Jacobs, “Increased rates of convergence through learning rate adaptation,” Neural Networks, Vol. 1, pp. 295-307, 1988.

[45] D. F. Stubbs, “Six ways to improve back propagation results,” Journal of Neural Network Computing, pp. 64-67, Spring 1990.

[46] D. Stork, “Is backpropagation biologically plausible?,” Proc. of the International Joint Conference on Neural Networks, Washington DC, Vol. 2, pp. 241-246, 1989.

[47] S. J. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall, Englewood Cliffs, 1995.

[48] L. E. Berk. Development through the lifespan. Boston: Allyn & Bacon, 1998.

[49] J. Piaget. Biology and knowledge. Chicago: University of Chicago Press, 1971.

[50] M. Csikszentmihalyi. Creativity : flow and the psychology of discovery and invention, 1st ed. New York : Harper Collins Publishers, 1996.

[51] L. S. Vygotsky. Mind in society: The development of higher psychological processes. Cambridge, Massachusetts: Harvard University Press (Original works published 1930, 1933, 1935), 1978.

[52] H. Beilin. "Piaget's enduring contribution to development psychology," Developmental Psychology, Vol. 28, pp. 191-204, 1992.

[53] D. Kuhn, Cognitive Development, in Developmental Psychology: An advanced textbook, 3rd edition, M. H. Bornstein and M. E. Lamb (Eds.), pp. 211-272. New Jersey: Erlbaum, 1992.

[54] J. F. Nash, "Equilibrium Points in N-Person Games," Proc. of the National Academy of Sciences of the United States of America, Vol. 36, pp. 48-49, 1950.

[55] J. F. Nash, "The Bargaining Problem," Econometrica, Vol. 18, pp. 155-162, 1950.

[56] J. F. Nash, "Non-Cooperative Games," Annals of Mathematics, Vol. 54, pp. 286-295, 1951.

[57] D. Klahr and J. Wallace. Cognitive Development: An information-processing view. Hillsdale, NJ: Erlbaum, 1976.

[58] C.M. Kennedy, "A Conceptual Foundation for Autonomous Learning in Unforeseen Situations," Proc. ISIC/CIRA/ISAS Joint Conference, pp 483 – 488, 1998,.

[59] "The Process of Conceptual Change", Handbook of Child Psychology, Fifth Edition, Vol. 2, Cognition, Perception, and Language, pp. 774-779, 1998.

[60] G. Hinton et. al. "The 'wake-sleep' algorithm for unsupervised neural networks," Science, Issue 268, pp. 1158-1161, 1995.

[61] F. Sun and G. Hu, "Speech Recognition Based on Genetic Algorithm for Training HMM," Electronics Letters, vol. 34, pp. 1563 – 1564, Aug. 1998.

[62] C.W. Chau, S. Kwong, C.K. Diu, and W.R. Fahrner, "Optimization of HMM by a Genetic Algorithm," Proc. ICASSP-97, vol. 3, pp. 1727 – 1730, 1997.

[63] S. Kwong and C. W. Chau, "Analysis of Parallel Genetic Algorithms on HMM Based Speech Recognition System," IEEE Trans. Consumer Electronics, vol. 43, pp. 1229 – 1233, Nov. 1997.

[64] E. Kee, S. Airey, and W. Cyre, "An Adaptive Genetic Algorithm," Proc. GECCO-2001, 2001, pp. 391 – 397.

[65] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Pub Co., 1989.

[66] L. Kallel, B. Naudts, and A. Rogers (Eds.). Theoretical Aspects of Evolutionary Computing. New York: Springer, 2001.

[67] T. W. Rondeau, C. J. Rieser, T. M. Gallagher, C. W. Bostian, "Online Modeling of Wireless Channels with Hidden Markov Models and Channel Impulse Responses for Cognitive Radios," IEEE Proc. IMS 2004, June 2004.

[68] Proxim Tsunami datasheet: http://www.proxim.com/products/bwa/multipoint/tsunami/multipoint/index.html

[69] N. Nefedov, "Discrete Channel Models for Wireless Data Services," IEEE VTC, pp.683-687, 1998.

[70] H. Bai and M. Atiquzzaman. "Error Modeling Schemes for Fading Channels in Wireless Communications: A Survey," IEEE Communications Society Surveys and Tutorials, The Electronic Magazine of Original Peer Reviewed Survey Articles, 4th Quarter 2003, Vol.5 No.2, http://www.comsoc.org/livepubs/surveys/public/2003/oct/bai.html.

[71] M. K. Simon and M. Alouini, "A unified approach to the performance analysis of digital communication over generalized fading channels," Proceedings of the IEEE, Vol. 86, Issue. 9, pp. 1860 – 1877, Sept. 1998.

[72] R. O'Donnell, "Prolog To A Unified Approach To The Performance Analysis Of Digital Communication Over Generalized Fading Channels," Proceedings of the IEEE, Vol. 86, Issue. 9, pp. 1858 – 1859, Sept. 1998.

[73] MATLAB-Simulink Adaptive Wireless Controller Model, Martin Clark: http://www.mathworks.com/matlabcentral/files/3540/IEEE80211a.jpg

[74] MATLAB-Simulink HyperLAN/2 Model, Chris Thorpe: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=766&objectType=file

[75] MATLAB-Simulink IEEE 802.11a WLAN Model, Martin Clark: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3540&objectType=file

[76] W. Tranter et al. Principles of communication systems simulation with wireless applications. Upper Saddle River, NJ : Prentice Hall, 2004.

[77] Rieser BioCR cognitive radio toolset demonstration webpage:
http://www.ee.vt.edu/~cjrieser/biocrtoolset.html

Appendix A: Glossary

1. **Accommodation:** To change one's understanding to include a new concept
2. **Adaptive controller:** Mechanism for changing the radio data rate based on changing SNR
3. **Adaptive radio:** A radio that may switch between preprogrammed radio profiles
4. **Adaptation:** Adapting to the world through assimilation and accommodation
5. **Assimilation:** To include directly into one's understanding a new concept
6. **AWGN:** Additive White Gaussian Noise, also called white noise due to its spectral flatness
7. **BER/SER:** Bit or symbol error rate, a measure of the number of errors in a channel
8. **Biologically inspired:** Derived from a biological system
9. **BioCR engine:** Shorthand for biologically inspired cognitive radio engine
10. **Channel:** Shorthand for wireless channel, or communications medium
11. **Classifier:** A mathematical mechanism that can categorize inputs based on a database
12. **Cognition:** The action or faculty of knowing taken in its widest sense, including sensation, perception, conception, etc., as distinguished from feeling and volition
13. **Cognitive:** Pertaining to cognition, or to the action or process of knowing
14. **Cognitive Radio:** A radio that can learn how to operate in unanticipated channels
15. **CSM:** Cognitive System Monitor, learns how to synthesize channel information to develop operational goals for WSGA radio evolver
16. **Dispersive fading wireless channel:** Transmitted energy arrives at the receiver at different times, superimposed on other symbols
17. **Distributed algorithms:** Mathematical processes that may be located in different logical or physical memory spaces
18. **Distributed model:** System design that may be used by many host platforms at once
19. **Evolution:** To change in time
20. **Fixed radio:** A radio which has its parameters set at the time of manufacture

21. **Flat fading wireless channel:** Frequency components of a received radio signal vary in the same proportion simultaneously
22. **Genetic algorithm:** GA, An algorithm based on biological mechanisms of evolution
23. **Goal evolver:** Mechanism in BioCR engine used to evolve LTM content through STM workspace
24. **Learning classifier:** A mathematical mechanism that is capable of learning how to categorize an input that is not in a classifier database
25. **Learning optimizer:** A mathematical mechanism that is capable of learning how to optimize a system
26. **LTM:** Long term memory, distributed memory space that contains channel statistics, WSGA goals, and other engine control data
27. **Meta-GA functions:** GAs that are able to learn to control and monitor other GAs
28. **Modulation:** Method to transmit a signal using properties of electromagnetic, optical, or sound waves including amplitude, frequency, phase, spatial orientation, or code
29. **Neural network:** A mathematical model of the biological brain approximating neural interconnection, communication, and processing functions.
30. **Optimizer:** A mathematical mechanism that can be used to change system configuration to provide optimal performance for a given set of constraints in time
31. **PER:** Packet error rate, a measure of the number of packet errors in a channel
32. **Power level:** Mean power of a radio transmitter
33. **Programmable radios:** Radios that may be changed to add or remove capabilities
34. **Radios:** Telecommunication by modulation and radiation of electromagnetic waves
35. **Radio profile:** The collection of radio parameters that define the radios operation
36. **Rayleigh fading:** In electromagnetic wave propagation, phase-interference fading caused by multipath, and which may be approximated by the Rayleigh distribution

37. **Rician:** Rayleigh fading with a strong line of sight content is said to have a Rician distribution, or to be Rician fading
38. **Robust:** Highly reliably
39. **Scaffolding learning:** To increase understanding from one stage to another
40. **SNR:** Signal to noise ratio
41. **STM:** Short term memory, workspace used by the BioCR engine to generate goals for the WSGA
42. **Symbol:** Mapping of values to modulation characteristics like amplitude, frequency, or phase
43. **WCGA:** Wireless Channel Genetic Algorithm, quantifies and models channel
44. **White noise:** Noise having a frequency spectrum that is continuous and uniform over a specified frequency band. Has equal power per hertz over the specified frequency band
45. **WSGA:** Wireless System Genetic Algorithm, evolves radio based on CSM goals

Appendix B: Cognitive Radio Engine Patent Application - VTIP 03.056

In June 2004 Virginia Tech Intellectual Properties (VTIP) submitted a patent application titled "Cognitive Radio Engine Based On Genetic Algorithms in A Network" covering the cognitive radio engine model presented in this dissertation, including the proof of concept software realization of that model in a cognitive engine that is capable of controlling both a simulated adaptive radio host and an agile hardware radio host.

The interested reader may contact VTIP to request further information about the patent application and licensing the cognitive radio engine (VTIP # 03.056). The patent application was about 70 pages. Table B.1 summarizes the original disclosure that was submitted to VTIP.

Table B.1: VTIP Disclosure No. 03-056
Title: Biologically Inspired Cognitive Wireless L12 Technology: Genetic Algorithms Applied to Cognitive Radio
Inventor: Christian J Rieser, Tom Rondeau, Charles W Bostian, Walling Cyre, and Tim Gallagher
Description: Wireless communications systems (radios) can be described as fixed, adaptive, or cognitive. The technical characteristics of fixed radios are set at the time of manufacture. An adaptive radio can respond to channel conditions that represent one of a finite set of anticipated events. Adaptive radios use artificial intelligence (AI) algorithms that are basically a series of "if, then, else" algorithms. A cognitive radio can respond intelligently to an unanticipated event - i.e., a channel that it has never encountered before. Our disclosure describes a novel and computationally efficient method to realize a truly cognitive radio based n genetic algorithms. An immediate market for this technology is in military and disaster communications, where radio systems must work under changing and unanticipated circumstances and in the presence of hostile jammers and interferers. The long-term market is in civilian radio communications systems like cellular telephones where spectrum and battery power are at a premium and in which the radio sets must continuously adapt to conserve these resources.
Patent Status: Patent Application Filed
Licensing Status:

Appendix C: CR Test bench Simulation Blocks and Source Code

This appendix presents documentation of the cognitive radio simulation test bench that I created using MATLAB-Simulink, including screenshots of the research process, code and program output, and file name listing with references. An explanation of each section of the simulation code base is provided. Full code for the cognitive radio toolset may be requested by contacting Virginia Tech Intellectual Properties, Inc.

C.1 Co-Simulation of Adaptive Radio Simulink Model and C++ Cognitive Engine

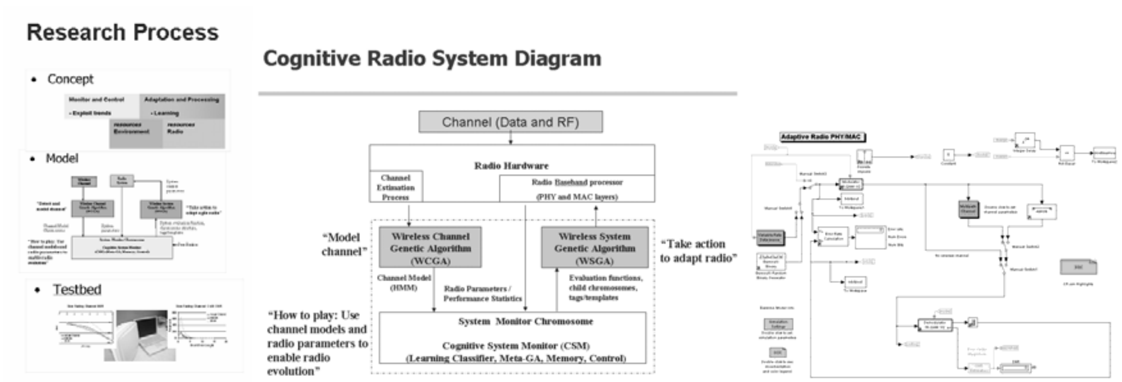


Figure C.1: Research process, cognitive radio (CR) system, and early CR test bench

C.2 Cognitive Engine Code and Program Output

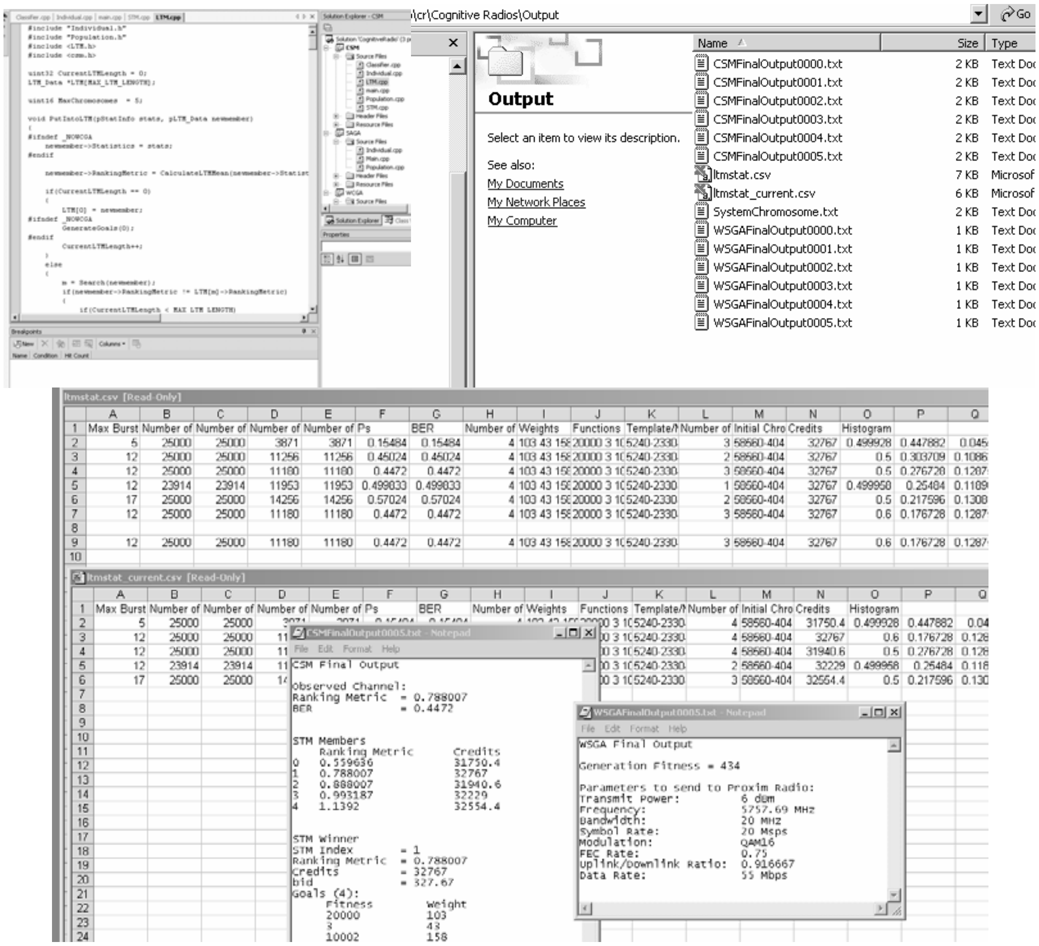


Figure C.2: Early cognitive engine code and output

C.3 Reference List of Experimental Code File Names

NOTE: Requests for full text of experimental code should be directed to VTIP (http://www.vtip.org), Reference VTIP#: 03.056

BioCR Toolset Code – Adaptive Radio Host Simulation Implementation

MATLAB-Simulink 6.5.1

By Christian Rieser of Virginia Tech

Summer 2004

Adaptive Radio model implementation based on freely available Modified Wi-Fi and HiperLAN2 models

with modules derived from MATLAB Central models:

IEEE 802.11a WLAN PHY by Martin Clark of The MathWorks

and

HIPERLAN/2 by Chris Thorpe of The MathWorks

Sample list of editable text files

CRModel/

- crsim12.mdl
- Rchanneldist.csv
- Nchanneldist.csv
- ltmerrordist.csv
- CRtest benchBERanalysisplot.csv
- crwcgainputsave.m
- crtrenddump.m
- crtoolset.m
- crtest benchdemo.m
- crtest benchdemo3.m
- crtest benchdemo2.m
- crtest bench.m
- crsimfadingmodenamelist.m
- crsimchannelangles.m
- crsimanalysis5.m
- crresetltmstat.m
- crresetltm.m
- crresetknobs.m

crcsmgoalevolver.m

crconfiguresim5.m

crbertest bench.m

crberresetknobs.m

crbermodesave.m

crberenginestatus.m

crberdatasave.m

crberanalysis.m

crber.m

cr3.m

cr2.m

cr1.m

c15_seglength.m

c15_intervals1.m

IEEE80211a_graphics.fig

IEEE80211a_init.mat

IEEE80211a_sfun.dll

Crberdata/

wsgaref.txt

***

BioCR Toolset Code - 80211 Reference

MATLAB-Simulink 6.5.1

By Martin Clark and Chris Thorpe of The MathWorks

Reference modules from MATLAB Central:

IEEE 802.11a WLAN PHY by Martin Clark of The MathWorks

and

HIPERLAN/2 by Chris Thorpe of The MathWorks

Sample list of editable text files

IEEE80211a and HIPERLAN/

    IEEE80211a_lib.mdl
    IEEE80211a.mdl
    IEEE80211a_udg.m
    IEEE80211a_settings.m
    IEEE80211a_open_graphics.m
    IEEE80211a_graphics.m
    hiperlan2.mdl

****

<u>BioCR Toolset Code – Cognitive Radio Engine Implementation</u>

C/C++ Microsoft Visual Studio 6.0

By Tom Rondeau of Virginia Tech, based on model/framework/algorithms created by Christian Rieser and genetic algorithm (GA) base code from Dr. Walling Cyre (WCGA code was written by Christian Rieser and Tom Rondeau as a class project in Dr. Cyre's GA class)
Summer 2004

Sample list of editable text files

Cognitive Radios/

    Channel Data/
        errorchannel_init0000.seq
    Executables/

parameters_WCGA.txt

parameters_CSM.txt

parameters_WSGA.txt

Libraries/

Classifier.h

CRMathInterp.h

csm.h

ExtendedMath.h

HMM.h

HMMSeqGen.h

HMMSeqStatGen.h

HMMStatGen.h

HMMStatGenFile.h

LTM.h

ProximAPI.h

RadioData.h

SimulinkAPI.h

STM.h

TCPIP.h

WSGA.h

WSGAFitFunc.h

Output/

ltmstat.csv

ltmstatReset.csv

SystemChromosome.txt

SystemKnobs.csv

SystemKnobsBERanalysis.csv

SystemKnobsReset.csv

SystemMeters.csv

wcgainput.csv

WSGAActions.csv

WSGAFinalOutput.csv

CRCode/

CognitiveRadio/

CognitiveRadio.dsp

CognitiveRadio.dsw

CSM/

Classifier.cpp

CSM.dep

CSM.dsp

CSM.dsw

CSM.mak

Individual.cpp

Individual.h

LTM.cpp

ltmstat_current.csv

main.cpp

Population.cpp

Population.h

STM.cpp

WCGAFinalOutput0000.txt

WSGAActions.csv

WCGA/

Definitions.h

Include.h

Individual.cpp

Individual.h

Main.cpp

Population.cpp

Population.h

WCGA.dep

WCGA.dsp

WCGA.dsw

WCGA.mak

wcgainput.csv

WSGA/

Definitions.h

Individual.cpp

Individual.h

Main.cpp

Population.cpp

Population.h

Setup Environment for WSGA.doc

WSGA.dep

WSGA.dsp

WSGA.dsw

WSGA.mak

WSGAOutput.txt

ExtendedMath/

BER.cpp

ExtendedMath.cpp

ExtendedMath.dsp

ExtendedMath.dsw

ReadMe.txt

StdAfx.h

Vector.cpp

HMM/

HMM.cpp

HMM.dep

HMM.dsp

HMM.dsw

HMM.mak

ReadMe.txt

StdAfx.cpp

StdAfx.h

HMMSeqGen/

HMMSeqGen.cpp

HMMSeqGen.dep

HMMSeqGen.dsp

HMMSeqGen.dsw

HMMSeqGen.mak

ReadMe.txt

StdAfx.cpp

StdAfx.h

HMMSeqGenFile/

HMMSeqGen.cpp

HMMSeqGen.dep

HMMSeqGen.dsp

HMMSeqGen.dsw

HMMSeqGen.mak

ReadMe.txt

StdAfx.cpp

StdAfx.h

HMMSeqStatGen/

HMMSeqStatGen.cpp

HMMSeqStatGen.dep

HMMSeqStatGen.dsp

HMMSeqStatGen.dsw

HMMSeqStatGen.mak

HMMSeqStatGenTemp.cpp

ReadMe.txt

StdAfx.cpp

StdAfx.h

HMMStatGen/

HMMStatGen.cpp

HMMStatGen.dep

HMMStatGen.dsp

HMMStatGen.dsw

HMMStatGen.mak

HMMStatGen-back01.cpp

ReadMe.txt

StdAfx.cpp

StdAfx.h

HMMStatGenFile/

HMMStatGenFile.cpp

HMMStatGenFile.dep

HMMStatGenFile.dsp

HMMStatGenFile.dsw

HMMStatGenFile.mak

ReadMe.txt

StdAfx.cpp

StdAfx.h

ProximAPI/

BSU_SUCmds.cpp

BSUCmds.cpp

ProximAPI.cpp

ProximAPI.dep

ProximAPI.dsp

ProximAPI.dsw

ProximAPI.mak

Readme.txt

StdAfx.cpp

StdAfx.h

SUCmds.cpp

WSGA.h

SimulinkAPI/

ReadMe.txt

StdAfx.cpp

StdAfx.h

SimulinkAPI.cpp

SimulinkAPI.dsp

SimulinkAPI.dsw

TCPIP/

ReadMe.txt

StdAfx.cpp

StdAfx.h

TCPIP.cpp

TCPIP.dep

TCPIP.dsp

TCPIP.dsw

TCPIP.mak

WSGAFitFunc/

ReadMe.txt

StdAfx.cpp

StdAfx.h

WSGAFitFunc.cpp

WSGAFitFunc.dep

WSGAFitFunc.dsp

WSGAFitFunc.dsw

WSGAFitFunc.mak

C.4 Detail of the Adaptive Radio MATLAB-Simulink Co-simulation

File: crsim12.mdl

Adaptive Radio Host Simulink Model

By Christian Rieser of Virginia Tech
Summer 2004

with modules from MATLAB Central model:
IEEE 802.11a WLAN PHY by Martin Clark of The MathWorks
and
HIPERLAN/2 by Chris Thorpe of The MathWorks

SUMMARY OF MODEL
* End-to-end 802.11a physical layer
* All mandatory and optional data rates: 6, 9, 12, 18, 24, 36, 48, and 54 Mb/s
* BPSK, QPSK, 16-QAM, 64-QAM modulations
* Forward error correction coding (convolutional; code rates 1/2, 2/3, 3/4)
* Viterbi decoding
* Data interleaving
* Data rates selectable on-the-fly
* Adaptive modulation demo over changing wireless channel
+THE ADAPTIVE RADIO BLOCK HAS BEEN REPLACED BY THE COGNITIVE ENGINE I/O+++

SUMMARY OF DISABLED FUNCTIONS
* OFDM transmission: 52 subcarriers, 4 pilots, 64-pt FFTs, circular prefix
* PLCP preamble (modeled as 2x2 long training sequences; see below)
* Receiver equalization

MODEL SIMPLIFICATIONS/ASSUMPTIONS
* Baseband-equivalent model (no up/down RF conversion)
* Random data transmision (no data scrambling used)
* Fixed number of data symbols per packet (no pad bits used)
* Continuous frame-to-frame operation (no coder state resetting via tail bits)
* Fixed transmit power level; link-SNR specified (on-the-fly)

* Idealized timing/frequency acquisition
* Not modeled:
  - MAC/PHY interface and PLCP header (TXVECTOR/RXVECTOR)
  - Short training sequences (for AGC, diversity, timing/frequency acq.)
  - Time windowing of OFDM symbols

C.5 Detail of the Cognitive Engine Model C++ Co-simulation

File: CSM.exe

Windows Visual C++ 6.0 Co-simulation Cognitive Engine formalism, framework, design, and implementation
By Christian Rieser and Tom Rondeau, Virginia Tech, Blacksburg, VA
Summer 2004

Controls Adaptive/Agile Radio simulation

(a) WCGA (Wireless Channel Genetic Algorithm)
- models channel and performance statistics based on channel measurements (sounder or simulation)
(b) WSGA (Wireless System Genetic Algorithm)
- evolves radio behavior for optimal response to unknown and known channels (changes radio knobs based on CSM goals)
(c) CSM (Cognitive System Monitor)
- uses knowledge of WCGA channel statistics and WSGA radio performance to synthesize goals for WSGA
- fully distributed learning optimizer/classifier and knowledge base

C.6 Detail of the CR Simulation Test Bench Co-simulation

File: crtoolset.m

Tests cognitive engine performance control of a simulated adaptive radio link in changing and unanticipated wireless channel conditions

By Christian Rieser, Virginia Tech, Blacksburg, VA
Summer 2004

NOTE: c15_seglength.m code from Dr. William Tranter's CAD for Communications book draft, Fall 2002, Virginia Tech, Blacksburg, VA, btranter@vt.edu

Produces a two-row matrix of error intervals and error-free intervals. Row 1 specifies the interval length and row 2 specifies the interval class (error(1) or no error(0)).

NOTE: c15_intervals.m code from Dr. William Tranter's CAD for Communications book draft, Fall 2002, Virginia Tech, Blacksburg, VA, btranter@vt.edu

Calculates and plots error intervals from a run-length error vector.

For an excellent text on simulation of communications systems, please see Dr. William Tranter's book "W. Tranter et al. Principles of communication systems simulation with wireless applications. Upper Saddle River, NJ: Prentice Hall, 2004." I would like to extend my thanks to Dr. Tranter for offering me the opportunity to read and provide feedback on his book before it was available at the bookstore. Dr. Tranter is both an outstanding author and an excellent teacher – my many thanks to him for taking time to chat with me over the years about communications and wireless system modeling. His MATLAB expertise and methodologies encouraged me to use that tool to architect and generate the final experimental research results I present in this dissertation.

Appendix D: BioCR Toolset Simulation Run Data Capture Logs

This appendix includes detailed BioCR toolset simulation run data capture logs from both the host radio and cognitive engine. An explanation of each trend step of the simulation run is provided.

CR Engine versus Traditional Adaptive Radio Controller – Unknown Channel

This section details how the cognitive engine responds to unknown channels. Figure D.1 shows a trace of the cognitive engine operating in four different channels. In this case all four channels were unknown to the engine when it began its operation.

(1) Additive White Gaussian Noise (AWGN) wireless channel
(2) Flat Fading Rayleigh wireless channel
(3) Dispersive Fading Rayleigh wireless channel
(4) Rician wireless channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05

Figure D.1: CR toolset trace showing cognitive engine reacting to unknown channel

The engine operated in an AWGN channel the first four steps of the demonstration (called "trend steps"), then encountered a flat fading channel from trend step five to seven. On trend step eight the engine encountered a dispersive fading channel. The engine operated in a Rician channel the remaining four trend steps.

The meanings of the WSGA fitness function codes are listed in Table D.1. WSGA fitness functions weights may range from 0 to 255.

Table D.1: WSGA Fitness Functions Used in Simulation

1 =	Minimize AWGN BER
2 =	Minimize Rayleigh BER
3 =	Minimize Rician BER
10000 =	Minimize Power Consumption
20000 =	Maximize Data Rate

The following section provides step by step analysis of the cognitive engine's operation in various unknown channels.

D.1 Trend Step 1 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202

Figure D.2: Trend step 1 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 1.05236
BER = 0.057202

LTM Members

	Ranking Metric	Credits
0	0	32784.2
1	1.05236	32767

STM Members

	Ranking Metric	Credits
0	0	32784.2
1	1.05236	32767

STM Winner
STM Index = 1
Ranking Metric = 1.05236
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
10000	128

Figure D.3: Trend step 1 engine data

In trend step 1 shown in Figures D.2 and D.3 the engine initialized its operation at data rate 6 Megabits per second (Mbps), radio mode 1, BPSK ½ rate forward error correction (FEC) code with a transmit power of 1 dBm, where dBm is decibel referenced to 1 milliwatt; 0 dBm equals one milliwatt. Note that to simplify the Simulink simulation the radio frequency noise floor was assumed to be 0 dBm, so the transmit power in dBm is also the signal to noise ratio (SNR = Signal Power / Noise Power dB = Signal Power dBm – Noise Power dBm). The bit error rate (BER) performance was poor because of this initial reset of the radio. At this time no learning had occurred.

D.2 Trend Step 2 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0

Figure D.4: Trend step 2 host radio data

```
CSM Final Output

Observed Channel:
Ranking Metric  = 0
BER             = 0


LTM Members
   Ranking Metric      Credits
0   0                 32767
1   1.05236            32784.2


STM Members
   Ranking Metric      Credits
0   0                 32767
1   1.05236            32784.2


STM Winner
STM Index       = 0
Ranking Metric  = 0
Credits         = 32767
bid             = 327.67
Goals (3):
    Fitness        Weight
    1             128
    10000          128
    20000          128
```

Figure D.5: Trend step 2 engine data

In trend step 2 shown in Figures D.4 and D.5 the engine increased its data rate to 9 Mbps, changed modulation and coding to radio mode 2 QPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) performance was reduced to zero due to the engines changes in radio parameters. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate.

D.3 Trend Step 3 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0

Figure D.6: Trend step 3 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 0
BER = 0

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1.05236	32784.2

STM Members

	Ranking Metric	Credits
0	0	32767
1	1.05236	32784.2

STM Winner
STM Index = 0
Ranking Metric = 0
Credits = 32767
bid = 327.67
Goals (3):

Fitness	Weight
1	128
10000	128
20000	128

Figure D.7: Trend step 3 engine data

In trend step 3 shown in Figures D.6 and D.7 the engine increased its data rate to 18 Mbps, changed modulation and coding to radio mode 4 16-QAM ¾ rate, and decreased its power to 19 dB. The bit error rate (BER) performance was maintained at zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate.

D.4 Trend Step 4 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER	
1	1	1	6250000	1	0.057202	
2	1	2	9375000	22	0	
3	1	4	18750000	19	0	
4	1	2	9375000	9	0	

Figure D.8: Trend step 4 host radio data

```
CSM Final Output

Observed Channel:
Ranking Metric  = 0
BER             = 0


LTM Members
   Ranking Metric      Credits
0   0                32767
1   1.05236            32784.2


STM Members
   Ranking Metric      Credits
0   0                32767
1   1.05236            32784.2


STM Winner
STM Index       = 0
Ranking Metric  = 0
Credits         = 32767
bid             = 327.67
Goals (3):
    Fitness       Weight
    1            128
    10000          128
    20000          128
```

Figure D.9: Trend step 4 engine data

In trend step 4 shown in Figures D.8 and D.9 the engine decreased its data rate to 9 Mbps, changed modulation and coding back to radio mode 2 QPSK ¾ rate, and decreased its power to 9 dB. The bit error rate (BER) performance was maintained at zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate. In this case it traded off lower data rate for lower power since each of its goals was equally weighted.

D.5 Trend Step 5 – Flat Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174

Figure D.10: Trend step 5 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 1
BER = 0.00017378

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5

STM Winner
STM Index = 1
Ranking Metric = 1
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
20000	128

Figure D.11: Trend step 5 engine data

In trend step 5 shown in Figures D.10 and D.11 the engine encountered a flat fading channel, increasing its data rate to 56 Mbps, changed modulation and coding to radio mode 8 64-QAM ¾ rate, and increased its power to 25 dB. The bit error rate (BER) increased to 1.74×10^{-4} due the more challenging channel. Note that the CSM instructed the WSGA to minimize BER and maximize data rate, in this case ignoring the need to conserve power.

D.6 Trend Step 6 – Flat Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0

Figure D.12: Trend step 6 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 0
BER = 0

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2

STM Winner
STM Index = 0
Ranking Metric = 0
Credits = 32767
bid = 327.67
Goals (3):

Fitness	Weight
1	128
10000	128
20000	128

Figure D.13: Trend step 6 engine data

In trend step 6 shown in Figures D.12 and D.13 the engine decreased its data rate back to 18 Mbps, changed modulation and coding to radio mode 4 16-QAM ¾ rate, and increased its power to 26 dB. The bit error rate (BER) performance in the flat fading channel was reduced to zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate. In this case the engine traded off lower data rate for better BER performance.

D.7 Trend Step 7 – Flat Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434

Figure D.14: Trend step 7 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 1
BER = 0.00043429

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5

STM Winner
STM Index = 1
Ranking Metric = 1
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
10000	128

Figure D.15: Trend step 7 engine data

In trend step 7 shown in Figures D.14 and D.15 the engine increased its data rate to 25 Mbps, changed modulation and coding to radio mode 5 16-QAM ½ rate, and decreased its power to 19 dB. The bit error rate (BER) increased to 4.34×10^{-4}. Note that the CSM instructed the WSGA to minimize BER and minimize power, ignoring the data rate.

D.8 Trend Step 8 – Dispersive Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0

Figure D.16: Trend step 8 host radio data

```
CSM Final Output

Observed Channel:
Ranking Metric  = 0
BER             = 0


LTM Members
   Ranking Metric      Credits
0   0                  32767
1   1                  32767
2   1.05236            32784.2


STM Members
   Ranking Metric      Credits
0   0                  32767
1   1                  32767
2   1.05236            32784.2


STM Winner
STM Index       = 0
Ranking Metric  = 0
Credits         = 32767
bid             = 327.67
Goals (3):
    Fitness      Weight
    1            128
    10000        128
    20000        128
```

Figure D.17: Trend step 8 engine data

In trend step 8 shown in Figures D.16 and D.17 the engine encountered a dispersive fading channel, decreasing its data rate to 9 Mbps, changed modulation and coding to radio mode 2 BPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) was reduced to zero. Note that the CSM instructed the WSGA to balanced fitness functions equally, minimize BER, minimize power, and maximize data rate.

Note that a problem occurred when simulating the dispersive channel. No software phase lock loop was implemented, so phase angle drift corrections were calculated manually and applied to each channel and modulation. This approach did not work when the dispersive channel generated inter-symbol interference, so the error rate values for the dispersive channels are inaccurate and therefore were disregarded.

D.9 Trend Step 9 – Dispersive Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041

Figure D.18: Trend step 9 host radio data

```
CSM Final Output

Observed Channel:
Ranking Metric  = 7.78949
BER             = 0.10041


LTM Members
   Ranking Metric      Credits
0  0                  34991.7
1  1                  34664
2  1.05236            34646.9
3  7.78949            32767


STM Members
   Ranking Metric      Credits
0  0                  34991.7
1  1                  34664
2  1.05236            34646.9
3  7.78949            32767


STM Winner
STM Index       = 3
Ranking Metric  = 7.78949
Credits         = 32767
bid             = 327.67
Goals (2):
    Fitness       Weight
    1             128
    20000         128
```

Figure D.19: Trend step 9 engine data

In trend step 9 shown in Figures D.18 and D.19 the engine increased its data rate to 37 Mbps, changed modulation and coding to radio mode 6 16-QAM ¾ rate, and decreased its power to 9 dB. The bit error rate (BER) increased to $1x10^{-1}$ due to the engine attempting to maximize data rate in a very challenging wireless channel. Note that the CSM instructed the WSGA to minimize BER and maximize data rate, ignoring power.

D.10 Trend Step 10 – Dispersive Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042

Figure D.20: Trend step 10 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 6.1726
BER = 0.13042

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	34134.2
2	1.05236	34117.1
3	6.1726	32767
4	7.78949	32969.1

STM Members

	Ranking Metric	Credits
0	1	34134.2
1	1.05236	34117.1
2	6.1726	32767
3	7.78949	32969.1

STM Winner
STM Index = 2
Ranking Metric = 6.1726
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
2	128
20000	128

Figure D.21: Trend step 10 engine data

In trend step 10 shown in Figures D.20 and D.21 the engine decreased its data rate to 18 Mbps, changed modulation and coding to radio mode 4 QPSK ¾ rate, and increased its power to 26 dB. The bit error rate (BER) increased to 1.3×10^{-1} due to the engine attempting to maximize data rate in a very challenging wireless channel. Note that the CSM recognized the channel as a Rayleigh channel and instructed the WSGA to minimize BER using fitness function 2, maximize data rate, and ignore power.

D.11 Trend Step 11 – Dispersive Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042

Figure D.22: Trend step 11 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 6.1726
BER = 0.13042

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	34134.2
2	1.05236	34117.1
3	6.1726	32767
4	7.78949	32969.1

STM Members

	Ranking Metric	Credits
0	1	34134.2
1	1.05236	34117.1
2	6.1726	32767
3	7.78949	32969.1

STM Winner
STM Index = 2
Ranking Metric = 6.1726
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
20000	128

Figure D.23: Trend step 11 engine data

In trend step 11 shown in Figures D.22 and D.23 the engine maintained its data rate at 18 Mbps, modulation and coding to radio mode 4 QPSK ¾ rate, and power at 26 dB. The bit error rate (BER) stayed at 1.3×10^{-1}. Note that the CSM instructed the WSGA to minimize BER, maximize data rate, and ignore power.

D.12 Trend Step 12 – Dispersive Fading Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042

Figure D.24: Trend step 12 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 6.1726
BER = 0.13042

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	34134.2
2	1.05236	34117.1
3	6.1726	32767
4	7.78949	32969.1

STM Members

	Ranking Metric	Credits
0	1	34134.2
1	1.05236	34117.1
2	6.1726	32767
3	7.78949	32969.1

STM Winner
STM Index = 2
Ranking Metric = 6.1726
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
20000	128

Figure D.25: Trend step 12 engine data

In trend step 12 shown in Figures D.24 and D.25 the engine maintained its data rate at 18 Mbps, modulation and coding to radio mode 4 QPSK ¾ rate, and power at 26 dB. The bit error rate (BER) stayed at $1.3x10^{-1}$. Note that the CSM instructed the WSGA to minimize BER, minimize power, and maximize data rate.

D.13 Trend Step 13 – Rician Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0

Figure D.26: Trend step 13 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 0
BER = 0

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2
3	6.1726	34461.9
4	7.78949	32767

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2
3	6.1726	34461.9

STM Winner
STM Index = 0
Ranking Metric = 0
Credits = 32767
bid = 327.67
Goals (3):

Fitness	Weight
1	128
10000	128
20000	128

Figure D.27: Trend step 13 engine data

In trend step 13 shown in Figures D.26 and D.27 the engine maintained its data rate at 18 Mbps, modulation and coding to radio mode 4 QPSK ¾ rate, and power at 26 dB. The bit error rate (BER) was reduced to zero because the engine encountered a Rician channel. Note that the CSM instructed the WSGA to minimize BER and maximize data rate, ignoring power.

D.14 Trend Step 14 – Rician Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0

Figure D.28: Trend step 14 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 0
BER = 0

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2
3	6.1726	34461.9
4	7.78949	32767

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2
3	6.1726	34461.9

STM Winner
STM Index = 0
Ranking Metric = 0
Credits = 32767
bid = 327.67
Goals (3):

Fitness	Weight
1	128
10000	128
20000	128

Figure D.29: Trend step 14 engine data

In trend step 14 shown in Figures D.28 and D.29 the engine increased its data rate to 54 Mbps, changed its modulation and coding to radio mode 8 64-QAM ¾ rate, and decreased its power to 23 dB. The bit error rate (BER) remained at zero. Note that the CSM instructed the WSGA to balance fitness functions equally, minimize BER, minimize power, and maximize data rate.

D.15 Trend Step 15 – Rician Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688

Figure D.30: Trend step 15 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 14.5263
BER = 0.0066881

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	36854.4
3	6.1726	35176.6
4	7.78949	34646.8
5	14.5263	32767

STM Members

	Ranking Metric	Credits
0	1.05236	36854.4
1	6.1726	35176.6
2	7.78949	34646.8
3	14.5263	32767

STM Winner
STM Index = 3
Ranking Metric = 14.5263
Credits = 32767
bid = 327.67
Goals (3):

Fitness	Weight
2	128
10000	255
20000	128

Figure D.31: Trend step 15 engine data

In trend step 15 shown in Figures D.30 and D.31 the engine decreased its data rate to 37 Mbps, changed its modulation and coding to radio mode 6 16-QAM ¾ rate, and decreased its power to 9 dB. The bit error rate (BER) increased to 6.68×10^{-3}. Note that the CSM instructed the WSGA to give highest priority to minimizing power, while equally weighting the need to minimize BER and maximize data rate.

D.16 Trend Step 16 – Rician Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05

Figure D.32: Trend step 16 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 1
BER = 8.6828e-005

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5
3	6.1726	34134.2
4	7.78949	32767
5	14.5263	32767

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5
3	6.1726	34134.2

STM Winner
STM Index = 1
Ranking Metric = 1
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
10000	128

Figure D.33: Trend step 16 engine data

In trend step 16 shown in Figures D.32 and D.33 the engine decreased its data rate to 18 Mbps, changed its modulation and coding to radio mode 4 QPSK ¾ rate, and maintained its power at 9 dB. The bit error rate (BER) decreased to 8.68×10^{-5}. Note that the CSM achieved this better error rate performance by instructing the WSGA to minimize BER and minimize power, ignoring data rate.

This section demonstrated that the cognitive engine is capable of operating in unknown wireless channels, balancing radio goals based on its ability to learn about the wireless environment it is operating in.

CR Engine versus Traditional Adaptive Radio Controller – Known Channel

The previous section analyzed the cognitive engine's ability to operate in unanticipated and unknown wireless channels. This section analyzes the engine as it encounters a wireless channel that it has seen before, in this case an AWGN wireless channel.

The trend steps highlighted in grey in Figure D.34 show the AWGN channel that the engine encounters following the channels of the previous section.

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05
17	1	2	9375000	22	0
18	1	7	50000000	6	0.018333
19	1	8	56250000	19	0.000869
20	1	2	9375000	22	0

Figure D.34: CR toolset trace showing cognitive engine reacting to known channel

D.17 Trend Step 17 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05
17	1	2	9375000	22	0

Figure D.35: Trend step 17 host radio data

```
CSM Final Output

Observed Channel:
Ranking Metric  = 0
BER            = 0


LTM Members
   Ranking Metric      Credits
0   0                 32767
1   1                 32767
2   1.05236            32784.2
3   6.1726            34461.9
4   7.78949            32767
5   14.5263            32767


STM Members
   Ranking Metric      Credits
0   0                 32767
1   1                 32767
2   1.05236            32784.2
3   6.1726            34461.9


STM Winner
STM Index      = 0
Ranking Metric  = 0
Credits        = 32767
bid           = 327.67
Goals (3):
    Fitness       Weight
    1            128
    10000         128
    20000         128
```

Figure D.36: Trend step 17 engine data

In trend step 17 shown in Figures D.35 and D.36 the engine encounters a channel that it has seen before, an AWGN channel. It decreased its data rate to 9 Mbps, changed its modulation and coding to radio mode 2 BPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) decreased to zero. Note that the CSM achieved this better error rate performance by instructing the WSGA to balance fitness functions equally, minimize BER, minimize power, and maximize data rate.

D.18 Trend Step 18 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05
17	1	2	9375000	22	0
18	1	7	50000000	6	0.018333

Figure D.37: Trend step 18 host radio data

```
CSM Final Output

Observed Channel:
Ranking Metric  = 13.8095
BER             = 0.018333


LTM Members
   Ranking Metric     Credits
0  0                 32767
1  1                 32767
2  1.05236           32767
3  6.1726            34941.7
4  7.78949           34411.9
5  13.8095           32767
6  14.5263           32674.2


STM Members
   Ranking Metric     Credits
0  6.1726            34941.7
1  7.78949           34411.9
2  13.8095           32767
3  14.5263           32674.2


STM Winner
STM Index      = 2
Ranking Metric  = 13.8095
Credits        = 32767
bid            = 327.67
Goals (3):
    Fitness      Weight
    2            128
    10000        255
    20000        128
```

Figure D.38: Trend step 18 engine data

In trend step 18 shown in Figures D.37 and D.38 the engine increases its data rate to 50 Mbps, changed its modulation and coding to radio mode 7 64-QAM ½ rate, and decreased its power to 6 dB. The bit error rate (BER) increased to 1.83×10^{-2}. Note that the CSM instructed the WSGA to give highest priority to minimizing power, while equally weighting the need to minimize BER and maximize data rate.

D.19 Trend Step 19 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05
17	1	2	9375000	22	0
18	1	7	50000000	6	0.018333
19	1	8	56250000	19	0.000869

Figure D.39: Trend step 19 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 1
BER = 0.00086889

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5
3	6.1726	34134.2
4	7.78949	32767
5	13.8095	32767
6	14.5263	32767

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32456.5
3	6.1726	34134.2

STM Winner
STM Index = 1
Ranking Metric = 1
Credits = 32767
bid = 327.67
Goals (2):

Fitness	Weight
1	128
10000	128

Figure D.40: Trend step 19 engine data

In trend step 19 shown in Figures D.39 and D.40 the engine increases its data rate to 56 Mbps, changed its modulation and coding to radio mode 8 64-QAM ¾ rate, and increased its power to 19 dB. The bit error rate (BER) decreased to $8.69\text{x}10^{-4}$. Note that the CSM instructed the WSGA equally weighting the need to minimize BER and minimize power, ignoring data rate. In this case since engine weighted the goals equally, it traded its goal of minimizing power for better BER performance and achieved better data rate in the process. Note that since the engine had already learned about the AWGN channel it was able to recognize that the observed wireless channel was an AWGN channel and quickly achieve higher data rates with decent BER much quicker than in trend steps one through four.

D.20 Trend Step 20 – AWGN Channel

Trend Number	Channel Type	Radio Mode	Data Rate	SNR	BER
1	1	1	6250000	1	0.057202
2	1	2	9375000	22	0
3	1	4	18750000	19	0
4	1	2	9375000	9	0
5	2	8	56250000	25	0.000174
6	2	4	18750000	26	0
7	2	5	25000000	19	0.000434
8	3	2	9375000	22	0
9	3	6	37500000	9	0.10041
10	3	4	18750000	26	0.13042
11	3	4	18750000	26	0.13042
12	3	4	18750000	26	0.13042
13	4	4	18750000	26	0
14	4	8	56250000	23	0
15	4	6	37500000	9	0.006688
16	4	4	18750000	9	8.68E-05
17	1	2	9375000	22	0
18	1	7	50000000	6	0.018333
19	1	8	56250000	19	0.000869
20	1	2	9375000	22	0

Figure D.41: Trend step 20 host radio data

CSM Final Output

Observed Channel:
Ranking Metric = 0
BER = 0

LTM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2
3	6.1726	34461.9
4	7.78949	32767
5	13.8095	32767
6	14.5263	32767

STM Members

	Ranking Metric	Credits
0	0	32767
1	1	32767
2	1.05236	32784.2
3	6.1726	34461.9

STM Winner
STM Index = 0
Ranking Metric = 0
Credits = 32767
bid = 327.67
Goals (3):

Fitness	Weight
1	128
10000	128
20000	128

Figure D.42: Trend step 20 engine data

In trend step 20 shown in Figure D.41 and D.42 the engine decreases its data rate to 9 Mbps, changed its modulation and coding to radio mode 2 BPSK ¾ rate, and increased its power to 22 dB. The bit error rate (BER) decreased to 0. Note that the CSM achieved this better error rate performance by instructing the WSGA to balance fitness functions equally, minimize BER, minimize power, and maximize data rate.

This section illustrated the cognitive engine's ability to recognize a known channel and quickly find a balance of radio parameters that meet the goals for that channel.

Appendix E: NSF IGERT IREAN Research Interactions

2004 - Cognitive Radio as a Multidisciplinary Research Theme

Dr. Charles W. Bostian
Electrical Engineering

With NSF support VT has developed a cognitive engine (a software system, which, together with its associated hardware, is capable of modifying its behavior in response to conditions that change quickly and in unexpected ways). Our cognitive engine can turn any radio transceiver with “meters” (outputs like data rate that indicate current performance) and “knobs” (inputs like channel frequency) into a cognitive radio (a radio that behaves like an intelligent being, sensing its environment and modifying its behavior to meet its goals). Our work in cognitive radio started with a single IGERT fellow (Rieser) who soon began a close collaboration with a (then) undergraduate classmate (Rondeau) and subsequently expanded to a team of five ECE Ph.D. students co-led by Rieser and Rondeau and including IGERT fellow Maldonado. Cognitive radio became the basis of a project called CRANIA (Cognitive Radio for Advanced Networking and Integrated Access) for the IREAN simulation and optimization course (taught by Koelling) and of a large NSF NetS proposal involving 10 faculty in 5 departments (ECE, FIN, BIT, ECON, GEOG). Rondeau and Rieser are offering an informal seminar for interested faculty and students covering: 1. Our concept of machine learning. 2. Genetic Algorithms and how they can realize an intelligent machine by being able to leverage sensing data, user input, and legal/regulatory requirements. 3. Distributed cognition -- learning and optimization augmented by intelligence throughout the network.

Appendix E: NSF IGERT IREAN Research Interactions

This appendix documents the various research summaries that I wrote describing the posters and presentations I presented at the IREAN research workshops.

2004 - Biologically Inspired Cognitive Radio Test bed Based on GAs

Christian James Rieser
Electrical Engineering
(Advisor: Dr. Charles W. Bostian)

Emerging cognitive radio technology may revolutionize wireless spectrum access technology, policy, and business. This presentation details a biologically inspired cognitive radio formalism based on genetic algorithms. A cognitive radio test bed utilizing the formalism is presented along with experimental results from the demonstration system, including evolution of an adaptive host radio to avoid an interfering radio system.

By our definition, the technical characteristics of fixed radios are set at the time of manufacture. An adaptive radio can respond to channel conditions that represent one of a finite set of anticipated events. Adaptive radios use artificial intelligence (AI) algorithms that are basically a series of “if,then,else” algorithms. A cognitive radio can respond intelligently to an unanticipated event – i.e., a channel that it has never encountered before.

The proposed cognitive radio formalism consists of a multi-tiered genetic algorithm architecture that allows sensing of a wireless channel at the waveform or symbol level using a broadband channel sounder and Wireless Channel Genetic Algorithm (WCGA), on the fly evolution of the radio’s operational parameters using a Wireless System Genetic Algorithm (WSGA), and cognitive functions through use of a learning classifier, meta-genetic algorithms, short and long term memory and control embodied in the Cognitive System Monitor (CSM).

Experimental results from the cognitive radio test bed demonstrate that the genetic algorithm (GA) based cognitive radio formalism provides significant flexibility in the face of unknown or changing wireless channels that are prevalent in disaster response and emergency communications scenarios. Research is being pursued to extend the formalism to enable a cognitive medium access control (MAC) data link layer based on genetic algorithms that would provide dynamic quality of service (QOS) functions based on available wireless spectrum resources and paths of opportunity.

This work is inherently interdisciplinary, influenced by engineering implementation, business, and policy issues. As such I maintain a research dialogue with several disciplines including Electrical Engineering, Genetics and Bioinformatics, Cognitive Development, Physics and Mathematics, Industrial Systems Engineering, Business and Policy.

Appendix E: NSF IGERT IREAN Research Interactions

2003 - Biologically Inspired Cognitive Wireless Layer 1 and 2 (L12) Functionality

Christian Rieser
Electrical Engineering
(Advisor: Dr. Charles Bostian)

Wireless communications systems (‘radios”) can be described as fixed, adaptive, or cognitive. The technical characteristics of **fixed** radios are set at the time of manufacture. An **adaptive** radio can respond to channel conditions that represent one of a finite set of anticipated events. Adaptive radios use artificial intelligence (AI) algorithms that are basically a series of “if,then,else” algorithms. A **cognitive** radio can respond intelligently to an unanticipated event – i.e., a channel that it has never encountered before. This presentation describes a novel and computationally efficient method to realize a truly cognitive radio based on genetic algorithms.

An immediate market for this technology is in military and disaster communications, where radio systems must work under changing and unanticipated circumstances and in the presence of hostile jammers and interferers. The long-term market is in civilian radio communications systems like cellular telephones where spectrum and battery power are at a premium and in which the radio sets must continuously adapt to conserve these resources.

This research has been pursued from the start with input from a number of disciplines, since the formulation of constraints and requirements for a computationally efficient method to realize a truly cognitive radio based on genetic algorithms requires synthesis of system issues that span across several disciplines. My PhD multidisciplinary research activities include dialogue and interaction in Electrical Engineering, Genetics and

Bioinformatics, Cognitive Development, Physics and Mathematics, and Business and Policy.

The presentation includes information on my background, the motivation for the research, a definition of the research problem, the research methodology including concept and framework, research timeline, research personnel, and a summary of the research.