Statistical Experimental Design Framework for Cognitive Radio

Ashwin E. Amanna

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

Jeffrey H. Reed, Chair
Tamal Bose
Allen B. MacKenzie
Madhav V. Marathe
Jung-Min Park

March 19, 2012
Blacksburg, Virginia

Keywords: Design of Experiments (DOE), Response Surface Methodology (RSM), Taguchi Designs, Cognitive Radio, Software-Defined Radio, Decision Making, Case-Based Reasoning

Statistical Experimental Design Framework for Cognitive Radio

Ashwin E. Amanna

(ABSTRACT)

This dissertation presents an empirical approach to identifying decisions for adapting cognitive radio parameters with no *a priori* knowledge of the environment. Cognitively inspired radios, attempt to combine observed metrics of system performance with artificial intelligence decision-making algorithms. Current architectures trend towards hybrid combinations of heuristics, such as genetic algorithms (GA) and experiential methods, such as case-based reasoning (CBR). A weakness in the GA is its reliance on limited mathematical models for estimating bit error rate, packet error rate, throughput, and signal-to-noise ratio. The CBR approach is similarly limited by its dependency on past experiences. Both methods have potential to suffer in environments not previously encountered. In contrast, the statistical methods identify performance estimation models based on exercising defined experimental designs. This represents an experiential decision-making process formed in the present rather than the past. There are three core contributions from this empirical framework: 1) it enables a new approach to decision making based on empirical estimation models of system performance, 2) it provides a systematic method for initializing cognitive engine configuration parameters, and 3) it facilitates deeper understanding of system behavior by quantifying parameter significance, and interaction effects. Ultimately, this understanding enables simplification of system models by identifying insignificant parameters.

This dissertation defines an abstract framework that enables application of statistical approaches to cognitive radio systems regardless of its platform or application space. Specifically, it assesses factorial design of experiments and response surface methodology (RSM) to an over-the-air wireless radio link. Results are compared to a benchmark GA cognitive engine. The framework is then used for identifying software-defined radio initialization settings. Taguchi designs, a related statistical method, are implemented to identify initialization settings of a GA.

Grant Information

The research presented in this investigation was supported by the Federal Railroad Administration, Office of Research and Development, FRA Grant No. DTFR53-09-H-00021. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the view of the Federal Railroad Administration and/or U.S. DOT. This research was also partially supported by the Institute for Critical Technologies and Applied Science (ICTAS) of Virginia Tech.

Dedication

This dissertation is dedicated to Ralph S. Amanna in answer to his question on why I haven't invented anything yet.

Acknowledgments

My heartfelt thanks go to the many people who helped me keep moving forward.

- First and foremost, this would not have been possible without my wife Karen. Her enduring support and understanding was beyond measure. Thank you to my sons, Ansel and Lucas, for letting me do my homework with you at the kitchen table.
- To my committee, Drs. Bose, MacKenzie, Marathe, and Park, thank you for the constant questioning, advice, and critique to make me a better researcher.
- To my advisor, Dr. Reed, thank you for your continual belief and encouragement.
- To my students, Matthew Price, Manik Gadhiok, Daniel Ali, and David Gonzalez Fitch, it has been a great privilege working and growing with you.
- To my fellow graduate students in the MPRG, my greatest regret is that I could not be a regular graduate student with you. Special thanks to Dr. Joe Gaeddert for his labor of love, *liquid* DSP.
- To those who came before me, Drs. Ingrid Burbey, Carlos Aguayo Gonzalez, Joe Gaeddert, An He, Tim Newman, Haris Volos, Jody Neel, S.M. Hasan, and Tom Rondeau, thank you for blazing the trail and inspiring me.
- To my sponsors, the Federal Railway Administration (FRA) and Institute for Critical Technology and Applied Science (ICTAS), thank you for making this possible.
- To fellow collaborators and co-workers, Drs. Clancy, McGwier, and Thamvichai, thank you for your thoughtful guidance.
- To my friends, Dr. John McDowell, Melissa Wynn, and Sunjay Nair, thank you for the encouragement.
- To my mother and father, thank you for inspiring.

Attributions

Several colleagues aided in the writing and research behind several chapters which have been published in journal or conference form. A brief description of their contribution is included here.

Chapter 5: Portions of Taguchi based parametric optimization have been accepted for publication by the *EURASIP Journal on Wireless Communications and Networking Special Issue on Cognitive Radio* and reused by permission. The article is currently in press.

Daniel Ali was a co-author and assisted in the data analysis. Daniel was a graduate research assistant in the Electrical Engineering department of Virginia Tech.

Manik Gadhiok was a co-author and assisted in the development of the MATLAB architecture for the cognitive engine. Manik was a graduate research assistant in the Electrical Engineering department of Virginia Tech.

Matthew Price was a co-author and assisted in the development of the case based reasoner and genetic algorithm modules of the cognitive engine. Matt was a graduate research assistant in the Electrical Engineering department of Virginia Tech.

Dr. Jeffrey Reed was a co-author and provided general guidance and advice on the research. Dr. Jeffrey Reed is the Willis G. Worcester professor in the Bradley Department of Electrical and Computer Engineering and the director of Wireless at Virginia Tech.

Chapter 6: Grey system theory application to wireless communications. Portions of this chapter were accepted for publication in the *Journal of Analog Integrated Circuits and Signal Processing* [1] and reused by permission. I was the main author and developed the original concept.

Dr. Kay Thamvichai was a visiting associate professor from St. Cloud State. Dr. Thamvichai was a co-author and adapted the Grey Relational Analysis algorithm towards an automatic modulation classification application by incorporating α-profile as an extracted feature.

Matthew Price was a co-author and further developed the simulations and performed experiments. Matt was a graduate research assistant in the Electrical Engineering department of Virginia Tech.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Current cognitive radio architecture must contend with incomplete observations about the surroundings and imprecise mathematical models. Consequently, there is an ongoing need for strategies to learn about the environment as well as developing improved performance estimation models. This dissertation investigates an empirical approach to addressing the learning process based on statistical experimental design. Experimental design is defined as a systematic approach to performing testing where purposeful changes are made to system inputs. The goal of experimental design is to identify the reasons for changes that are observed in the outputs. It can provide representative knowledge of system performance early in the learning cycle. Gaining this knowledge through statistical methods combats many limitations in existing architectures including reliance on incomplete theoretical models and knowledge of channel conditions.

1.1 Problem Description

Wireless communications has experienced significant advances in the last decade. Initially, a radio was defined as technology that wirelessly transmits or receives data through electromagnetic radiation [5]. Traditional radios performed all processing of the radio frequency within hardware. Hence, the functionality of the radio was static. Recent advances in computer processing provided the foundation for software-defined radios (SDR). The reconfigurable nature of software enabled

a new level of dynamic radio frequency processing. This capability allows the radio to modify operational parameters previously fixed in hardware. This shift from static to dynamic operations has profound implications to advancing wireless communications in the areas of interoperability, coexistence, improving spectrum efficiency, and enhancing wireless link performance.

SDR created the foundation for what is known today as cognitive radio. The approach integrates the reconfigurable nature of SDR with concepts of human cognition, where cognition is defined as the capability to perceive, retain, and reason about information [5]. Based on awareness of the environment, a cognitive radio automatically changes the SDR's configurations in order to achieve predefined goals.

The key concepts of manipulating SDR configurations and achieving defined goals led to the colloquial expressions of 'reading the meters' and 'adjusting the knobs'. Observations of system performance as well as environmental conditions provided the information necessary for a cognitive engine to decide how to adjust the radio configuration parameters. This decision is then implemented by reconfiguring the software of the radio.

Fundamentally, there exists a set of **inputs**, in the form of SDR knobs, that result in desired **outputs**. The actual hardware platform and application define the specifics of the inputs and outputs. For example, consider a SDR wireless link consisting of a transmitter and a receiver. The transmitter might have configurable inputs that include transmit power, modulation, and coding. The goals of this link are to transmit a file at the highest throughput possible while minimizing packet error rate (PER). A solution to this problem would be a specific set of inputs that achieve these combined goals. Note that the experimental design literature often uses the terminology of 'responses' for outputs and 'response factors' for inputs. This dissertation uses inputs and outputs except when referring to a specific experimental design methods.

A general problem in cognitive radio is how to identify the settings of the inputs that meet the desired objectives. One can approach the problem with an exhaustive search that tests every possible combination of input parameters across the entire range of each input. However, performing this type of search in a real-world application is unrealistic. Alternatively, one can recursively select inputs at random until an acceptable solution is identified. This too has drawbacks in terms of reliance on randomness for success. A key problem in both approaches is the lack of a defined

starting point for the search. A starting point that is closer to the optimal setting would enable a search algorithm to reach an acceptable solution faster.

Current cognitive engine architectures attempt to address this problem by incorporating past experience. Case-based reasoning (CBR) draws on successful past decisions. The concept is that a solution to a new situation can be identified from a past similar experience. The decisions implemented in these past experiences provide potential starting points for a random-based search. Additionally, the engine could implement a past solution as is. However, when no past experience exists, the architecture must again rely on randomness to identify an acceptable solution.

The specific problem this dissertation addresses is how to efficiently search the inputs with no past experience. This dissertation uses known statistical experimental design methods, not traditionally used in wireless communications, to systematically probe the inputs. Similar to the benefits of CBR, the identified solution from experimental methods can provide a good starting point for a random-based search. Furthermore, existing heuristic cognitive engines rely on theoretical models that may be imprecise and require knowledge of the current noise and channel environments. The experimental design approach relies only on controlling the inputs and observing the outputs to draw all conclusions.

In addition, this dissertation expands the concept of inputs to include the configuration parameters of a cognitive engine. Likewise, it expands outputs to incorporate cognitive engine performance in addition to traditional radio performance. For example, a genetic algorithm (GA) decision-making module requires initialization of crossover rate, mutation rate, population size, and maximum generations. Similarly, a hybrid GA/CBR cognitive engine will have performance criteria that includes time-to-decision, or percentage use of the CBR compared to the GA [3]. The experimental design framework is extended to the problem of how to identify initial configuration settings for cognitive engines that lead to desirable radio and engine performance.

Further benefits of an experimental design approach include quantification of input significance and interaction effects between inputs. Two inputs interact if the effect of one input is different depending on the level of the other input. More importantly, identification of insignificant inputs enables simplification of output estimation models. This knowledge extends the understanding of cognitive radio operations to a deeper level than previously researched.

The scope of implementing experimental design is confined to a single wireless link realized in simulation and over-the-air. Emphasis is placed on the use and implementation of experimental design and not on identifying the best methods for multi-criteria optimization for solving the models.

1.1.1 Historical Context

The motivation for this research lies in limitations of current architectures. This section introduces historical context of current cognitive architectures. Foundational work in cognitive radio architectures has trended towards heuristic optimization algorithms and experiential case-based learning. Inspired by biological systems, early cognitive engine architectures drew upon GA. The pioneering work of Rondeau and Rieser adapted the powerful search techniques of GA to a cognitive radio application space [6]. Driven by a combination of random search and attributes from past generations, the GA-based cognitive radio engine was able to identify solutions.

The optimization process identifies a waveform based on the success of objective functions. Objective functions, also known as utility functions, are typically weighted and combined into a single quantification of fitness. The GA must make estimates of objective functions for any potential solutions that it identifies. These estimates are driven by mathematical models inherent to the GA. Past implementation incorporated a power spectral density sensor to provide information about the external interference environment. This enabled the GA to customize the objective functions and associated weights [7]. Once the GA identifies a solution that maximizes the objective functions, it implements the selected waveform on the SDR platform and measures final performance. The difference between the resulting performance and the estimated performance defines the overall success of the solution.

However, the GA approach suffers from several drawbacks. The partial reliance on random search can be considered a weakness. Repeatability is difficult and places dependency on the random selection of the initial parents. Convergence of a GA may not occur fast enough to meet the changing conditions of the environment. Other pertinent factors to success include the strength of the initial parents and the fitness algorithms for discerning if a potential solution would produce suitable performance.

The most limiting aspects, however, pertain to the fitness estimation procedures inherent in the GA approach. Accurate estimation of objective functions requires specific knowledge of system conditions. The required knowledge includes channel conditions, signal to noise ratio (SNR) at the receiver, estimated noise power at the receiver, and estimated path loss [7](page 57). Additionally, the effect of interference is difficult to incorporate into the estimation algorithms because the distribution of the interference power is unknown.

Rondeau and Rieser expressed the need for dynamic fitness estimations based on the current link conditions [8]. This requires a higher-layer of intelligence and a sensor for measuring existing noise. The reliance on a sensor and additional logic required increased complexity. Hence, the opportunity to misinterpret sensor data increases. Rondeau showed how errors in the power spectral density sensor logic can lead to incorrect frequency selection [6] (page 112). Without a sensor to aid in the noise measurement, a single set of GA estimation procedures required adjustment for different environments [9].

To address these drawbacks, researchers identified decision making based on past experiences to provide better parent initialization of the GA [6, 10]. CBR algorithms incorporated both elementary algorithms and a means to learn from new solutions that showed improvements in decision time. Past experiences that are similar enough to the current situation are implemented as-is without modification. If no solution met predefined similarity thresholds, then the engine can fall back to the GA decision module. However, the top ranked cases can provide seeds to the GA. Now the GA starts its initial generation with some parents made up of top seeds from the CBR and the remainder from randomly generated parents. This initialization procedure provides a better starting position than purely random seeds. If the new solution identified from the GA shows performance improvements, then appending the solution description to the library enables long term learning [3, 11]. However, lack of experience is a key limitation to the validity of CBR. If the engine has not experienced a similar situation some time in the past, then there is limited benefit of the CBR. Training or pre-seeding the case-base library with known solutions and descriptions of the environment is key to success.

Solutions identified from either the GA or CBR suffer a core limitation. The outcome of the identified solution is unknown until the engine implements the decisions and checks resulting per-

formance for improvement. The GA relies on objective function estimations to predict the outcome of a potential solution, while the CBR leverages past successes. Hence, both CBR and the GA have weaknesses in situations not previously seen before.

The design of experiments methodology is fundamentally different than the current approach. It does not rely on theoretical models of performance or past experience. In contrast, only empirically obtained metrics drive the development of performance estimation models. Rather than basing a decision on similarity to past experience, the solution is obtained from recently obtained empirical data.

1.1.2 Testing for cognitive radio

Design of experiments has strong potential to support testing of cognitive radios, albeit from a non-traditional perspective. Cursory review of testing within cognitive radio indicates virtually complete focus on functional testing and meeting design specifications [12, 13, 14, 15, 16]. This is expected given the heavy focus on dynamic spectrum access in early cognitive radio and the application-driven nature of deployments.

Experimental methods take on more importance as one moves beyond simple function in order to gain deeper understanding of the system. Rather than testing the system, perhaps a better viewpoint would be exercising the system across the range of available configuration parameters. Here is where design of experiments improve on traditional methods, such as manipulating one variable at a time while holding others static. The systematic procedures for varying multiple parameters at a time provide benefits in terms of efficiency as well as knowledge gain. Beneficial results include assessments of parameter significance, insignificance, and interaction effects between parameters.

1.2 Thesis

The overall problem is **how** to perform an empirical assessment of wireless and cognitive architectures with no previous training or *a priori* knowledge.

Thesis

Design of experiments can efficiently *probe* the system to elicit enough knowledge of performance to:

- Develop statistically strong estimation models of performance metrics
- Combine multivariate models to select recommended settings
- Identify recommended initialization settings for traditional cognitive engines
- Identify significant parameters and interaction effects

Ramifications include using the developed solutions as training cases for CBR or replacing the GA's estimation formulas with the empirical models. Additional implications tie to developing procedures for self-initialization of a cognitive radio node.

Research questions focus on developing a generalized approach for applying statistical methods on a cognitive radio platform. This research addresses statistical designs, performance comparisons to existing cognitive architecture, and implementation hurdles. The research questions addressed are summarized below:

1. What process is required to adapt experimental design to cognitive radio systems?
2. What are the trade offs between decision quality and reduction in the number of tests?
3. What are the limitations of experimental design for cognitive radio applications?
4. By what metrics does one establish a statistical design's performance?
5. What level of statistical fit is necessary to identify a good solution?
6. What effect do dynamic interference environments have on statistical fit?
7. How do statistical solutions compare to existing GA cognitive engines?
8. How much improvement can be achieved in cognitive engine performance by using experimental design to identify initialization parameters?

1.3 Original Contributions

My original research direction focused on similarity analysis and comparing human-based learning. At the preliminary exam, the committee recommended shifting the focus to experimental design applications to cognitive radio. Prior to the change, a new approach to automatic modulation classification was developed that resulted a conference paper and journal publication [1]. Chapter 6 discusses this contribution and is acknowledged that it ties peripherally to the main focus of the dissertation.

Specific original contributions resulting from the focus on experimental design include:

- **Identified a technique for adapting existing estimation formulas for practical application in over-the-air systems:** Estimation formulas for performance based on basic channel conditions perform poorly outside of simulation and controlled environments. A unique method for biasing the estimation based on measured conditions yielded improved performance in over-the-air operations.
- **Developed algorithm for identifying the adjustment to improve estimation performance:** An algorithm was developed based on a simple binary search to automatically identify a recommended setting for the calibration factor. Within six iterations, a suitable calibration value is identified.
- **Developed a framework to perform statistical design of experiments for cognitive radio systems:** Each cognitive radio system is unique in terms of platform, application, and overall architecture. The proposed framework abstracts a system down to the basic parameters necessary for implementing an experimental design analysis. This approach has broader impacts beyond cognitive radio and can be applied to a variety of different systems.
- **Compared performance of RSM statistical designs to a GA cognitive engine under real-world testing conditions** Statistical designs showed performance within 4% of existing methods without requiring knowledge of the channel conditions or past experience with an environment. The empirical approach showed improved performance in a dynamic environment compared to existing methods.

- **Determined that statistical fit is not a requisite for good decisions:** It is known that statistical fit degrades in dynamic interference environments. The natural assumption is that decisions derived form poorly fitting models will not perform well. The findings of this research indicate that decisions made from models with lower statistical fit can still produce acceptable decisions. The identified output models, regardless off statistical fit, identified a general pattern of behavior that led to generally good decisions.

- **Applied Taguchi designs to cognitive radio configuration optimization:** The Taguchi method was implemented on a cognitive radio decision-making module. This approach contrasts to typical *ad hoc* best guess methods or varying one factor at a time. This successfully showed how fractional factorial designs enabled identification of solutions while significantly reducing the overall number of individual tests.

1.4 Publications

1.4.1 Journals

Published or Accepted for Publication

1. **A. Amanna**, R. Thamvichai, and M. Price, "Grey System Theory Applications to Communications," *Analog Integrated Circuits and Signal Processing (AISCP)*, Springer, July, 2011, DOI: 10.1007/s10470-011-9719-1.

2. **A. Amanna**, D. Ali, M. Gadhiok, M. Price, and J.H. Reed, "Cognitive Radio Engine Parametric Optimization Utilizing Taguchi Analysis, " *EURASIP Journal on Wireless Communications and Networking Special Issue on 10 Years of Cognitive Radio*, 2012:5, DOI:10.1186/1687-1499-2012-5

3. **A. Amanna**, D. Ali, D. Gonzalez, and J.H. Reed, "Hybrid Experiential-Heuristic Cognitive Radio Engine Architecture and Implementation," *Hindawi Journal of Computer Networks and Communications: Special Issue on Trends and Applications of Cognitive Radio*, Accepted with minor revisions Jan. 2012.

Invited

1. **A. Amanna**, D. Ali, D. Gonzalez, and J.H. Reed, "Statistical Framework for Parametric Optimization of Cognitive Radio Systems," *Analog Integrated Circuits and Signal Processing (AISCP)*, Springer, Submitted for review Feb. 2012.

1.4.2 Magazines

Published

1. **A. Amanna**, M. Gadhiok, M. Price, J.H. Reed, "Railway Cognitive Radio, " *IEEE Vehicular Technology Magazine*, September 2010.

1.4.3 Published Peer Reviewed Conference Papers

1. **A. Amanna**, D. Ali, M. Gadhiok, M. Price, and J.H. Reed, "Statistical Framework for Parametric Optimization of Cognitive Radio Systems," SDR Forum and Technical Conference, Dec. 2011.

2. M. Gadhiok, **A. Amanna**, M. J. Price, J. H. Reed, "Metacognition : Enhancing The Performance of a Cognitive Radio," Proceedings of IEEE CogSIMA, Feb. 2011, pp.213-218.

3. **A. Amanna**, R. Thamvichai, and M. Price, "Grey System Theory Applications to Communications," SDR Forum and Technical Conference (SDR'10), Nov. 2010.

4. **A. Amanna**, M. Price, S. Bera, M. Gadhiok, and J.H. Reed, "Cognitive Architecture for Railway Communications," Proceedings of the 2010 ASME Rail Transportation Division Fall Technical Conference, Oct. 2010.

5. **A. Amanna** and J. Reed, "Survey of Cognitive Radio Architectures," IEEE SoutheastCon 2010, Mar. 2010.

6. **A. Amanna**, M. Ghadiok, Matthew, J. Reed, W. Siriwongpairat, and T. Himsoon, "Rail-CR: Cognitive Radio for Enhanced Railway Communications," ASME/IEEE Joint Railway Conference, Apr. 2010

1.4.4 Planned Publications

This research identified further areas requiring exploration. Planned publications include:

1. A. Amanna, D. Ali, D. Fitch, and J.H. Reed, "Statistical Response Surface Designs for Profiling Wireless Link Performance," *IEEE Transaction on Wireless Communications*

2. "*Design of Experiments for Wireless Communications.*" This paper would provide a tutorial overview of experimental design with specific application to wireless communications. *IEEE Communications Surveys and Tutorials.*

3. "*Dynamic Adjustment of Genetic Algorithm Objective Function Estimations for Improved Over-the-Air Performance*" Existing estimation formulas based on simple channel conditions perform poorly outside of simulation and controlled conditions. This paper presents a calibration technique for improving existing estimation methods under different environmental conditions. An algorithm based on binary search enables identification of the calibration factor within six iterations leading to improved performance compared to unadjusted models. *IEEE Transactions on Wireless Communications.*

4. "*Design of Experiments Training for Case Based Reasoning Cognitive Radios.*" Identified solutions to statistical performance models present a unique training tool for cognitive radio engines. The models are situation dependent and create experience that the engine can take advantage of at a later time. The situation and solution will be hand coded into a case based reasoner. The validity of the approach will be tested be rerunning the same situation on the traditional cognitive radio engine. This paper is targeted at an IEEE conference such as *DySPAN 2013.*

1.5 Organization

The remainder of this document is organized as follows. Chapter 2 presents a tutorial background on design of experiments. The key designs reviewed and compared include factorial, fractional factorial, response surface, and Taguchi designs. The analytical techniques used in assessing the performance

of designs is discussed and illustrated with a reference example. In addition, the chapter briefly summarizes related work of experimental design applications within wireless communications and networking.

Chapter 3 presents new techniques for improving cognitive radio operations in practice. First a discussion on utility and fitness functions provides a reference for comparing performance between different methods when multiple metrics are involved. Next, existing approaches used in heuristic cognitive engines are introduced. Specific limitations are address through a unique adjustment technique. An algorithm for automating selection of the adjustment level is presented. A framework for implementing statistical techniques which overcome several inherent limitations in heuristic approach is presented. The methods are presented as a complimentary alternative to existing architectures. An example on how to apply the methods is presented that illustrates response surface methodology and the Taguchi method using a single performance metric for comparison. Finally, the chapter concludes with discussion of several of the assumptions and limitations of these methods.

Chapter 4 presents a comparison, under real-world testing conditions, between two RSM designs and a calibrated GA cognitive engine. Results indicate that without adjustment, the GA engine is unable to maintain a connection under interference conditions. The RSM techniques perform within 4% of the calibrated GA in all environments and improve upon it in dynamic interference conditions. Statistical fit of the RSM designs drops in dynamic environments yet is still able to identify a good decisions.

Chapter 5 investigates fractional factorial designs by implementing a Taguchi design on GA configuration parameters. Fractional factorials are a tool for significantly reducing the number of tests compared to a traditional factorial or RSM design. This is of benefit when the number of input parameters grows higher. The trade off is that less information is gained from the analysis. The Taguchi method also views the outputs from a new perspective of variation around a mean rather than on just optimizing a mean. This perspective came from the manufacturing world where consistency in a process is valued.

Chapter 6 presents research in the area of vector similarity analysis. This research was performed prior to the preliminary exam. At the preliminary, the committee's recommendation was to focus

more on the testing aspects of cognitive radio and not on similarity analysis. The grey relational analysis algorithm quantifies vector similarity between a reference vector and set of observed vectors in a unique way. Traditional distance measures, calculate the vector difference between each case and the reference vector on an individual basis. Typically, there is no tie between one case and another case. In contrast, the grey approach incorporates the overall minimum and maximums of all the vector distances when defining the final similarity. This fundamental premise inspired a grey relational analysis approach to automatic modulation classification. While not directly tied to the current focus in statistical methods, the two algorithms have been integrated in other research. The grey approach has also been implemented in concert with Taguchi analysis to aid in optimizing multiple outputs [17, 18].

The dissertation concludes with a summary, overview of implications of the research, and discussion of further research needs.

1.6 Summary

This dissertation presents a statistical experimental design framework for cognitive radio. The approach uses known statistical methods to overcome current limitations in traditional cognitive architectures. In addition to developing empirical models of performance, the method identifies input factor significance which enables simplifying the system representation. The contributions of this dissertation extend beyond cognitive radio to other applications that can be realized in terms of controllable inputs and measurable outputs.

Chapter 2

Background

This chapter provides tutorial background on basic concepts of experimental design. The elements discussed include the most common designs, implementation steps, analysis methods, estimation models, and parameter significance. The chapter also presents related work in wireless communications and networking utilizing experimental methods.

2.1 Introduction to Experimental Design

Experimental design, also known as design of experiments (DOE), is defined as a purposeful approach to manipulating the variables of a system in order to investigate performance. The design philosophies are important due the balance they provide between information gain and efficient use of resources. The approach is particularly useful for identifying significance of each input parameter, quantifying interactions between inputs, and developing estimation models of the output.

Experimental designs are implemented by first defining the overall system parameters, including the controllable inputs, their operating ranges or discrete levels, performance metrics, and identifying a repeatable load for the system. For example, in a single software-defined radio wireless link, the inputs might include transmit power, packet size, modulation, and coding, while the outputs might include packet error rate, throughput and spectral efficiency. A file transfer across the link can act as a repeatable load.

Typically, a matrix is developed where each column represents an input parameter and each row defines the configuration for a specific test. Each row is considered a unique test while the entire matrix is considered a complete design. A single test is implemented by setting the inputs according to the specific row of the design matrix, placing a load across the system, and recording the resulting outputs. This is repeated until all the tests in the design matrix have been implemented. A single design can be replicated to improve the measure of experimental error, if desired. Upon completion of the design, the results are analyzed using several well researched techniques.

This section reviews the following experimental designs: factorial design, response surface methodology (RSM), and fractional factorial including Taguchi designs. A reference example is introduced that illustrates a complete analysis of a full factorial and central composite design (CCD) RSM implementation. The analysis includes assessment of significant factors, developing linear and quadratic estimation models of outputs, quantification of statistical fit, and identification of recommended input settings for desired performance.

2.2 Defining a System

At a basic level, a system can be defined in terms of configurable input parameters and measured outputs, as shown in Figure 2.1. A goal of experimental design is to develop estimation models of each output, as shown in (2.1). Each output is defined as y_i, where $i = \{1, 2, \cdots, n\}$ and $x_1, x_2, \cdots, x_k$ are the inputs. Typically, one might consider only the main input parameters when estimating system performance. However, experimental design uncovers terms in the model that incorporate higher order combinations of parameters.

$$y_i = f(x_1, x_2, \cdots, x_k) \tag{2.1}$$

2.3 2^k Factorial Design

A first step towards defining an elementary design is to set input parameter boundaries within low and high levels the operating range. The most basic experimental design is the full factorial design

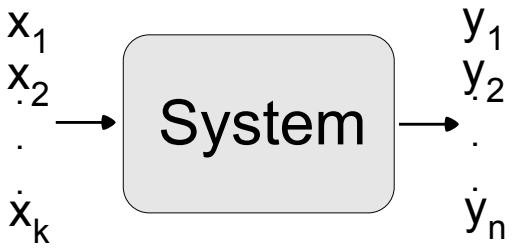


Figure 2.1: Basic system model

which considers all the possible combinations of low and high levels of the inputs. Consider a two input system. To test every combination, four distinct tests would be required. If the number of inputs is k, then there are 2^k tests in a single replicate of a factorial design. Figure 2.2 shows a two input system consisting of A and B. A common geometric convention for discussing designs is to denote each input by a capital Latin letter. The output associated with each combination of inputs is designated by corresponding lowercase letters in the following manner. The use of a high level of an input is designated by using the corresponding lowercase letter for the output value. Conversely, the use of a low value of the input is indicated by the absence of the matching letter. Therefore, a corresponds to the output when input A set to its high value, and B to its low value. Similarly, ab corresponds to both A and B set to high values. The convention for designating the output that corresponds to both inputs set to their low values is to use '(1)'. Table 2.1 lists the possible combinations of A and B and their corresponding notation [2].

Table 2.1: 2^2 factorial design

Input A	Input B	Combination	notation
-	-	A low, B low	(1)
+	-	A high, B low	a
-	+	A low, B high	b
+	+	A high, B high	ab

To determine the average effect of an input factor, one needs to determine the change in the output that is generated by a change in the level of that factor. If y is the output generated at any point,

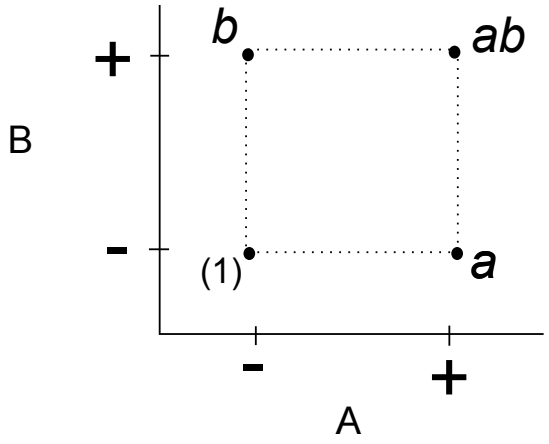


Figure 2.2: Two input factorial design

than $\bar{y}_{A^+}$ is the average of y when factor A is set to +, and $\bar{y}_{A^-}$ is the average of y when factor A is set to -. Therefore, the main effect of A can be written as $\bar{y}_{A^+} - \bar{y}_{A^-}$. If there are n replicates taken at each point, than $\bar{y}_{A^+}$ is given by the (2.2) and $\bar{y}_{A^-}$ is given by (2.3). The main effect of A is rewritten in (2.4). Similarly, the effect of B is given in (2.5). Finally, the **interaction** of AB is driven by the effect of A when B is high and the effect of A when B is low, as shown in (2.6).

$$\bar{y}_{A^+} = \frac{ab + a}{2n} \tag{2.2}$$

$$\bar{y}_{A^-} = \frac{b + (1)}{2n} \tag{2.3}$$

$$A = \frac{1}{2n}\left[ab + a - b - (1))\right] \tag{2.4}$$

$$B = \frac{1}{2n}\left[ab + b - a - (1))\right] \tag{2.5}$$

$$AB = \frac{1}{2n}\left[ab + (1) - a - b\right] \tag{2.6}$$

A concept underlying calculation of each factor's effect is the concept of **contrast**. The contrast of A is determined by differencing, the results when A is high with when A is low. Therefore the contrast of A is $ab + a - b - (1)$. Note that these contrasts are also identified by following the + and - signs on Table 2.1 underneath each input. The column associated with A has - next to (1), + next to a, - next to b, and + next to ab.

These same concepts generalize as the number of input variables increases. For example, a three input system requires eight design points which create the vertices of a cube, as shown in Figure

2.3. In any case, one can consider the space inside these vertices to be considered a search space of the variables where a factorial design tests the boundary conditions of the search space. Performing tests at the vertices of the design space can provide enough information to draw conclusions on parameter significance, identify interaction between parameters, and create an estimation model of performance. Another advantage of factorial designs is their efficiency compared to the traditional method of changing one-factor-at-a-time. Consider the two variable system of Figure 2.2. Informa-

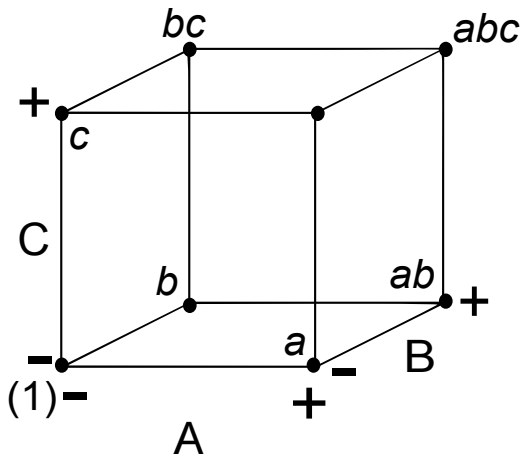


Figure 2.3: Test Points in 2^3 Factorial Design (based on [2] used under fair use, 2012)

tion on both variables could be obtained by changing the inputs one at a time while holding the other constant. Therefore, tests would be required at (-1,-1), (1,-1), and (-1,1). However, in order to account for some element of experimental error, at least two observations are required at each location. Consequently, a total of six observations are required. In contrast, if one performed one test at the four corners of the square, then there would be two observations available to identify the effect of A and B which provides a minimum accountability for experimental error. Therefore, the relative efficiency of factorial designs is (6 tests/ 4 tests)=1.5 compared to traditional approaches. Efficiency grows to 3.5 when there are six input factors. Furthermore, if the one-factor-at-a-time approach identified that tests at (-1,+1) and (1,-1) yielded a better output than the (-1,1) point, a conclusion might be that (1,1) would provide even better performance. However, if an interaction between the two inputs exists, this conclusion would be in error. Section 2.10 explores interactions in more depth.

In brief, factorial designs test all the combinations of the low and high values of the input pa-

rameters. These designs provide a more efficient approach to testing a system as compared to traditional one-factor-at-a-time-approaches. Subsequent analysis of the results, as discussed in Section 2.8, identifies the statistical significance of each input parameter as well as combinations of parameters. This also quantifies the interaction between parameters. Furthermore, the analysis yields a linear estimation model of each output as a function of the inputs. A drawback in the factorial design is the exponential rise in the number of tests compared to the number of inputs. The next section discusses fractional factorial designs which were developed to address this limitation.

2.4 Fractional Factorial Designs

The estimation models developed from factorial designs are functions of each main input parameter as well as every combination of inputs. In systems with a large number of inputs, an assumption commonly made, is that many higher-order interactions are negligible [2]. For example, the estimation model from a system consisting of 7 inputs of $A, B, C, \cdots, G$, would not have a significant effect from the $A*B*C*D*E*F*G$ term or many of the other higher-order interaction terms. In this case, a **fractional factorial** design can still identify factor significance and potential interactions related to the main inputs and lower-order interactions without resorting to the full factorial design. Fractional factorial designs significantly reduce the number of tests performed and can act as an initial screening for large parameter systems. Fractional factorial designs also provide the foundation for a unique type of experimental design process called Taguchi method, discussed in Section 2.6.

The two keys to defining fractional factorial designs include determining how to divide the full factorial and identifying aliases. Aliases are defined as contrasts that are equivalent. A common fractional approach is to perform 2^{k-p} tests. When $k = 3$ and $p = 1$, this is known as a one-half fractional design. In this case, the fractional factorial design will require four tests which was reduced from the full factorial run of eight tests. Other designs include the 1/4 fractional design.

Consider the 2^3 factorial design described in Figure 2.3. Table 2.2 details the design and the associated combinations of outputs. The 1/2 fractional design is constructed by breaking the original design into two parts determined by the factor ABC. Table 2.3 reorders the design and

splits it into two parts where the first four rows are associated with ABC set to + and the second four associated with ABC set to - [2]. The top and bottom halves are each considered a half-factorial design. The next step is to identify if any of the combinations can be removed because they provide redundant information. To determine if there are contrasts that are equivalent, one

Table 2.2: 2^3 factorial design (based on [2] used under fair use, 2012)

Run	A	B	C	Combination
1	-	-	-	(1)
2	+	-	-	a
3	-	+	-	b
4	+	+	-	ab
5	-	-	+	c
6	+	-	+	ac
7	-	+	+	bc
8	+	+	+	abc

Table 2.3: Breaking a 2^3 factorial design into 1/2 factorial designs (based on [2])

	Input Factor						
Combination	A	B	C	AB	AC	BC	ABC
a	+	-	-	-	-	+	+
b	-	+	-	-	+	-	+
c	-	-	+	+	-	-	+
abc	+	+	+	+	+	+	+
ab	+	+	-	+	-	-	-
ac	+	-	+	-	+	-	-
bc	-	+	+	-	-	+	-
(1)	-	-	-	+	+	+	-

can compare the pattern of the + and - in upper and lower parts of the original design. Considering just the upper half, the pattern of A matches that of BC. In other words, the contrast for A in the top half is $a - b - c + abc$ which is the same for BC. Similarly, B is equivalent to AC and C is equivalent to AB. The final fractional factorial design is created by eliminating the redundancy. Therefore, a half-factorial design would include the four data points a,b,c, abc as shown in Figure 2.4. A similar process performed on the lower half of the full factorial design results in an alternate design consisting of *(1)*, ab, ac, bc. This process has sufficiently reduced the original design of eight

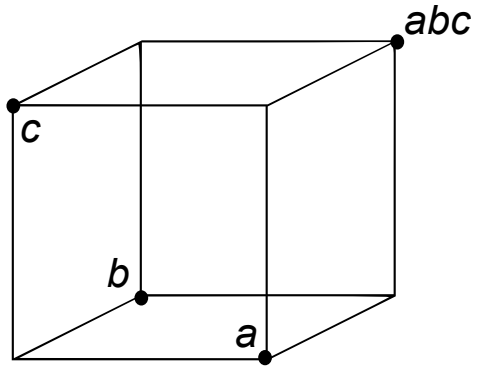


Figure 2.4: Half-factorial design (based on [2], used under fair use, 2012)

data points down to four.

Together, the fractional factorial and full factorial designs provide researchers an approach to first screening for parameter significance and then developing a linear estimation model of the output, as discussed Section 2.8.2. The limitation with the factorial-based model is that it cannot capture curvature in the output. However, in small regions of interest, the linear models provide information to identify direction of improved performance. Methods such as steepest ascent/descent identify how one can direct a search of parameters towards the location of the probable optimum. Experimental methods often follow a search pattern where the results of one design lead the experimenter to the next until the desired optimum is found. At some point, the researcher will most likely desire a better model to estimate performance. In order to identify a better model, more sophisticated experimental designs are required. The trade-off for improved modeling capabilities is a higher cost in terms of the number of required tests.

2.5 Response Surface Methodology

While the basic models developed with factorial designs provide a general indication of output trends, their shortcomings motivated the development of more powerful designs. The designs, known as second-order RSM were developed to identify quadratic models capable of capturing curvature in an output. Factorial designs are considered a first-order RSM design, however when RSM is discussed in literature it almost always refers to second-order designs. This section discusses

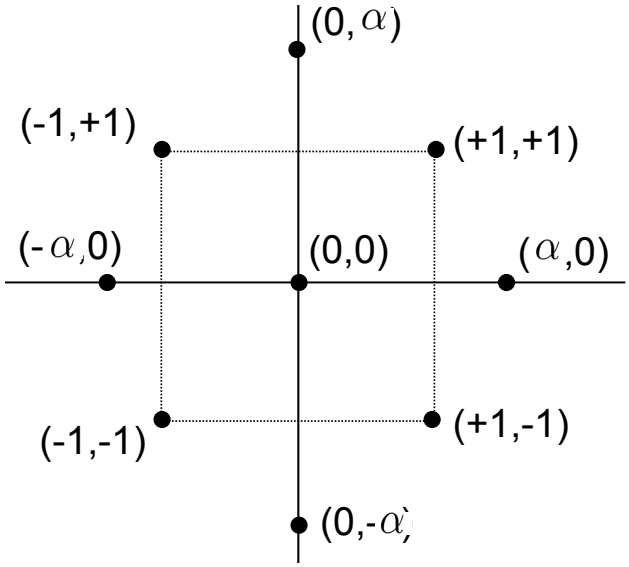


Figure 2.5: Two variable CCD (based on [2] used under fair use, 2012)

the central composite design (CCD) and Box-Behnken RSM designs.

2.5.1 Central Composite Design

As discussed earlier, the factorial design provides a starting point for most experimenters. A common next step is to augment the factorial design with additional data points along the center axis. The resulting design is called a CCD. Two elements are required to define a CCD design. This includes the distance away from the center point along the axis, commonly known as α, and the number of center points. Typically, three to five center points are included to provide sufficient variance in the experimental error [2]. Figure 2.5 illustrates a two-variable CCD and Figure 2.6 shows a three-variable CCD where $\alpha = 1$, also known as a face-centered CCD. It may seem non-intuitive for α to be greater than 1. Consider the boundaries [-1,+1] to be *operational* ranges in which the test was implemented and not the overall physical limit on the system. This perspective enables settings that are outside the operational range but within the physical limits of the system.

A key goal of second-order designs is for the estimation model to provide stable and consistent variance of the predicted response at any point x. The variance of a predicted response at a point x is defined in (2.7), where σ^2 is the variance of the residual between the actual observed values and

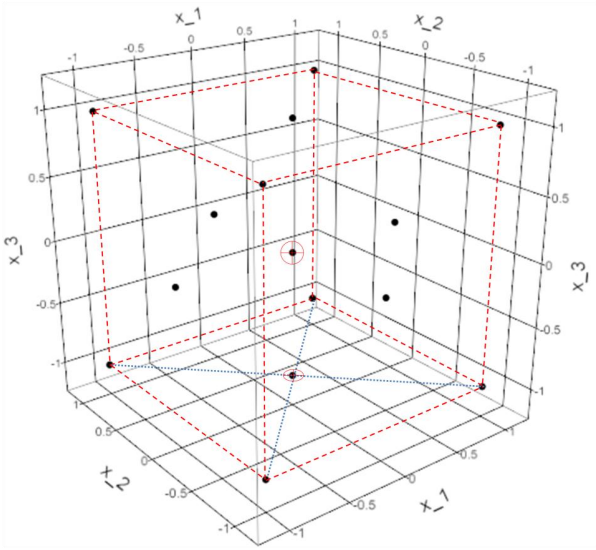


Figure 2.6: Face-centered CCD

the predicted values [2]. If the variance is the constant for all points x that are the same distance away from the center, then the design is know as *rotatable*. This feature enables the design to provide equal precision of the estimation in all directions.

$$V[y(\hat{x})] = \sigma^2 \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x} \tag{2.7}$$

The selection of α determines if a particular CCD design rotatable. A fully rotatable CCD design is created when $\alpha = (n_f)^{1/4}$ where n_f is the number of points used in the factorial part of the design. A face-centered CCD, where $\alpha = 1$ is typically more common due to the ease in creating the design and setting the input parameters. The face-centered design is considered near-rotatable by definition.

CCD Variations In addition to rotatability, there are two other design aspects of CCD that lead to variations of the basic design. The first is blocking. There may exist nuisance factors which are not considered one of the controllable inputs in the design. For example, if experiments are performed on separate days or in multiple batches, it would be desirable to perform these tests in similar blocks. The CCD-orthogonal blocking (CCD-OB) design separates analysis of the factorial points from the axial points. The factorial points are analyzed together with a portion of the center points, while the axial elements are analyzed with the remaining center points. Therefore, if only

a few data points are to be collected in one session, than it is preferable to perform the factorial points in one session and the axial points in another.

The second design aspect to consider is the number of additional center points included in the design. Typically, at least three to five additional center points are tested along with the rest of the design. These replicates of the design provide necessary data points for estimating the experimental error and are helpful in quantifying the fit of the model. A uniform precision (UP) design selects enough additional center points to ensure that the prediction variance at the center is similar to the prediction variance at the vertices [19].

The relative disadvantages of these variations, compared to the basic CCD, is the increase in data points. For example, a three input CCD with three center points requires 17 tests, and CCD-OB or CCD-UP with six additional center points both require 26 tests.

2.5.2 Box-Behnken Design

Another common design is the Box-Behnken design, as shown in Figure 2.7. Table 2.4 contrasts the Box-Behnken design with the face-centered CCD for a three-variable system. The Box-Behnken design falls underneath a subcategory of spherical designs where all test points fall within a sphere of radius $\sqrt{2}$. The design differs from CCD in the lack of data points at the extremes of the factors. In other words, there are no points at the vertices of the cube. This is desirable if there are extreme factor settings which can be difficult to set or are unrealistic in a live application. However, it is know for providing better estimation capabilities near the center than at the extreme settings of the input parameters. Additionally, the Box-Behnken design is more sensitive to missing data. The CCD requires more tests initially, however this makes them more tolerant of missing runs.

One must consider several factors when selecting between CCD or Box-Behnken design, as summarized in Table 2.5. In general, Box-Behnken provides higher accuracy for a model around the center point given the spherical nature of the point selection around the center. It uses less overall data points than the CCD, therefore performance degrades more when data points are lost. In the case where it is physically difficult or undesired to set system parameters to their extremes, then Box-Behnken may be desirable. However, because of this it provides less modeling accuracy

Table 2.4: Box-Behnken and CCD Design

	Box-Behnken			CCD		
Test	x_1	x_2	x_3	x_1	x_2	x_3
1	-1	-1	0	-1	-1	-1
2	-1	1	0	-1	-1	1
3	1	-1	0	-1	1	-1
4	1	1	0	-1	1	1
5	0	-1	-1	1	-1	-1
6	0	-1	1	1	-1	1
7	0	1	-1	1	1	-1
8	0	1	1	1	1	1
9	-1	0	-1	-1	0	0
10	1	0	-1	1	0	0
11	-1	0	1	0	-1	0
12	1	0	1	0	1	0
13	0	0	0	0	0	-1
14				0	0	1
15				0	0	0

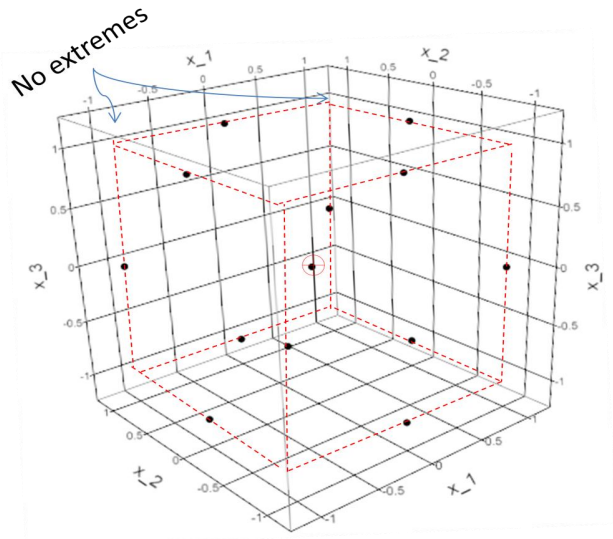


Figure 2.7: Box-Behnken Design

at the edges of performances. CCD is generally easier to implement if one has already performed a factorial design, as the design uses factorial as its foundation. If one uses a face-centered design, then only three levels of the inputs are needed, otherwise five will be needed which can cause more difficulty in setup. Because there are more data points overall, it is more robust to missing data.

Table 2.5: Comparison Between CCD and Box-Behnken

Design	Advantage	Disadvantage
CCD	Robust to missing data Good prediction at edges Stepping stone from factorial	Requires more tests May require more levels
Box-Behnken	Strong prediction at a center Requires 3 levels less tests	Weak at edges Sensitive to lost data

In summary, fractional factorial designs are capable of identifying parameter significance, however are limited in their capability to develop models of system performance. Factorial designs require more tests than fractional designs yet can develop a linear model of system performance. Next, RSM models improve on the modeling capability of experimental designs, yet also incur a higher cost in terms of number of tests.

There is an alternate approach, known as the Taguchi method, that draws upon advantages of the previously described designs. Taguchi uses a type of fractional factorial design in order to reduce the number of tests. Rather than solving a model or set of models for a recommended solution, Taguchi limits its solutions to a discrete set of inputs associated with the original design. In this manner, Taguchi can significantly reduce the number of tests required, and still identify a solution.

2.6 Taguchi Methods

The Taguchi methods are considered a contribution to the field of experimental design for their innovative fractional factorial arrays and unique perspective on performance compared to the traditional focus solely on minimization or maximization of a output. Taguchi was a process engineer, and his experience trained him to emphasize variation around a target as much as optimizing an output. This focus led to using experimental design to develop processes and products that were more insensitive to variation in the environment. This section will describe the major differences in between Taguchi and traditional factorial-based designs, selection of the Taguchi design arrays, the concept of Taguchi signal-to-noise as a performance metric, and relative advantages and disadvantages.

One of the most obvious differences between the Taguchi approach and traditional methods is the use of discrete value inputs. Both factorial and RSM approaches consider input values on the familiar [-1,+1] continuum between a low and a high value. The models developed by factorial and RSM are continuous and therefore recommended solutions fall in between the tested design points. In contrast, Taguchi only tests a certain subset of discrete points and the identified solutions are in terms of discrete values. The use of a discrete inputs is reflected in the different notation used in Taguchi designs. The Taguchi designs first need to define how many discrete levels of parameter will be tested. Rather than use -1 to represent the low level and +1 to represent the high level, Taguchi arrays use "1", "2", "3", etc. For example, if a design considers 4 discrete levels then a "1" in the Taguchi array corresponds to the -1 or - in a traditional design and "4" corresponds to +1 or +.

2.6.1 Selection of Taguchi design array

The Taguchi design arrays take advantage of a type of fractional factorial design, called orthogonal arrays (OA) [20] to identify designs that minimize the testing cost compared to factorial designs. Consider a system with k configuration parameters and l levels, and N total test cases. A traditional full factorial design requires $N = l^k$ unique configurations to test each combination of parameters and levels. For example, a system with $k = 4$ and $l = 3$ would require $3^4 = 81$ test cases for a

full factorial design. An equivalent Taguchi array requires 9 tests, as shown in Table 2.7. Taguchi recognized the need for reduced testing matrices given the realistic constraints on time, personnel, and resources.

Orthogonal Arrays

This section describes the OA testing matrix that provides an efficient combination of configuration parameters to minimize the testing resource cost. These arrays are based on fractional factorials discussed in Section 2.4. The formal notation is based on the literature and is somewhat non-intuitive. The array is defined as $OA(N, k, l, t)$ where an array of size $N \times k$ is created from m parameters consisting of l levels which are a subset of S and strength t such that $(0 \leq t \leq k)$ [21]. The selection of strength is driven by the potential for interaction amongst factors. Pairwise interaction between any two parameters is adequately address with a strength of $t = 2$. One can increase t to incorporate higher order interactions, at the cost of more tests. Strength of $t = 2$ is adequate for this application.

The informal notation of LN is often used for convenience over the formal notation. For example, Table 2.6 shows the OA(9,4,3,2) or L9 array. There are 9 total tests, 4 input parameters, and 3 discrete levels of each parameter. The elements of the array are selected from $S = \{1, 2, 3\}$ representing three discrete levels of input parameters which are mapped to the labels:A, B, C, D. Any two columns, as defined by $t = 2$ will always show nine possible combinations in the rows: (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), and (3,3). Each row represents a unique test case where the control parameters are set to the designated level. For example, test ID 7 of the OA(9,4,3,2) sets parameter A to level 3, parameter B to level 1, C to level 3, and D to level 2.

There exist formalized OA tables to assist in the development of an experimental design. Table 2.7 lists suggested designs based on the number of input parameters and discrete levels. Taguchi methods are also known for their capability to investigate the output created from combinations of controllable parameters and uncontrollable parameters. The OAs described above are also known as *inner arrays*. A second array can also be incorporated around each point of the inner array that includes variations of uncontrollable parameters. For the purposes of testing, these typically uncontrollable parameters are fixed at minimum and maximum values of their uncontrollable pa-

Table 2.6: OA(9,4,3,2) L9 Orthogonal Array

Test ID	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table 2.7: Selecting a Taguchi design based on number of parameters and discrete levels

	Number of Parameters							
Levels	2	3	4	5	6	7	8	9
2	L4	L4	L8	L8	L8	L8	L12	L12
3	L9	L9	L9	L18	L18	L18	L18	L27
4	L16	L16	L16	L16	L32	L32	L32	L32
5	L25	L25	L25	L25	L25	L50	L50	L50

rameters' range. This is often called the outer array. This dissertation only considers inner array test cases.

Taguchi Signal to Noise Ratio

Another key area in which Taguchi differs from traditional experimental design methods is the focus on the success of the output. Traditional methods strive to identify and maximize/minimize the mean output of a system. Taguchi's production background trained him to emphasize not only mean of an output but also to minimize the variation around the mean, which he believed to be a truer measure of overall quality. Taguchi refers to this relationship between variation and the output as signal to noise ratio (SNR). To avoid confusion with common wireless communications terminology, this dissertation will refer to this as SNR_{Tag}. This concept lends itself well to communications systems problems given the goal of minimizing uncertainty and sensitivity to environmental changes in order to ensure quality of service (QoS). There is a tendency to increase power to improve link

margins. If the uncertainty is minimized then the link margin can also be minimized.

There are three distinct formulas for calculation of SNR_{Tag}: 1) lower-is-better (LIB) as shown in 2.8, 2) higher-is-better (HIB) as shown in 2.9, and 3) nominal-is-better (NIB) as shown in 2.10. Each sample of the metric under consideration is denoted as y_i, and y is the overall sample mean given from n replicates. The variance of the sample is denoted by σ^2.

$$SNR_{Tag} = -10\log\left(\sum_{i=1}^{n}\frac{y_i^2}{n}\right) \tag{2.8}$$

$$SNR_{Tag} = -10\log\left[\sum_{i=1}^{n}\left(\frac{1}{y_i^2}\right)/n\right] \tag{2.9}$$

$$SNR_{Tag} = -10\log\left[\sum_{i=1}^{n}\left(\frac{y_i^2}{\sigma^2}\right)\right] \tag{2.10}$$

Implementation of a Taguchi design proceeds similarly to traditional experimental design. A desired target or the overall goal of HIB or LIB must be identified. Then each test case in the OA is implemented on the system and the corresponding output. Multiple runs of each test case are recommended and averaged. The mean and SNR_{Tag} are calculated. The mean and SNR_{Tag} represent two separate outputs. Hence, further analysis is similar to the RSM method where multivariate optimization is performed on several outputs. The statistical package, SAS JMP 9.0, utilized desirability functions for analyzing Taguchi arrays. The final recommended parameters strive to optimize both the mean of the output and the SNR_{Tag}. Given the small number of specific tests, it is likely that the recommended solution was not one of the original tests. In this case, a **confirmation test** is implemented where the specific solution is implemented as an actual test across the system.

2.6.2 Limitations of the Taguchi Method

There are several drawbacks and limitations when following the basic Taguchi methodology. First, each parameter value requires discrete values. Estimated results are only calculated at these discrete values. In contrast, the RSM methodology enables prediction at points not located at dedicated values of the control parameters. Second, the method discussed assumes that interaction effects between parameters are negligible. A complete factorial screening that evaluates every combination

of parameter values has the capability of identifying interaction effects amongst parameters. In many cases this is not feasible given the sheer quantity of test cases required. Often, experimenter knowledge of the overall systems enables intuitive judgments about which effects lend themselves towards interaction.

The Taguchi method predicts a parameter setting that will result in the best performance based on the initial definition. It is unable to define a model for the system behavior. RSM techniques are suggested for this level of system understanding; however, the cost in terms of increased testing cases can be significant. The analysis process for identifying a favorable parameter set is performed offline and typically is not conducive to an automated process for real-time operations. Therefore, the Taguchi method alone might not provide a real-time solution for CE operations. There are several hybrids of the Taguchi method with heuristic methods, such as a GA, which might provide more real-time functionality. Finally, the resulting solution set of configuration parameters typically requires a confirmation test in which these parameters are run through the testing configuration.

2.7 General Comparisons of Designs

Several factors must be considered when comparing between the presented experimental designs. These include the goals of the analysis and the tolerance for testing. The goals of testing need to indicate whether a model is required and if so, what level of accuracy is desired. Further goals might include identifying interactions between inputs and determining parameter significance. Each of these must be considered in relation to the required number of tests and replicates needed to perform a particular design. Figure 2.8 generally compares designs based on their capabilities for identifying knowledge of the system performance compared to their testing cost. Additionally, Figure 2.9 illustrates how the number of tests increases dramatically as the number of inputs under consideration grows.

Taguchi designs are a special consideration. The drawbacks presented in Section 2.6.2 discuss how the selection of inputs used in the Taguchi fractional factorial based array are assumed to have little to no interaction. This requires some knowledge of the system performance, and may require running one of the other designs, such as a factorial. In addition, the Taguchi requires defining

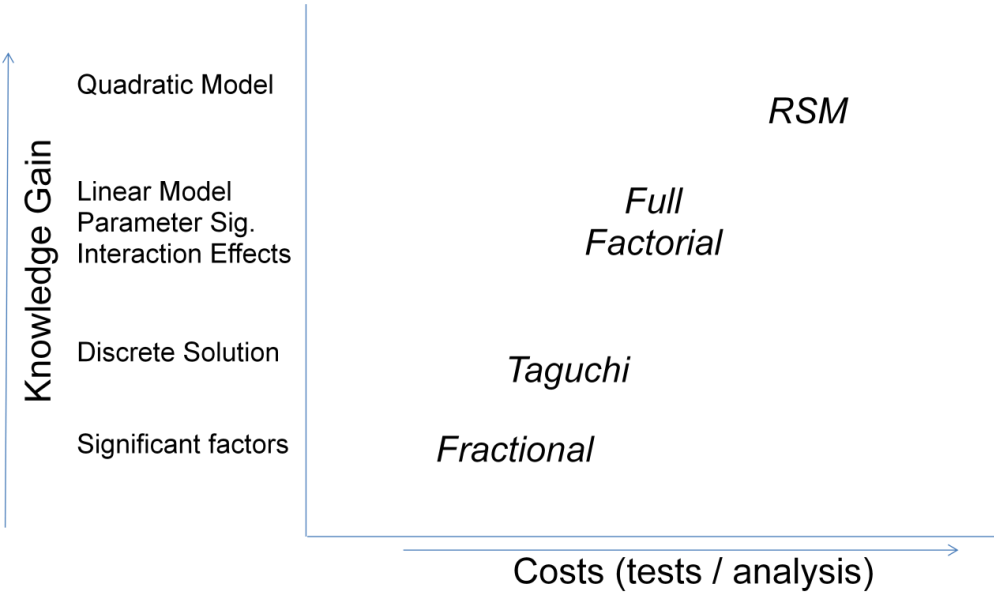


Figure 2.8: Comparison of designs considering testing cost and knowledge gain

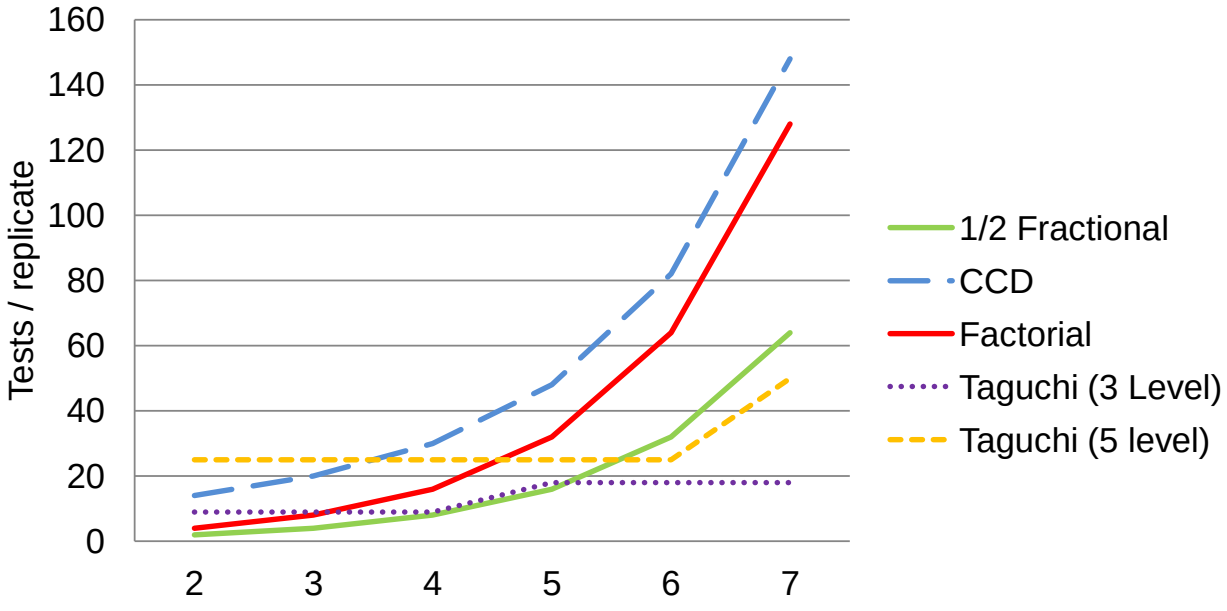


Figure 2.9: Number of tests required per input parameters

the discrete number of inputs that are available. The lower the number, the more limiting the suggested solutions will be.

2.8 Analysis of Experimental Designs

The benefits that these experimental designs afford include quantification of input parameter significance, identification of interactions between parameters, and development of estimation models of outputs. This section details several of the analysis steps required to acquire this information. Techniques discussed include analysis of variance and regression modeling. A reference design example is presented to illustrate analysis of a factorial and CCD design. A qualitative approach to identifying regions of desired performance is presented in addition to a quantitative multi-variate method.

2.8.1 Analyzing a Factorial Design for Input Parameter Significance

The techniques of analysis of variance (ANOVA) provide a mechanism to identify the statistical significance of input parameters. This section provides a brief overview of ANOVA. Refer to [2, 22] for more details. Each main input and combinations of inputs are analyzed to determine the effect that each input has on the output. ANOVA identifies the most important inputs with regards to affecting the output. It also identifies interactions between inputs that would be impossible to uncover when varying only one factor at a time. An ANOVA procedure seeks to partition the variation in the output into components that identify the sources of variation. The procedure breaks the total variation in the output into the portion due to random error and the portion due to changes in each input as well as combinations of inputs. This is important in order to simplify subsequent estimation models. Input deemed insignificant can be dropped from models without adversely affecting the estimation.

ANOVA is essentially a hypothesis test of H_0: input effects are 0 versus H_1: at least one of the input effects is not 0. ANOVA quantifies whether the input effects on the output are zero or not. A common way of reporting the results of hypothesis test is to state whether the null hypothesis was accepted or rejected at a specific level of significance, also known as an α-value. Typically α

is set to 0.05. This approach is limiting because it provides no additional quantification on why the hypothesis was rejected. There are two types of error to be cognizant of. Type I errors are related to the level of significance and when the null hypothesis is rejected when it is true. Type II errors occur when the null hypothesis is not rejected when it is actually false. The experimenters tolerance for type II errors plays a part in determining the minimum number of replicates of a design.

The ***p*-value** approach is a widely adapted technique to convey more information about the weight of the evidence against the null hypothesis. The p-value is the smallest level of significance that would lead to a rejection of the null hypothesis H_0 [2]. Therefore the p-value is the smallest level of α at which the data are significant. Statistical software packages or spreadsheets are typically used to calculate p-values using pre-defined t distributions.

An example data set is shown in Table 2.8, where there are a different tests, also known as treatments, and n replicates observed for each. The average of each row is calculated and the overall mean of all the row averages is considered a grand mean [4]. The following description will utilize "dot" notation which implies summation over the subscript that the (.) replaces.

Table 2.8: Example form of a data set (based on [4] used under fair use, 2012)

	Replicate				
Test Run	1	2	$\cdots$	n	Average
1	y_{11}	y_{12}	$\cdots$	y_{1n}	$\bar{y}_{1.}$
2	y_{21}	y_{22}	$\cdots$	y_{2n}	$\bar{y}_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
a	y_{a1}	y_{a2}	$\cdots$	y_{an}	$\bar{y}_{a.}$
				grand Mean	$\bar{y}_{..}$

To illustrate how ANOVA is performed, start with a basic model of the output shown in (2.11) where y_{ij} is the i'th observation ($i = 1, 3, \cdots, a$) of the j'th input ($j = 1, 2, \cdots, n$); μ is an effect common to the whole experiment; τ_i is the effect of the i'th input; and ε_{ij} is the random error seen in the ith input and the j'th observation. A basic assumption is that errors are normal and independently distributed with a mean of zero. If one considers μ_i to be the mean of the ith population, then $\tau_i = \mu_i - \mu$ making μ a fixed parameter.

$$y_{ij} = \mu + \tau_j + \varepsilon_{ij} \tag{2.11}$$

An ANOVA test consists of comparing the two hypotheses:

- $H_0 : \tau_1 = \tau_2 = \cdots \tau_k = 0$
- $H_1 : \tau_i \neq 0$ for at least one i

The ANOVA seeks to identify the variance that is caused by the change in the inputs compared to the variance that is caused by error. One way of looking for this is to evaluate the sum of squares of the inputs and the sum of squares of the error. Consider (2.12) that breaks down the variance into the variation within a cell and the variation between cells. This is equivalent to 2.13.

$$\sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \overline{y}_{..})^2 = n\sum_{i=1}^{a}(y_{i.} - \overline{y}_{..})^2 + \sum_{i=1}^{a}\sum_{j=1}^{n}(y_{ij} - \overline{y}_{.i})^2 \tag{2.12}$$

$$SS_T = SS_{Treatments} + SS_E \tag{2.13}$$

Two more key values needed for the ANOVA are the *error mean square* shown in (2.14) and the *Treatment mean square* shown in (2.15).

$$MS_{error} = \frac{SS_{error}}{N - a} \tag{2.14}$$

$$MS_{treatment} = \frac{SS_{Treatment}}{a - 1} \tag{2.15}$$

Finally, the most common reported statistic is the F-ratio which is defined in (2.16)

$$F = \frac{MS_{Treatment}}{MS_{error}} \tag{2.16}$$

A large F-ratio implies that the variation caused by the model is larger than variation caused by the residual error. When a high F-ratio is associated with a p-value of less than 0.05, then the model is said to be statistically significant at the $\alpha = 0.05$ level. The results of the ANOVA provide a quantification of input parameter significance. This enables one to also identify insignificant factors. These insignificant factors can effectively be dropped from the future estimation model as their effect has been deemed statistically insignificant. This process is known as a **screening**, and

often one of the first steps in a statistical analysis. The factorial approach has drawbacks as the number of input parameters increase. The corresponding increase in tests makes implementation unrealistic. However, a screening test early on can save costs in future testing by identifying insignificant parameters early on. Fractional factorial designs, discussed in Section 2.4, provide an alternative to full factorial designs for screenings at reduced testing costs.

2.8.2 Developing an Estimation Model

Experimental designs are capable of identifying estimation models of output parameters. The method of least squares provides an estimation of the regression coefficients of models such that the sum of the square of the errors is minimized. The models are typically written into matrix format shown in (2.17).

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.17}$$

Where $\mathbf{y}$ is a vector of the outputs, $\mathbf{X}$ is a matrix of the levels of the independent variables, $\boldsymbol{\beta}$ is the vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is the vector of the experimental errors as show in (2.18) [2].

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{2.18}$$

The least squares estimator is shown in (2.19) where $\hat{\boldsymbol{\beta}}$ is the estimate of the coefficients. This method requires that the errors are normally distributed with a mean of 0. Justification for this assumption is discussed in Section 2.8.3. When residual errors are normally distributed with a mean of zero, the error term drops out because the expected value of a normal random variable with a mean of 0 is 0. Finally, the estimate of each output, shown in (2.20), is a product of the estimated β coefficients and the inputs. The estimated model of the output is given by (2.21). The factorial design is often called an orthogonal design because the off-diagonal elements of the $\mathbf{X}'\mathbf{X}$ matrix are zero.

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} \tag{2.19}$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{2.20}$$

$$\hat{y_i} = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_{ij} \quad i = 1, 2...n \tag{2.21}$$

RSM designs lead to models in the form of (2.22) where k is the number of input parameters, x_i is a main input factor, β_0 is the intercept, and ε is the statistical error.

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum \sum_{i<j} \beta_{ij} x_i x_j \tag{2.22}$$

2.8.3 Justifying Assumptions of Residuals

It is a common when implementing design of experiments to assume that the residual error between observed values and estimation models is distributed normally with a zero mean [23, 2, 24]. Similar assumptions are made in autoregressivemoving-average (ARMA) time series models. Several approaches exist for verifying that this assumption is valid. Typically, the residual error is calculated and verified against a normal distribution. Goodness of fit metrics, such as the Shapiro-Wilk metric [25], are used to quantify how well the residual distribution matches a normal distribution. Similarly, graphical analysis in the form of normal quantile plots quickly indicate how well the residual error fits a normal distribution [26, 27].

Normal quantile plots, also known as normal probability plots, compares the quantiles (or percentiles) of one distribution against another. In this case, the quantiles of the observed distribution are plotted on the Y axis and the quantiles of a theoretical normal distribution are plotted on the X axis. Therefore, the line created from $X = Y$, indicates a perfect normal distribution. The closer the observed distribution is to a normal distribution, the closer the data points will fall in relation to this line.

Figure 2.10 shows the normal quantile plot of the residual error for packet error rate generated from the implementation of a Box Behnken response surface methodology statistical design. The x-axis shows two scales, the empirical cumulative probability for each value and the normal quantile values. The probability values are defined by (2.23) where r_i is the rank of the i'th observation, and N is the number of non-missing observations. The quantile values are identified from (2.24) where Φ is the cumulative probability distribution function for the normal distribution. The solid

red line represents where the observed data points would fall if the residual error was normally distributed. The slope of this line also indicates the standard deviation of the distribution. The horizontal dotted line indicates the mean of the distribution. In this case, the mean is verified as 0. Confidence intervals are indicated by dotted lines and are based on the Lilliefors test for normality [28]. This test is similar to the Kolmogorov-Smirnov test, however the specific mean and variance of the distribution is not needed. All of the data points fall within these confidence intervals which indicates that the assumption of normality is verified.

$$\frac{r_i}{N+1} \tag{2.23}$$

$$\Phi^{-1}\left(\frac{r_i}{N+1}\right) \tag{2.24}$$

In addition to visual checks like the normal quantile plots, tests such as the Shaprio-Wilk metric can quantify how well the residual error fits a normal distribution. The Shapiro-Wilk test defines the W statistic such that small values indicate a deviation from a normal distribution [29]. A $p-value$ is typically presented in conjunction with the W statistic such that small values reject the null hypothesis that the data is from a normal distribution. The W statistic is calculated in Equation (2.25) where x is a random sample of $x_1, x_2, \ldots x_n$, $x_{(i)}$ is the ordered sample values, and a_i are constants generated from the means, variances and covariances [29].

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2} \tag{2.25}$$

Table 2.9 lists W metric as well as the associated $p-value$ for several output metrics. The residuals were identified from the implementation of a Box Behnken response surface design. The W metrics are all high indicating strong goodness of fit combined with $p-values$ that are higher than 0.05. Based on the normal quantile plots and the goodness of fit analysis it is justified to assume that residuals generally meet the assumption of being normally distributed.

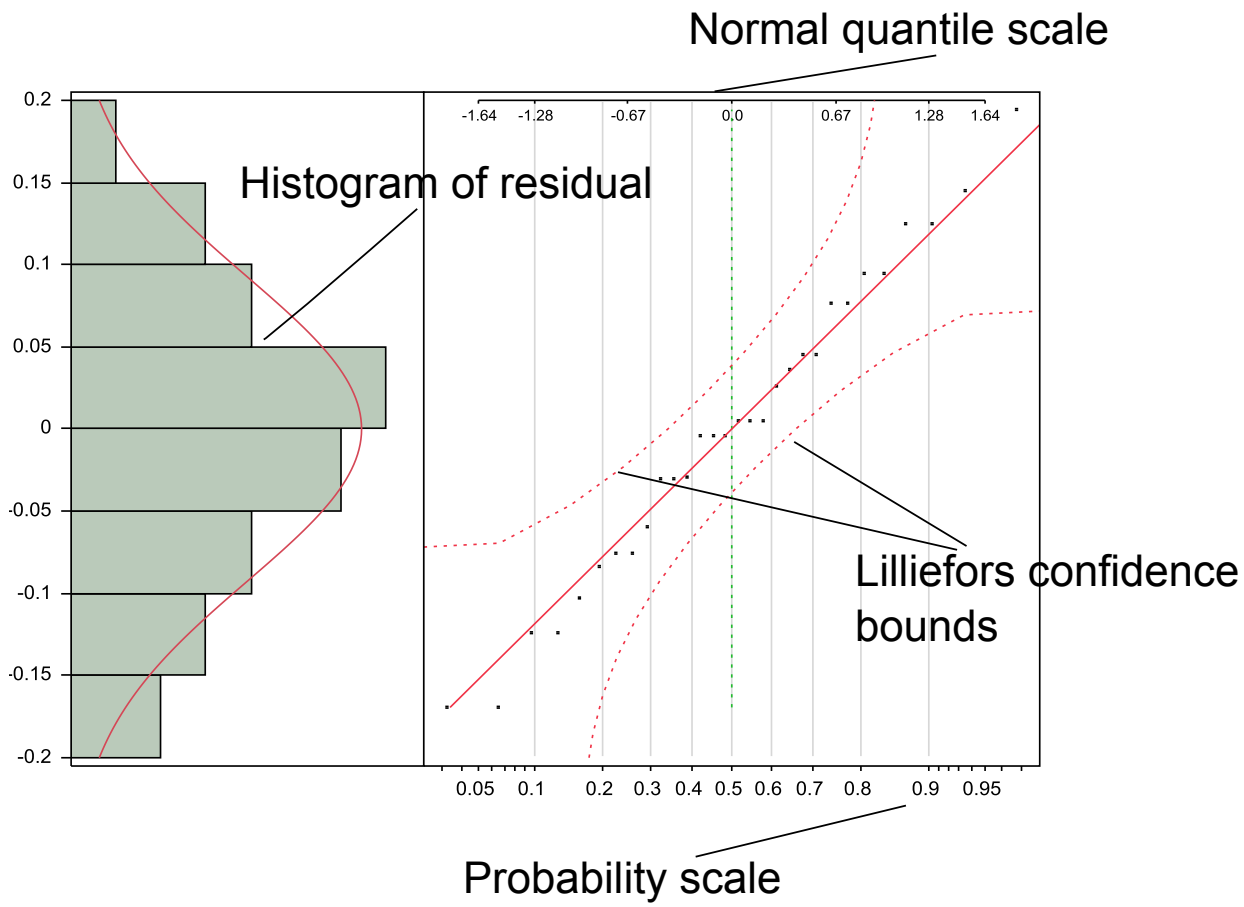


Figure 2.10: Residual error distribution and normal quantile plot for PER generated from a Box Behnken design

Table 2.9: Shapiro-Wilks goodness of fit test results for Box Behnken response surface design

	W	*p-value*
PER	0.984	0.911
Spectral Efficiency	0.950	0.171
Throughput	0.950	0.171
SNR	0.959	0.285
Goodput	0.950	0.167

2.9 Reference design example

This section presents a reference design example and the resulting ANOVA and model development analysis. The reference example used to explain concepts comes from a chemical engineering process. The goal of the experiment was to identify ideal operating conditions that maximized the yield and viscosity, and minimized molecular weight of a chemical process [2]. The two controllable inputs to the process were reaction time (seconds) and temperature (°F). The chemical process was performed in a single location. The full data set for this experiment is contained in Table 2.10. This data will be used to illustrate factorial and response surface methods implementation.

Table 2.10: Experimental data for reference example (adapted from [2] used under fair use, 2012)

Experiment No.	Reaction Time	Temperature	Yield	Viscosity	Molecular Weight
1	80	170	76.5	62	2940
2	80	180	77	60	3470
3	90	170	78	66	3680
4	90	180	79.5	59	3890
5	85	175	79.9	72	3480
6	85	175	80.3	69	3200
7	85	175	80	68	3410
8	85	175	79.7	70	3290
9	90	175	79.8	71	3500
10	80	175	78.4	68	3360
11	85	175	75.6	71	3020
12	85	180	78.5	58	3630
13	85	170	77	57	3150
14	80	170	76	61	2943
15	80	180	77.3	59	3466
16	90	170	78.5	67	3677
17	90	180	79	60	3888

2.9.1 Normalizing the inputs

Implementing an experimental design requires defining the input ranges based on low and high values that they can take. This enables a normalization on a [-1,1] scale where -1 corresponds to the low value and +1 corresponds to the high value. A nominal value is normalized to 0. This process, commonly known as coding in the experimental design field, is performed for several reasons. First,

it removes units of measure from the inputs. This also simplifies defining the experimental design tables by denoting input levels as - or + or -1 and +1. Furthermore, estimation models that are based on the normalized input values provide a more intuitive understanding of how the output is affected by each input. A model with coefficients in the natural scale of the input make it difficult to understand the true impact that each input might have. However, when all inputs are coded onto the [-1,+1] scale, the relative impact of any input is apparent.

Consider an input parameter ζ with a low value of ζ_{low} and a high value of ζ_{high}. The normalized value, x of the parameter is given by (2.26)

$$x = \frac{\zeta - \frac{\zeta_{low}+\zeta_{high}}{2}}{\frac{\zeta_{high}-\zeta_{low}}{2}} \tag{2.26}$$

In the reference example, consider the time input to be ζ_1 and the temperature to be ζ_2. The low for ζ_1 was 80 seconds and the high was 90 seconds. Similarly, the low for ζ_2 was 170 °F and the high was 180 °F. Table 2.11 shows the result of the normalization process.

Table 2.11: Reference design input parameters on normalized scale

Experiment No.	x_1	x_2
1	-1	-1
2	-1	1
3	1	-1
4	1	1
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	1	0
11	-1	0
12	0	1
13	0	-1
14	-1	-1
15	-1	1
16	1	-1
17	1	1

Upon inspection of the set of inputs used in the experiment, it is apparent that the first four and last four tests incorporated every combination of the low and high values of the two inputs. This

is indicative of the basic factorial design. The other tests incorporate nominal values which will be discussed in Section 2.5.

The estimation model for this system is in the form of (2.27), where x_1 is time and x_2 is temperature. The three outputs, y_1, y_2, and y_3 are yield, viscosity, and molecular weight. Table 2.12 lists the results of the ANOVA and Table 2.13 list the estimates of the parameter coefficients and their statistical significance.

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 \tag{2.27}$$

Table 2.12: ANOVA results of reference example

Output	F-ratio	prob>F
Yield	32.4286	0.0029
Viscosity	43.667	0.0016
M. Weight	69613.65	$< .0001$

Table 2.13: Input parameter coefficient estimates and significance

	Yield (y_1)		Viscosity ((y_2))		M. Weight (y_3)	
Term	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
Intercept (β_0)	77.5	$< .0001$	61.75	$< .0001$	3494.25	$< .0001$
Time (β_1)	1.025	0.0009	1.25	0.0075	289.5	$< .0001$
Temperature (β_2)	0.475	0.0143	-2.25	0.0008	184.25	$< .0001$
Time*Temperature (β_3)	0.025	0.8379	-1.25	0.0075	-79	$< .0001$

Table 2.12 indicates that the F-values indicate that the models for all outputs are considered statistically significant. The two main inputs, time and temperature, are significant parameters for all outputs, and the coefficient related to the combination of the two is also significant to the outputs of viscosity and molecular weight. However, the combination of Time*Temperature is not significant to yield. In the estimation model, the product of Time*Temperature represents the interaction between the two inputs. If this parameter was statistically significant, than the value of it would be much higher, indicating that any increase or decrease in this product would have a significant impact on the output.

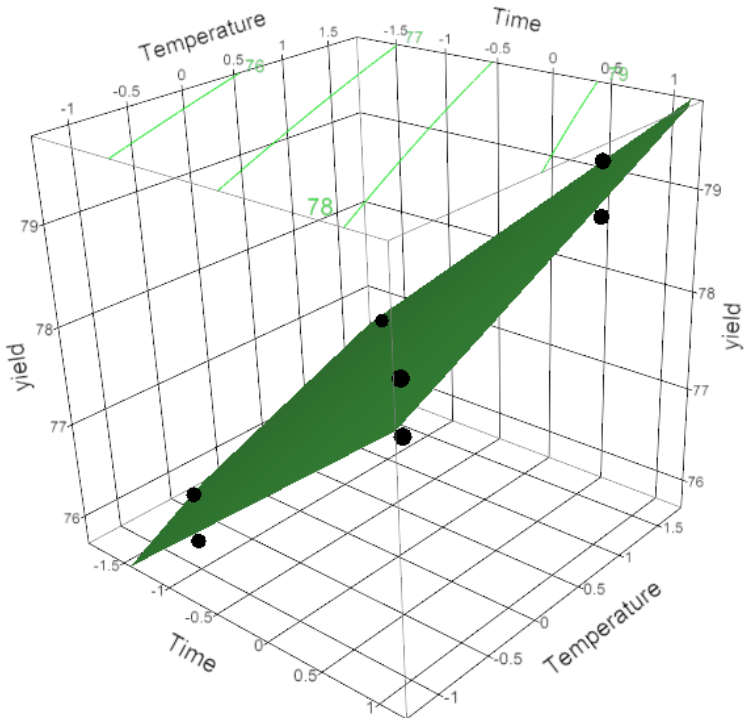


Figure 2.11: Model for yield from factorial example dataset

This means that it could be dropped from the model without adversely affecting the estimation. The estimates of the model based on the listed coefficients provide enough information to plot a linear representation of the process. Figures 2.11, 2.12, and 2.13 show the planes created by the models. The measured data points are shown in black dots. Note that the yield plot is a perfect plane, while the other responses show a slight twisting in the plane. The twisting comes from the effect caused by the second order term of Time*Temperature. Contour lines of the output are reflected onto the top surface of the plot. With an estimation model and the original data points, one can quantify how well the model fits the data.

2.9.2 Measuring the statistical fit of the model

In experimental design, the most common measure of statistical fit is the R^2 measure of fit. It is based on the residuals which are defined as the difference between an observation and the corresponding estimated value. This statistic is also known as *R-Square* or the *Coefficient of Determi-*

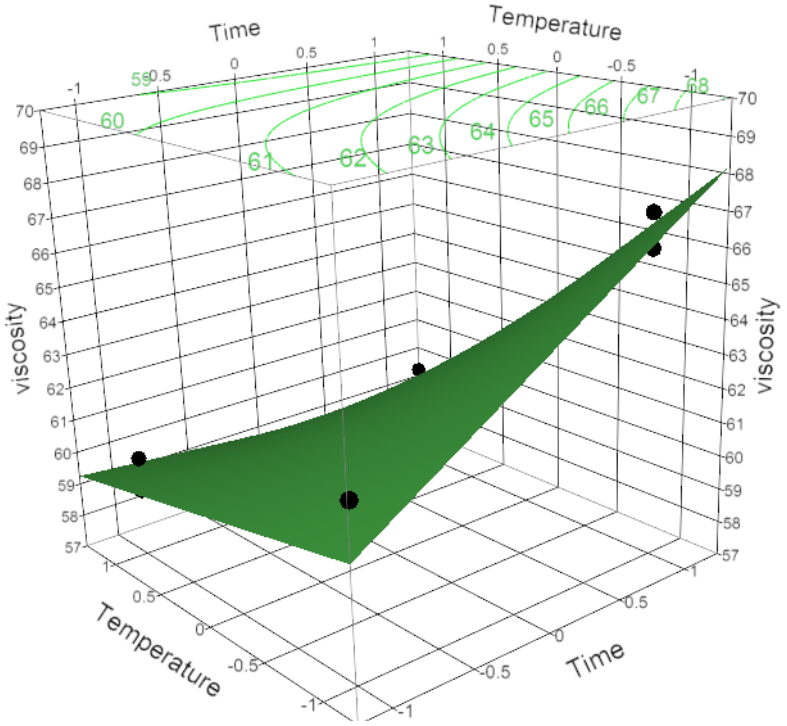


Figure 2.12: Model for viscosity from factorial example dataset

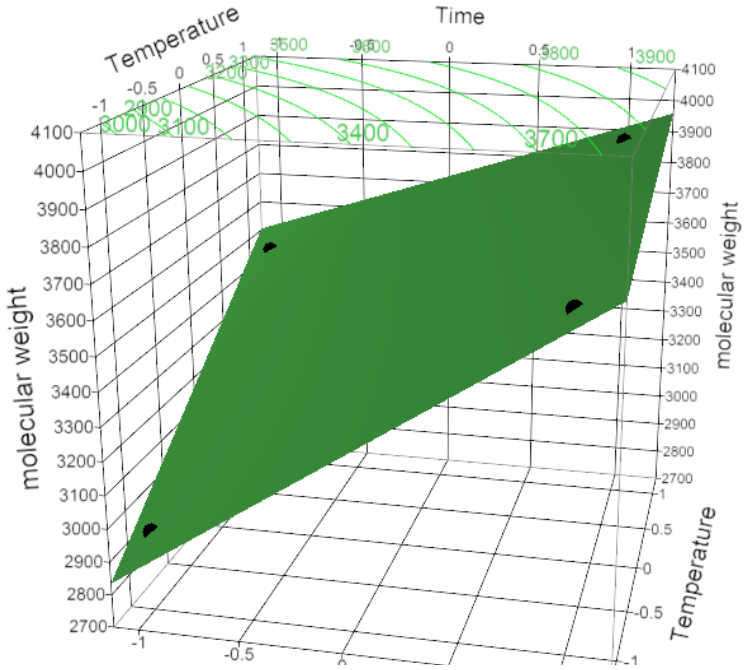


Figure 2.13: Model for molecular weight from factorial example dataset

nation [2]. R^2 is defined in (2.28) as 1 minus the ratio of variability in the residual where SS_{total} is the total the sum of squares, as shown in (2.29), SS_{err} is the sum of square of the residuals, as shown in (2.30), $\hat{y}_i$ is the estimated model that corresponds to the output y, and $\bar{y}$ is the mean of the observed values of the output. This is measure of how much of the variation can be attributed to the model as opposed to the error. A perfect fit corresponds to an R^2 value of 1. For example, a value of 0.4 means that 40% of the variability in the observations can be explained by the model, while 60% is caused by error.

Typically, statistical analysis packages provide a plot of actual observed values compared to the estimated values. An example of this type of plot, based on a dummy data set analyzed in *JMP* 9.0 where R^2 is 0.71, is shown in Figure 2.14. The diagonal line represents a perfect fit between predicted and actual and the dotted blue line is the mean. Each observed data point is plotted against the predicted value. The difference between the point and the perfect fit is the residual. Further residual analysis is often performed where the distribution of the residuals is plotted and fitted against a normal distribution.

$$R^2 = 1 - \frac{SS_{err}}{SS_{total}} \tag{2.28}$$

$$SS_{tot} = \sum_i (y_i - \bar{y})^2 \tag{2.29}$$

$$SS_{error} = \sum_i (y_i - \hat{y}_i)^2 \tag{2.30}$$

A summary of the statistical fit for the factorial design in the reference example is shown in Table 2.14. These high measures indicate that based on only the observed data points, the fit of the estimation model is very good. The problem with this, is that as other data points are added, there is most likely curvature in the system that the factorial analysis cannot discern. This is where more complicated designs are used. In subsequent sections, the statistical fit of the RSM design will show a drop in the R^2 statistical fit. This is due to the higher number of data points that are incorporated into the calculation. With more data points, there is higher chance that the estimation will not match the observed values.

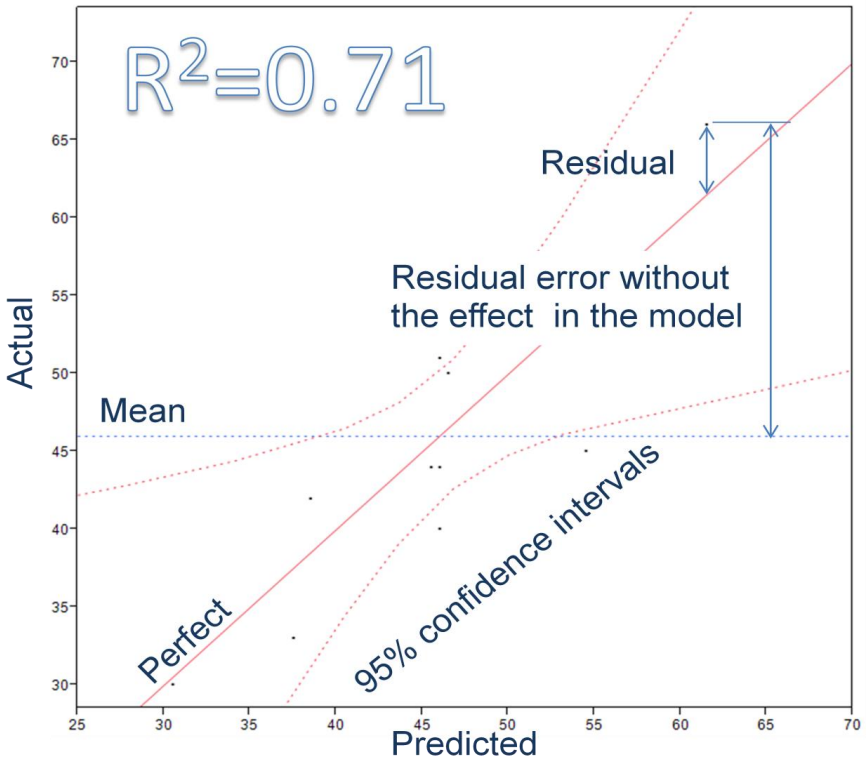


Figure 2.14: Example of an actual by predicted plot with an R^2 value of 0.71

Table 2.14: Summary of R^2 values from factorial analysis

Output	R^2
Yield	0.960508
Viscosity	0.97037
M. Weight	0.999981

2.10 Identifying Interactions

The previous sections have shown how to implement a factorial design, identify input parameter significance, and develop an estimation model. Another important outcome of the factorial design is the capability to quantify and show the interaction between parameters. The results caused by these interactions are often referred to as interaction effects in the literature. Interactions are indicated when change in one factor fails to reproduce the same result if another factor has changed. Traditional approaches of holding all variables constant and changing one variable at a time cannot identify these effects. Conversely, changing multiple variables at a time is inherent in the factorial and experimental design philosophy.

Analytically, interactions are quantified in an analysis of variance (ANOVA) procedure. This technique, discussed in Section 2.8.1 identifies the significance of each component of an estimation model on the measured output. For example, Table 2.13 lists model coefficients and their statistical significance. If the β_3 term's $p-value$ is less than 0.05 then it indicates that there is an interaction between Time and Temperature. A visual approach using interaction plots provides a more intuitive method to identify these relationships. These plots are created by overlaying one plot of the output when a variable is set to its maximum on top of another chart where the same variable is set to a minimum. Figures 2.15, 2.16, and 2.17 show interaction plots from the reference design example for yield, viscosity, and molecular weight respectively.

These non-traditional plots assist in visualizing the result of changing two variables at a time. To read the plot first look at the labels along the right side vertical axis. These labels indicate the input that will be fixed at -1 and +1 while the other input is varied across its entire range and indicated on the x-axis. A blue and red line are plotted on top of each other. The blue line corresponds to an input fixed at +1 and a red line to the same input fixed at -1.

Strong interactions between input pairs are indicated when the blue and red lines are non-parallel. If there was no interaction between these two inputs then the red and blue lines would be parallel. Parallel red and blue lines indicate that there is no appreciable effect from the change in the variable on the x-axis. However when the red and blue lines cross, it indicates that a change in the variable on the x-axis causes different two different reactions when the other variable is at a

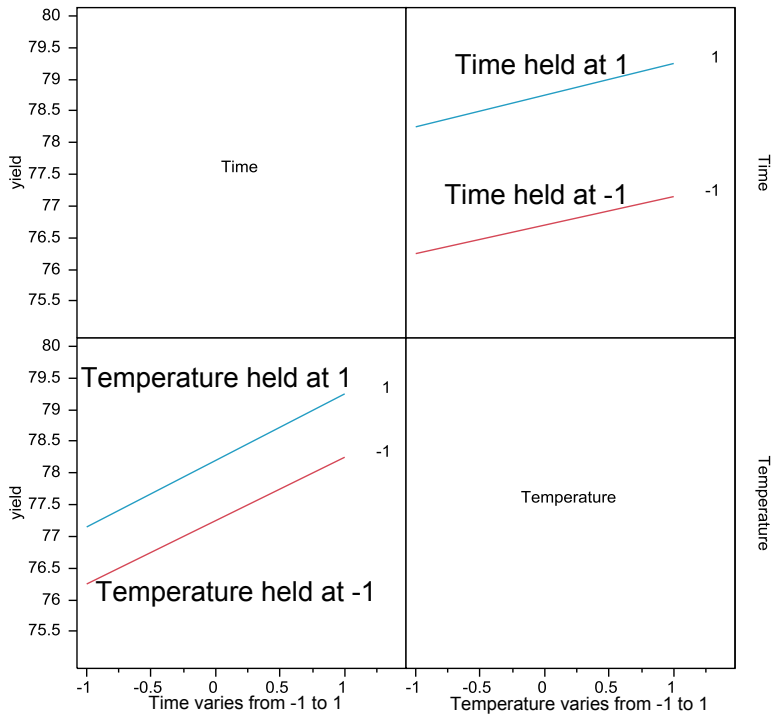


Figure 2.15: Interaction plot for yield from reference example

minimum and at a maximum. This corresponds to the greater statistical significance assigned to the Time*Temperature term for the viscosity and molecular weight outputs.

Interpretation of interaction plots is tied to the results of ANOVA. Referring to Table 2.13, the coefficient associated with the term Time*Temperature has strong significance for the viscosity and the molecular weight output. However, for the yield output it is not statistically significant. This is reflected in interaction plots in terms of how parallel the red and blue lines are. Parallel or near-parallel line are indicative of insignificant interaction. Consequently, Figure 2.15 shows nearly parallel red and blue lines. This indicates that trend in the output is the same regardless of whether time or temperature changes independently of each other.

However, Figures 2.16 and 2.17 show that the red and blue lines are non-parallel. This indicates that the trend in the output response from either time or temperature set to the minimum will be different when they change to maximum levels.

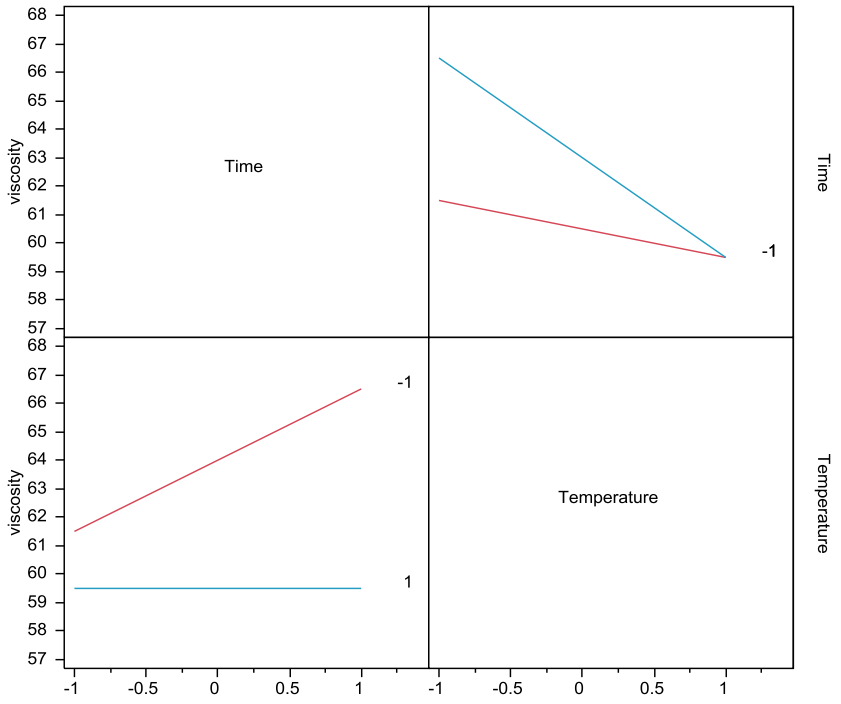


Figure 2.16: Interaction plot for viscosity from reference example

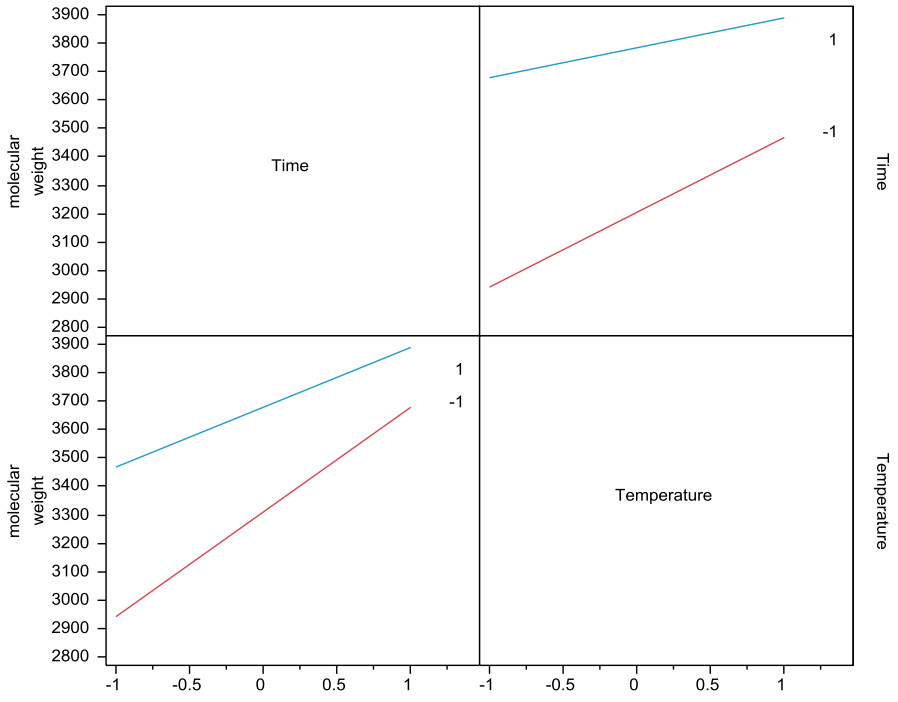


Figure 2.17: Interaction plot for molecular weight

2.11 Results of CCD Design for Reference Example

Inspection of the full data set of the reference example reveals a face-centered CCD. The analysis of the collected data yielded three response surfaces as shown in Figures 2.18, 2.19, and 2.20. The estimated model coefficients are listed in Table 2.15. These coefficients were defined in (2.22) and the Intercept term (β_0) is the value of the model when all input parameters are 0. These plots illustrate how the inclusion of additional data points and a second-order model produce more realistic model of the outputs. Table 2.16 lists the R^2 measure of fit for these second-order models. As discussed, the measure of fit has gone down compared to the first linear models listed in Table 2.14.

Table 2.15: Estimated RSM model coefficients

Term	Coefficient	Yield	Viscosity	M. Weight
Intercept	β_0	79.50	69.28	3332.19
Time	β_1	1.10	0.70	265.60
Temperature	β_2	0.53	-1.70	195.40
Time*Time	β_{11}	-1.40	2.01	-32.66
Time*Temperature	β_{12}	0.03	-1.25	-79.00
Temperature*Temperature	β_{22}	-0.65	-9.99	167.34

Table 2.16: Tabulation of R^2 for RSM analysis of reference example

Output	R^2
Yield	0.79452
Viscosity	0.879606
Molecular Weight	0.887019

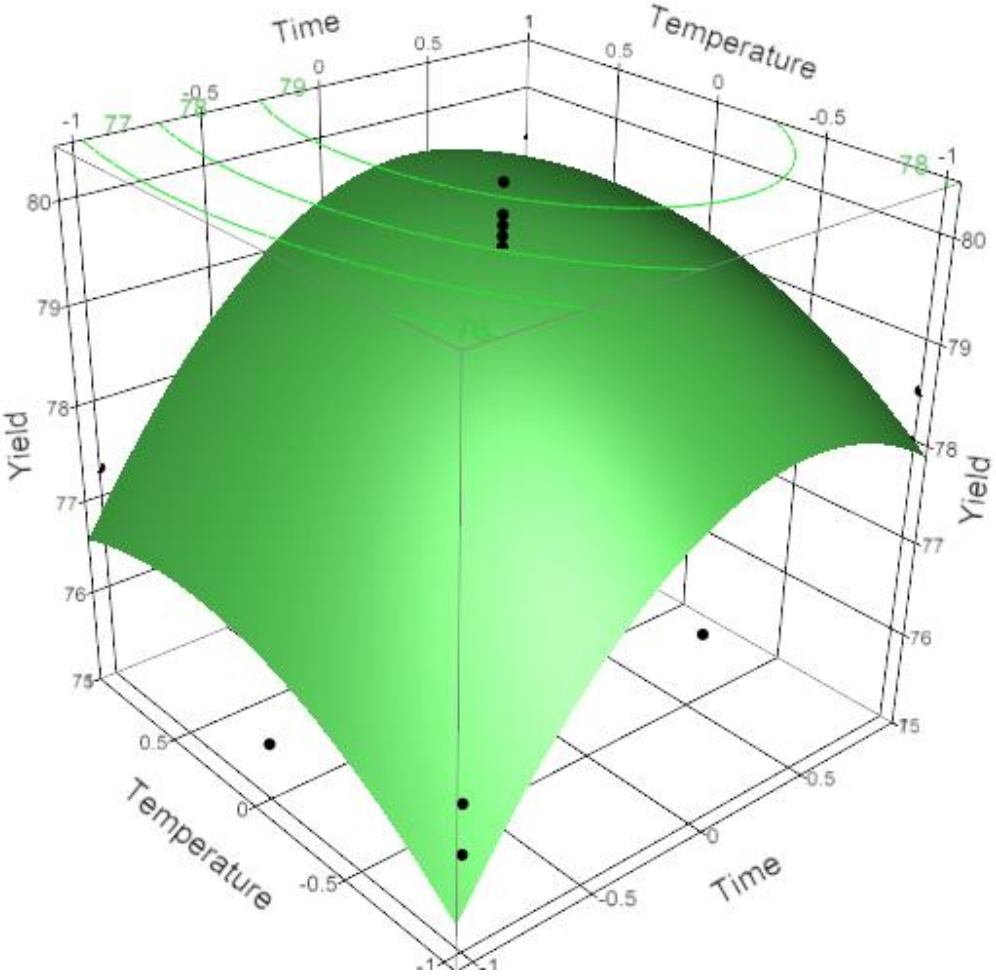


Figure 2.18: Yield response surface

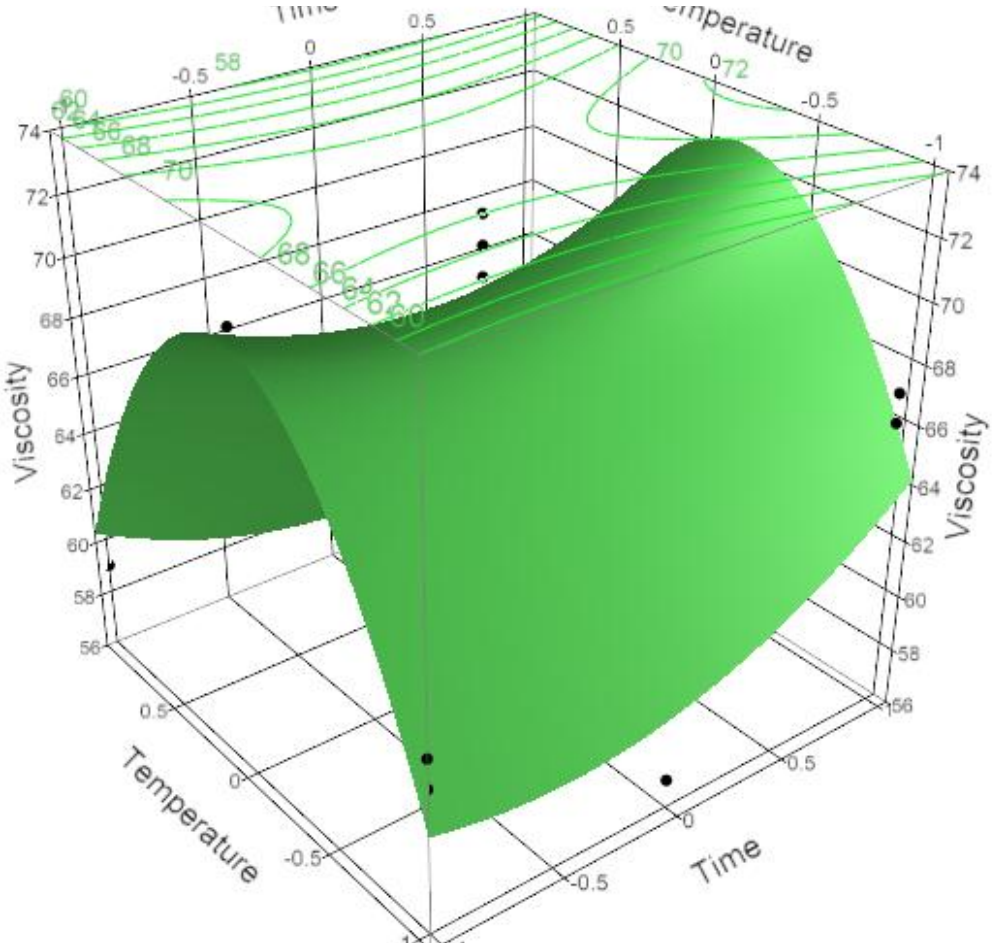


Figure 2.19: Viscosity response surface

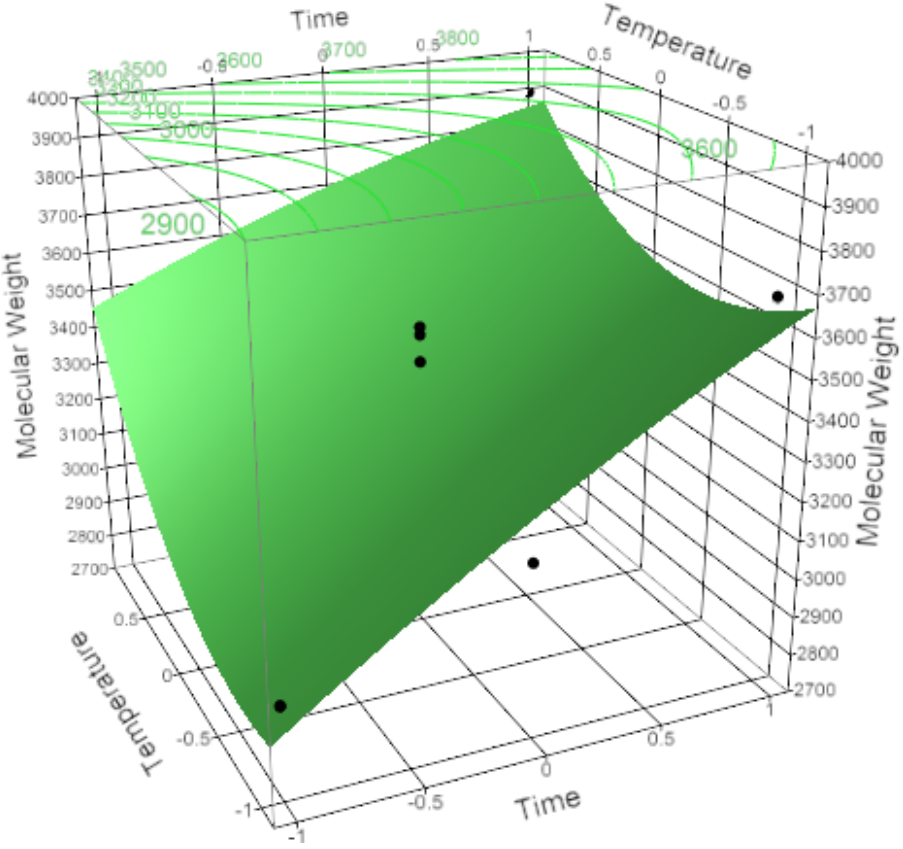


Figure 2.20: Molecular weight response surface

2.12 Identifying a Solution From Multiple Outputs

The statistical methods produce estimation models for each of the observed output metrics. While solving a single model for a maximum or minimum is straightforward, it is more typical that a solution is identified from a combination of metrics. Section 3.2 describes the utility and fitness approach that is prevalent in cognitive radio. This section presents two alternate methods of interest to those implementing statistical techniques to profile wireless performance. The first technique is a contour overlay method based on graphical techniques and are useful for a small number of outputs. It provides a qualitative method useful for system users to easily find overlaps between multiple performance criteria. The second technique, is called desirability analysis, and is more similar to the utility/fitness approach. Desirability is introduced here because most statistical analysis software contains built-in desirability analysis.

2.12.1 Contour overlaying approach

When multiple outputs are involved, different approaches are required that incorporate all metrics. An informal approach that an experimenter can implement if the system is not too complex (between two and three outputs) is known as *contour overlay*. The three-dimensional surface developed in the response can be viewed in terms of contours of constant value. This is similar to the concept of a contour map where elevations of constant value are shown. An experimenter can define ranges of desired performance. This is an informal approach to a constrained optimization method.

In this example, an experimenter defines the three desired regions: yield$\geq$ 78.5, 62 $\leq$ viscosity$\leq$ 68, and molecular weight$\leq$ 3400. From the surface profiles shown in Figures 2.18, 2.19, and 2.20, the shapes of these contours will be ovals for the yield and molecular weight, and two ridges for viscosity. Figure 2.21 shows these three profiles overlayed on top of another. The regions where all three profiles overlap provides a range of temperature and time inputs that will meet the defined goals for all the metrics. This approach is more informal and useful in simple cases with limited number of outputs. A more formalized approach is required for identifying a single desirable solution rather than a range.

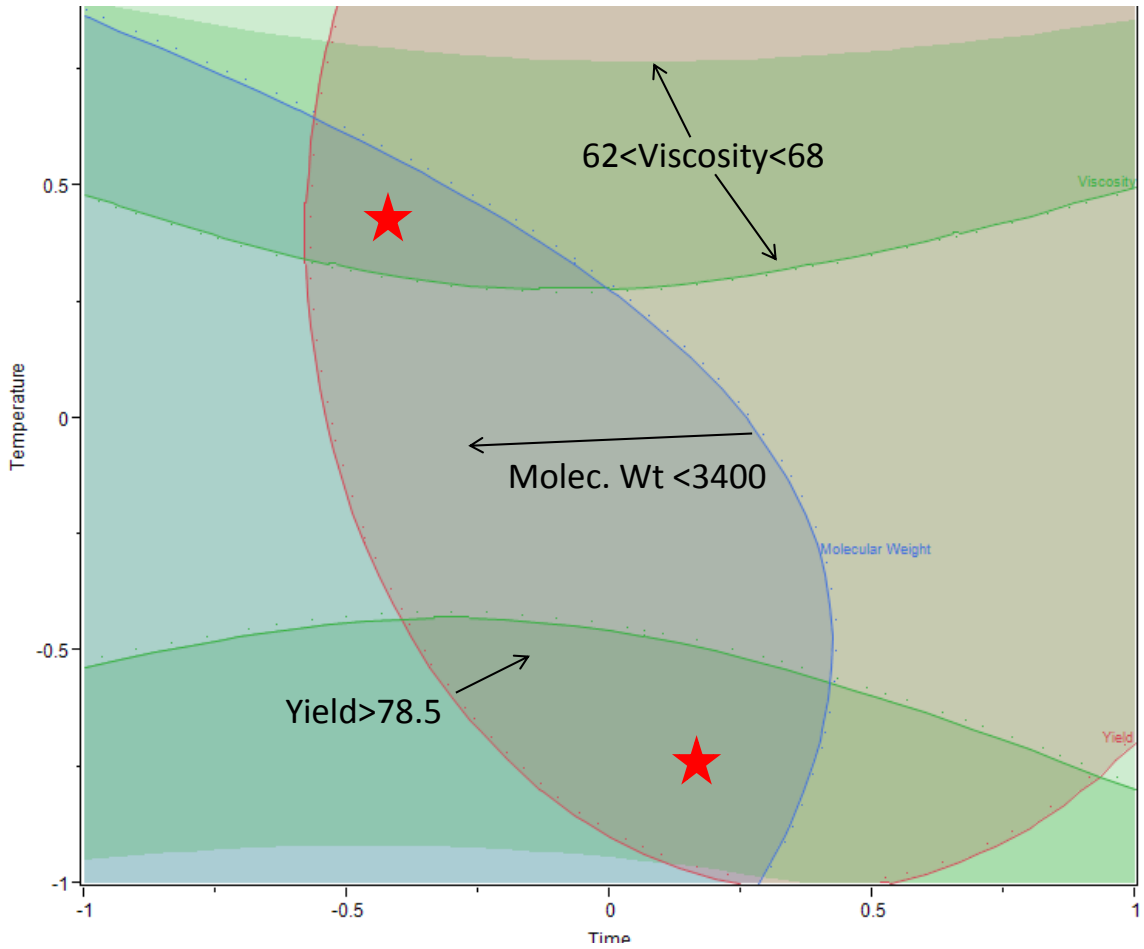


Figure 2.21: Contour overlay for example for $yield \geq 78.5$, $62 \leq viscosity \leq 68$, and $molecularweight \leq 3400$

2.12.2 Multi-variate Optimization Through Desirability Functions

The problem of identifying a solution from multiple output metrics is well known in cognitive radio [6]. Cognitive radio approaches to addressing multi-criteria optimization center around creating utility functions in order to place outputs of different units onto a normalized scale. These utilities are then combined into a single quantification through a fitness function [10, 30]. Fitness functions can incorporate varied weightings and methods of how to combine the utilities. Regardless, the goal is to identify one value that represents the 'goodness' of a solution from combined metrics.

Historically, experimental design has followed a similar type of analysis. The field typically uses a method called *desirability* analysis which is essentially the same approach. The difference is the type of function used to normalize the metrics. Most statistical software packages come with desirability analysis tools, therefore this background includes a summary of the approach. For all intents and purposes, desirability analysis and utility analysis are the same thing. This alternative approach comes from industrial process design optimization. Multiple outputs make up the overall quality of a product. The desirability approach developed by Derringer and Suich is one of the most widely used multi-criteria optimization methods in production and processing industry [31, 32, 29]. This method is based on the philosophy that a process or product is made up of several quality measures and if any one of them falls outside of a defined limits, then the entire product is unacceptable. The difference between utility functions and desirability functions is simply the manner in which the overall utility or desirability is calculated. Utility functions are based on a modified hyperbolic tangent while desirability are defined based on functions defined below [10]. Ultimately, they strive for the same goal of normalizing performance onto a similar scale and identifying most desirable and least desirable conditions.

In general, the inputs $x_1, x_2, \cdots, x_m$ drive measurable performance or quality criteria defined as $y_1, y_2, \cdots, y_n$ such that $y_i = f(x_i, x_2, \cdots x_m)$ [33]. When models of the outputs are developed, the outputs are estimated with $\hat{y}_i = f(x_i, x_2, \cdots, x_m, \varepsilon)$, where ε is the statistical error. A measure of desirability is defined in (2.31). An aggregate of individual desirabilities is combined into the unitless metrics D, as generally shown in (2.32).

$$d_i(y_i)(i = 1, 2, \cdots, n), d : \mathbb{R} \rightarrow [0, 1] \tag{2.31}$$

$$D = f(d_1, d_2, \cdots, d_n), D : [0,1]^n \rightarrow [0,1] \tag{2.32}$$

A desirability function, $d_i(y_i)$ is defined for each output, y_i, such that $d_i(y_i)$ ranges between [0,1] where a value of 1 corresponds to the most desirable value of y_i. Typically the desirability function is defined for three cases corresponding to the goals of the output: maximization, minimization, or 'target is best'. This paper utilizes maximization for the outputs spectral efficiency, throughput, and SNR, while minimization is associated with BER and PER. The desirability function for maximization is defined in (2.33), and minimization in (2.34), where L_i, U_i, T_i are the lower, upper, and target values and s determines how crucial reaching the target value is. When $s = 1$ the function increases linearly towards T_i [29].

Individual desirabilities are combined utilizing a geometric mean leading to an overall desirability, D, as shown in (2.35). There is a severe penalty if any d_i falls to zero. If this occurs, then the entire desirability, D, goes to zero. Similarly, D will be 1 only if all the desirabilities have the maximum value of 1.

$$d_i(\hat{y}_i) = \begin{cases} 0 & \text{if } \hat{y}_i < L_i \\ \left(\frac{\hat{y}_i - L_i}{T_i - L_i}\right)^s & \text{if } L_i \leq \hat{y}_i \leq T_i \\ 1.0 & \text{if } \hat{y}_i > T_i \end{cases} \tag{2.33}$$

$$d_i(\hat{y}_i) = \begin{cases} 1.0 & \text{if } \hat{y}_i < T_i \\ \left(\frac{\hat{y}_i - U_i}{T_i - U_i}\right)^s & \text{if } T_i \leq \hat{y}_i \leq U_i \\ 0 & \text{if } \hat{y}_i > U_i \end{cases} \tag{2.34}$$

$$D = \left(\prod_{i=1}^{n} d_i\right)^{\frac{1}{n}} \tag{2.35}$$

Optimization of D can be approached from several ways. Statistical software packages, such as *SAS JMP 9.0* implement proprietary search methods for identifying the final solution. Other options include constrained optimization and nonlinear programming methods. Additionally, one can implement other heuristics such as genetic algorithm search.

The reference example was processed using the JMP response profiler tool and simplified desirability functions for illustrative purposes, shown in Figure 2.22. Three desirability functions, shown in the right column, were defined based on a 'higher-is-better' perspective for yield and viscosity, and

'lower-is-better' perspective for molecular weight. Typically a process oriented design might seek to match a target viscosity or yield.

The profiler tool plots each output twice. The first column of graphs show the outputs where the time variable is across the x-axis. The graph displayed is the output with the other variable fixed at the dotted vertical red line in the second column. Similarly, the middle column plots the outputs with the temperature variable on the x-axis. These plots are graphed with time fixed to where its vertical red line is. This tool is dynamic so the vertical red lines can be moved and the graphs will change accordingly. The first row shows the outputs converted onto a desirability scale The statistical package searches across all the combinations of the two variables and identifies a solution based on the highest desirability. In this simple example, the solution identified (one the normalized [-1,+1] scale was time=-0.174 and temperature=-0.124.

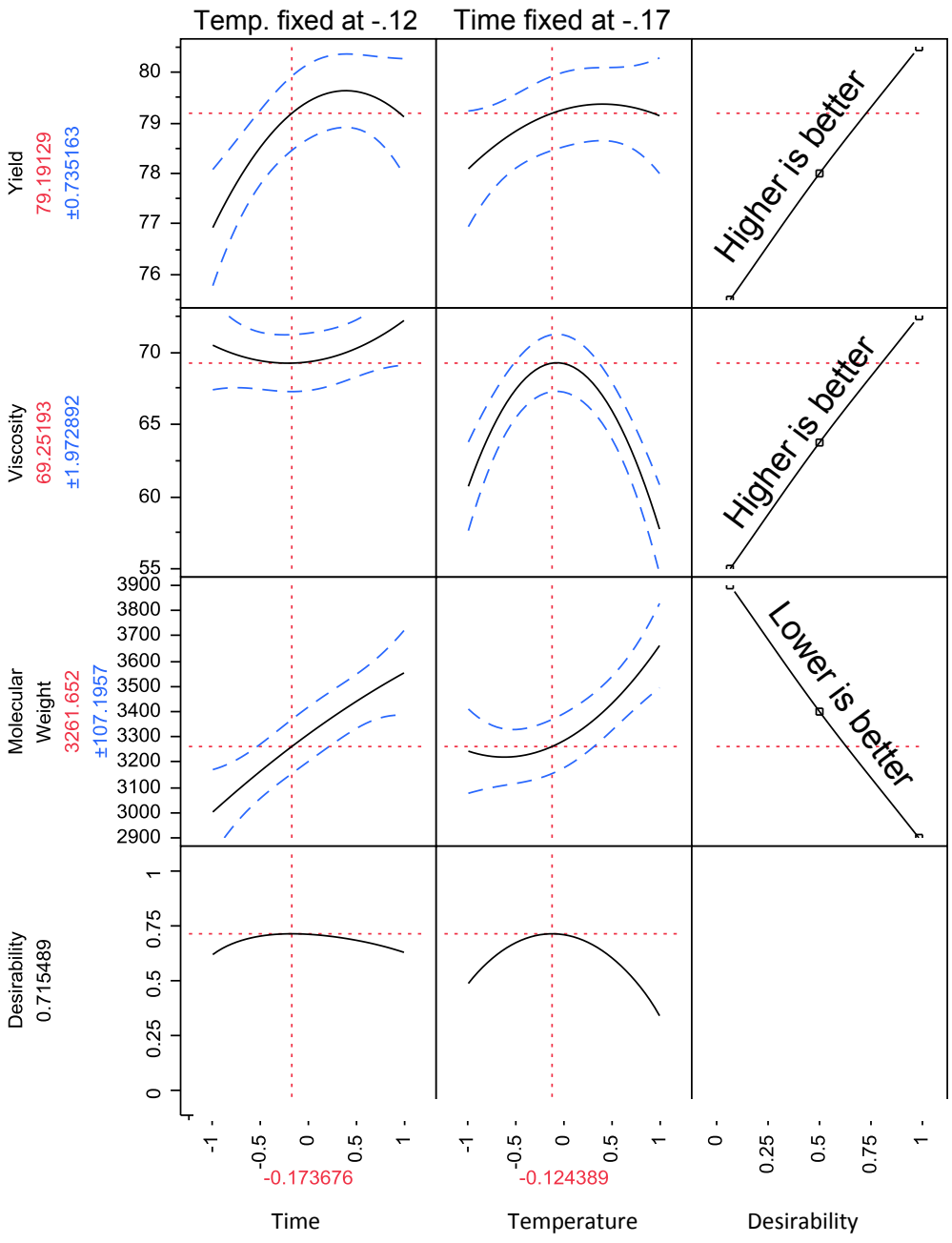


Figure 2.22: Response profiler solving tool in JMP

2.13 Related Work

Many fields of science, engineering, and manufacturing have utilized design of experiments since the 1930s. However, it has seen relatively little use within wireless communications, networking and cognitive radio. The purpose of this section is to summarize prior art related to design of experiments, response surface methodology (RSM), and Taguchi analysis within a wireless communications and cognitive radio application space that ties to this dissertation. In general, experimental design methodologies are well suited to analyzing wireless and networking systems with some caveats. System definition in terms of the input parameters and output metrics is a crucial step. The boundaries, discrete values, and quantity of input parameters will affect the implementation of the experimental design. Similarly, the measurable resolution of output metrics and defining how multiple metrics will be combined play key roles.

Wireless and networking research in experimental design has generally focused on three uses. The first is identifying parameter significance and interactions. The purposeful changes made to input parameters of experimental designs provide a tool for identifying significance and interactions that typical testing might not find. The second use is for developing empirical performance models as wireless systems can be difficult to model analytically. The experimental designs can yield models that provide accurate characterization of performance in small regions. Finally, experimental design provides an approach for near-optimization of design parameters. Areas such as antennas, filters, and waveform design with distinct inputs and defined measures of success are well suited to experimental methods.

2.13.1 Wireless Communications and Wireless Networks

This dissertation asserts that experimental design can develop estimation models of system performance which can identify desirable configuration parameter settings. Past work in investigating the design factors in direct sequence spread spectrum (DSSS) receivers illustrates this as well. Factorial designs provided an empirical foundation for assessing the design factors for spectrally encoded waveforms and their effect on performance of a DSSS receiver [34, 35]. The goal was to use factorial and RSM to identify the best combination of input parameters in order to min-

imize a DSSS receiver's bit error rate. The input factors consisted of configuration elements of a spectrally modulated, spectrally encoded (SMSE) waveform. The two factors were the number of SMSE carriers and symbol duration. Each factor was mapped to three levels (low, medium, high) that corresponded to the typical experimental design notation of [-1, 0, +1]. The measured outputs considered a DSSS receiver's bit error rate as a measure of coexistence. Results showed the method was capable of successfully predicting the system behavior and identification of the SMSE configuration parameters to minimize DSSS system bit error. The design consisted of a 2 factor, three level factorial statistical design for factor screening and a RSM design that incorporated four additional center points. This is consistent with a central composite design that this dissertation implements on a cognitive radio platform.

A benefit of statistical approaches, claimed by this dissertation, are that they can operate without previous system knowledge. Fundamentally, systems are distilled to black boxes where the experimental design only requires control of the input parameters and observation of the final performance results. Past work in multi-user detection illustrated this capability. A Taguchi statistical approach was implemented in the study of multiuser detection in a direct sequence code division multiple access system under multipath conditions [36]. The Taguchi method provides an alternative to the traditional experimental design approach that focuses on variation around a target goal as opposed to global system optimization. Simulations show the Taguchi optimized design was capable of performing better than conventional receivers and performance was similar to mean-square-error detectors. Advantages identified over conventional designs include insensitivity to channel noise or starting configuration parameters and that the Taguchi method can operate blind without the need for channel estimation.

New efforts in optimizing radio network parameters of Long Term Evolution (LTE) have also used Taguchi methods [37]. This work employs a Taguchi approach to adjust radio network parameters in contrast to the existing commercial planning tools that rely on local search. The author's support this dissertation's claim that the success of local search lies in the initialization parameters. Their simulation results showed the Taguchi approach provided comparable performance to simulated annealing for identifying power control and antenna azimuth optimizations. In addition, the Taguchi method improved upon the simulated annealing for antenna tilt optimization.

Further advantages of the empirical approach relate to its capability of developing more realistic estimation models when compared to analytical models. RSM designs were implemented within Wi-Fi voice over Internet protocol (VOIP) simulation to develop a meta-model of performance [38]. In this case, meta-model is defined as a mathematical expression of a simulation model, or a model of a model. Inputs manipulated were VOIP sample interval, and PHY data rate. The RSM based meta-model showed closer fit to the simulation model while the analytical model overestimated performance.

2.13.2 Networking

An issue seen with application of factorial and RSM is encoding qualitative parameters. These methods are intended for continuous values, however one can define coded values as illustrated in past work applying experimental design to networking. Foundational work presented an empirical study of the interaction effects amongst routing and medium access control (MAC) layer protocols under varying mobility parameters [39]. Outputs included latency, throughput, number of packets received, and fairness. Control factors included choice of routing protocol, selection of MAC protocol. The factorial analysis contributed methodology for quantifying the interaction between protocols and other parameters such as injection rate. Results indicated that the entire protocol stack must be considered as a single model in order to improve interaction effects amongst all the factors.

2.13.3 Design of Experiments in Cognitive Radio

The inspiration for using experimental design within the decision cycle of cognitive radio came from prior work by Weingart et al. [40, 41]. A predictive model for cognitive radio was proposed where experimental designs were implemented on a wireless link enabling development of a performance model. The study investigated a wireless link simulated in OPNET in the presence of a third interferer. The predictive algorithm suggested factorial designs as a technique to identify optimum parameter settings to meet one of several outputs. The suggested parameters settings are then pared down by repeating the design with focus on another meter. This process is repeated for

each output available until a final parameter setting is identified. The results provided accurate estimation for throughput but not for latency. Building upon this work, the authors developed a reconfiguration algorithm that draws its decision from multivariate analysis on the system. The algorithm was implemented on an open-source software controller for off-the-shelf 802.11 wireless cards [42].

Follow-on work developed implemented a recursive design process on wireless MAC platform exploring power, frequency, maximum transfer unit (MTU), forward error coding, and data rate [43]. Their design incorporated 128 probe tests. The decision-making architecture incorporated enhancements over the basic factorial analysis of variance by first filtering out insignificant factors and then prioritizing best candidate configurations based on statistical results. This enhancement provided faster decisions.

2.14 Summary

This chapter provided fundamental overview of experimental design techniques used in this dissertation. The methods included factorial design, fractional factorial designs, RSM, and the Taguchi method. A summary of the designs and their relative advantages and disadvantages are given in Table 2.17. Factorial designs are capable of identifying both input factor significance and interactions, however as the number of variables increase their benefits diminish. The models developed from factorial designs are limited to linear models which may not meet the resolution desired by the user. Fractional factorials provide less information than factorial designs at a lower cost in terms of number of tests. They are used as a first screening in order to identify insignificant effects. Taguchi designs are a type of fractional design that also incorporate variation around the mean. The disadvantages include limitations to discrete inputs and lack of a continuous estimation model. The CCD and BB designs provide continuous estimation models and enable surface profiles. The BB design avoids test cases at the extremes of the design space. This limits its estimation capabilities at the vertices, however improves performance closer to the center of the input ranges. In contrast, the CCD design is an extension of the full factorial design, with design points at the extremes of the input ranges. Hence, it provide better estimations at these levels. Both CCD and BB provide

quadratic models which incorporate curvature and provide better estimation resolution than factorial designs. The chapter also reviews the basic implementation and analysis steps for designs, which includes defining a design, implementation on a system, analyzing the data, developing performance models, interpreting model fit, and identifying a solution from multiple output models. Finally, a brief review of related work in wireless communications and networks was presented.

Table 2.17: Summary of designs

Design	Advantages	Disadvantages	Uses
Factorial	identifies factor significance identifies interactions linear estimation model	poor scalability no curvature	screening simple models
Fractional Factorial	reduction in number of tests identifies factor significance	no model no interactions	screening
Taguchi	reduction in tests incorporates variation	assumes no interactions discrete inputs only	robust design discrete solutions
CCD	continuous model estimates at extremes	more variation at extremes more tests than factorial	quadratic models 3D surface profiles
BB	better near center point continuous model	Worse at extremes	quadratic models 3D surface profiles

Chapter 3

New Techniques for Cognitive Radio Operations

3.1 Introduction

Heuristic optimization is one current standard for cognitive-radio decision making. These algorithms, such as the genetic algorithm (GA), search potential radio settings by evaluating functions based on performance estimation formulas. While seminal to the cognitive radio field, the methods have limitations related to accuracy of estimations and efficiency in operating in new situations.

Specifically, GA cognitive engines, rely on implicit system models that may not accurately represent performance under operational conditions. For example, current architectures rely on estimations based on basic channel models. Without modification, simple models perform poorly outside of simulation and controlled environments. Attempts to incorporate multiple models to accommodate different conditions can improve accuracy. However, it is difficult to identify a model for every conceivable situation. Furthermore, multiple models require additional sensors and logic to determine which approach to implement.

This chapter introduces new techniques to address the practical hurdles faced by over-the-air implementation of standard architectures. First, a unique method of calibrating existing performance

estimation methods is presented. This approach adapts basic formulas to provide more realistic estimations in over-the-air operations in a specific environment. The strategy is further enhanced by a dynamic algorithm that identifies an adjustment factor for adapting estimation formulas prior to normal operations. While an improvement over unadjusted estimation formulas, this approach is limited because the adjustment must be performed for every new environment.

Experimental design, introduced in Chapter 2, is proposed as a unique approach to address remaining limitations. The method presents a compliment to standard architectures by developing empirical models without the requirement for past experience or knowledge of environmental conditions. The statistical approach is based solely on observed metrics which provide a closer representation to the current operating conditions. Chapter 4 implements these techniques under real-world conditions and compares their performance under a common frame of reference. This approach can also compliment existing approaches by replacing existing estimation functions in a GA, providing training cases for CBR, and validating existing. This dissertation does not explicitly implement these complimentary applications and focuses more on developing the initial application of experimental methods.

The structure of this chapter is as follows: first a general approach for drawing performance comparisons is presented. Utility and fitness functions, based on existing work, provide an underpinning required for implementing multi-criteria optimization and for comparing different statistical designs to standard architectures. Next, existing GA cognitive engine processes and their limitations are introduced. An adjustment method that enhances existing approaches is introduced followed by an algorithm for dynamic implementation of the procedure. Statistical methods are then illustrated on a basic level to a wireless application. This includes discussion of general performance comparisons and limitations. An abstract process flow provides a framework for applying the methods onto a variety of platforms and applications. Finally, the chapter concludes with a summary of the approach and potential impacts to cognitive radio.

3.2 Drawing Performance Comparisons

In order to compare cognitive radio techniques and draw meaningful conclusions output metrics must be normalized onto a similar scale. Past work by Rondeau, Gaeddert, and Zhao discussed the issue extensively [8, 44, 14]. Three elements are considered, including utility functions for normalizing metrics, a method for combing all the utility functions, and defining the operational goals of the radio. These goals drive the weighting for each metric's utility function. This concept of fitness will be referred to throughout this dissertation when discussing comparative performance involving several metrics.

The utility functions implemented for normalizing the output metrics on a scale of [0,1] were based on previous work by Gaeddert [44]. A generic utility function for the metric β is given by (3.1) where $\dot{\beta}$ is a target metric, σ is a spread parameter, and η is a threshold parameter.

$$q\left(\beta, \dot{\beta}, \eta, \sigma\right) = \frac{1}{2} + \frac{1}{2}\tanh\left\{\sigma\left[log\left(\frac{\beta}{\dot{\beta}}\right) - \eta\right]\right\} \tag{3.1}$$

This approach transforms the utility into a monotonic and bounded function based on a hyperbolic tangent. By defining q_0 and q_p such that $q(\dot{\beta}) = q_o$ and $q(p\dot{\beta}) = q_p$, the spread parameter is then defined by (3.2) and the threshold parameter by (3.3). Therefore, utility functions can be fully defined in terms of the parameters $q_0, \dot{\beta}, q_p$, and p.

$$\sigma = \frac{1}{2\log p}\left[\log\left(\frac{1}{q_0} - 1\right) + \log\left(\frac{q_p}{1 - q_p}\right)\right] \tag{3.2}$$

$$\eta = \frac{1}{2\sigma}\log\left(\frac{1}{q_0} - 1\right) \tag{3.3}$$

In order to aggregate the multiple utility functions into a single fitness quantification, a weighting system and method for combining utilities is required. Approaches include weighted summation, product of utilities, or weighted geometric mean. This implementation used a geometric mean approach as, shown in (3.4) where q_i is the utility and w_i is the associated weight for that metric. This has similarity to the desirability method of combining multiple metrics shown in (2.35). This approach yields a fitness close to 1 only if all the utilities are desirable. If one utility approaches 0 then the overall fitness also drops. The weights are not required to sum to unity, as is required in a weighted sum approach. Additionally, the weights do not have to be less than one.

$$F = \sqrt[N]{\prod\left(q_i^{w_i}\right)} \tag{3.4}$$

3.3 Current Approach to Genetic Algorithm Cognitive Engines

Evolutionary processes provided the inspiration for the GA as a tool for optimization of a mathematical function. This section briefly describes the concept of genetic selection. Biological cells are defined by strings of DNA known chromosomes which are comprised of sets of genes. These genes define physical attributes of the cell or organism, such as hair color. As organisms reproduce, the genetic information from both parents is combined into new chromosomes comprised of genes from both parents. In addition, random mutations occur that change individual genes. The concept of 'survival of the fittest' states that the best combination of genes and their resulting chromosomes yields the strongest *individual* which will survive the longest.

These concepts led to the development of the GA. The first step in implementing the GA requires that a problem be defined such a way that its solutions can be encoded into a chromosome. In the case of CR, the configurable radio parameters, such as transmit power, modulation, coding, or packet size represent genes of a chromosome. GA's typically encode solutions as bit strings of 1's and 0's. Once the parameters are encoded into genes and combined into a chromosome, the GA starts by creating a *population* of several individuals. The fitness of each individual needs to be determined in order to quantitatively compare the population. Each individual's fitness is assessed, and individuals are ranked in order of highest fitness. Top individuals become *parents* for the next generation of the GA, while the weakest performers are discarded. The *children* of surviving parents are created by crossing over genes between parents, as shown in Figure 3.1. In this manner, strong characteristics from two sets of parents are combined. In addition, random mutations of single bits enable searching more of the variable space. A critical step in this process

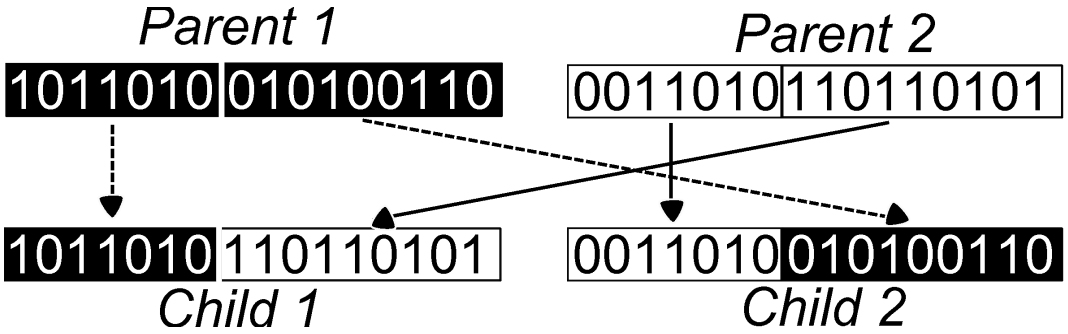


Figure 3.1: Example of crossover of genes from parents to children

is how the GA calculates overall fitness of each individual. Current architectures rely on implicit formulas to estimate bit error rate (BER) of a potential configuration. From this estimation, the utility of additional metrics are estimated and used to calculate overall fitness. This specific process and the liabilities related to the reliance on theoretical estimations are discussed next.

3.3.1 GA Performance Estimation Formulas

Currently, performance estimation begins with theoretical formulas of BER. This calculation depends on several factors, including the modulation, channel type, signal to noise ratio (SNR), and the transmit power. There is a trade off between accuracy of the formulas and complexity. One approach is to provide estimation formulas for several channel models to provide greater flexibility for varied conditions. The requirement for sensing and decision logic to determine what model to use limits this approach. Additionally, even with increased complexity, it is unrealistic to be able to characterize every situation encountered.

Foundational work by Rondeau focused on a performance estimation formulas using a relatively simple additive white Gaussian noise (AWGN) channel because of the difficulty in determining the type of channel [7](page 57). Example formulas of BER (P_e) for binary phase shift keying (BPSK), M-PSK, and M-quadrature amplitude modulation (QAM) are listed in (3.6), (3.7), and (3.8), where energy per bit to noise energy ratio, E_b/N_0, is given by (3.5) in dB, B is the bandwidth in Hz, M is the number of symbols in the modulation, S is the signal power in dBm, and N is the noise power in dBm.

$$\frac{E_b}{N_0} = \gamma = 10\log_{10}\left(\frac{B}{R_s\log_2(M)}\right) + (S - N) \tag{3.5}$$

BPSK:

$$P_e = \frac{1}{2}erfc(\sqrt{\gamma}) \tag{3.6}$$

M-PSK(M$\geq$2):

$$P_e = \frac{1}{\log_2(M)}erfc\ \left(sin\left(\frac{\pi}{M}\right)\sqrt{\log_2 M}\sqrt{\gamma}\right) \tag{3.7}$$

M-QAM:

$$P_e = \left(\frac{2}{\log 2(M)}\right)\left(\frac{\sqrt{M}-1}{\sqrt{M}}\right) erfc\left(\sqrt{\frac{3\log_2 M}{2(M-1)}\gamma}\right) \tag{3.8}$$

This approach works well in simulation and highly controlled environments, however, theoretical models do not represent over-the-air conditions well enough to enable adequate performance from the cognitive engine. Implementation in a variety of environments, discussed in Chapter 4, showed that decisions based on these formulas are unable to initiate a connection. The formulas optimistically return low performance estimations which the cognitive engine then uses to identify and rank fitness of potential solutions. These decisions typically suggest lower transmit powers and more aggressive modulations. Clearly, it is a problem that if the model is not good enough, that complete failure is possible. While operations would benefit from more realistic models, another strategy is to adapt existing models to return higher estimates of performance with the goal of influencing decisions towards more conservative selections.

3.4 New Calibration Technique

This section introduces what is believed to be a unique technique to improve GA operations in practice while leveraging existing formulas. The motivation behind this technique is that the current methods will not work unless the existing algorithms are adjusted for specific situations. The problem is that using basic estimation curves in over-the-air operations typically produces a BER estimation ($B\hat{E}R$) that is too low. In other words, the estimate is too optimistic. It is expected that algorithmic estimations would not match observed performance in environments with structured interference. Operationally, the goal is to influence the existing estimation formulas to return a more conservative estimate of BER that leads the engine to produce more conservative decisions. A simplistic solution is to artificially bias the estimation process by shifting the E_b/N_0 used to calculate an estimated BER to a lower value [3, 9]. Conversely, if the BER estimation happened to be higher than the observed than a similar shift to a higher SNR will return a lower estimated BER.

This technique considers what it would take to influence the value of SNR used when assessing the BER performance curves. This can be accomplished by considering the original definition of E_b/N_0

as a function of $(S - N)$ where $(S - N)$ is defined as the signal to noise ratio. The digital signal processing software, *liquid* DSP, is capable of returning a received signal strength indicator (RSSI) which is equated to the signal strength. Hence, SNR is shown in (3.9). Manipulating the value of the noise power (N) used in calculating SNR can effectively change the value of the returned estimated BER.

$$SNR = RSSI - N \tag{3.9}$$

We defined a bias term, α, that is equivalent to the inverse of N. Therefore, SNR can be written as $SNR = RSSI + \alpha$. An increase in α results in a increase in SNR which leads to a lower $B\hat{E}R$. Conversely, a decrease in α results in a decrease in SNR and an increase in the $B\hat{E}R$. One can iteratively adjust α in order to customize the $B\hat{E}R$ such that estimation of BER is closer to reality which yields better overall decisions. The same philosophy applies to PER which is used in implementation of the method given the higher reliability of measuring PER. One advantage of this technique, is that it is not necessary to know the value of the actual noise power in order to use a trial-error approach to identifying a good value for α. Manual adjsutment involves iteratively testing several values of α until an acceptable performance in terms of residual between estimated and observed performance values and final fitness of a solution is achieved. This adjusted model may not represent performance exactly, however it is better since it is adapted for a specific situation.

Figure 3.2 shows a hypothetical situation where α_1 is used to identify an $B\hat{E}R$. The returned estimated when using $SNR = RSSI + \alpha_1$ is compared to the observed BER. In this case, the observed BER (as designated by the star) is worse, or higher, than $B\hat{E}R$. In order to return a higher $B\hat{E}R$, the adjustment factor needs to be decreased. If α_2 were used instead of α_1 then the returned $B\hat{E}R$ would be closer to the original observed BER. Conversely, consider Figure 3.3. Here, the observed BER with the adjustment factor set to α_1 is lower, or better, than the $B\hat{E}R$ returned from the model. Increasing the adjustment factor to α_2 would return a more realistic estimate.

3.4.1 Dynamic Adjustment Algorithm

Identification of a suitable α naturally lends itself to an automated adjustment routine. The goal is to find a setting for α such that the difference between the observed PER and $P\hat{E}R$ falls within a defined range. PER was selected as the decision metric given the higher reliability of measuring it

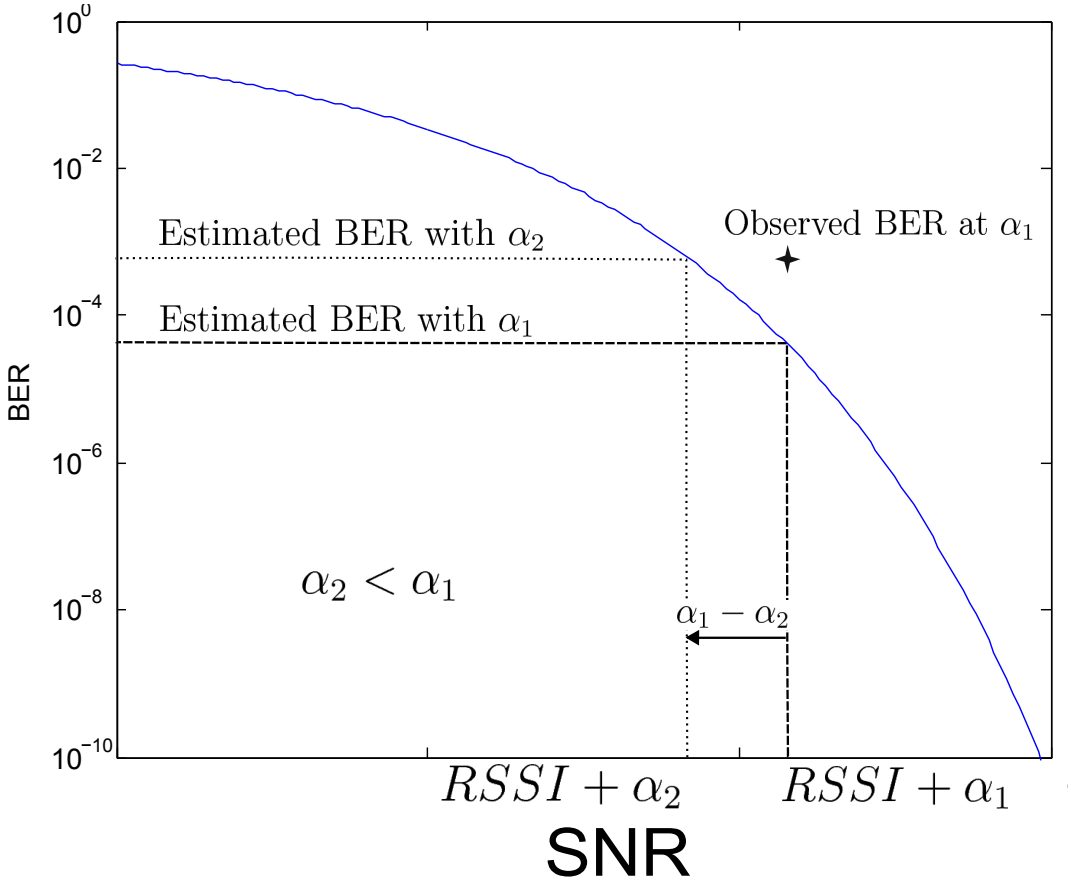


Figure 3.2: Hypothetical situation where the observed BER is worse than the estimated BER when using an adjustment factor of α_1. Decreasing the adjustment factor to α_2 returns a more realistic estimate.

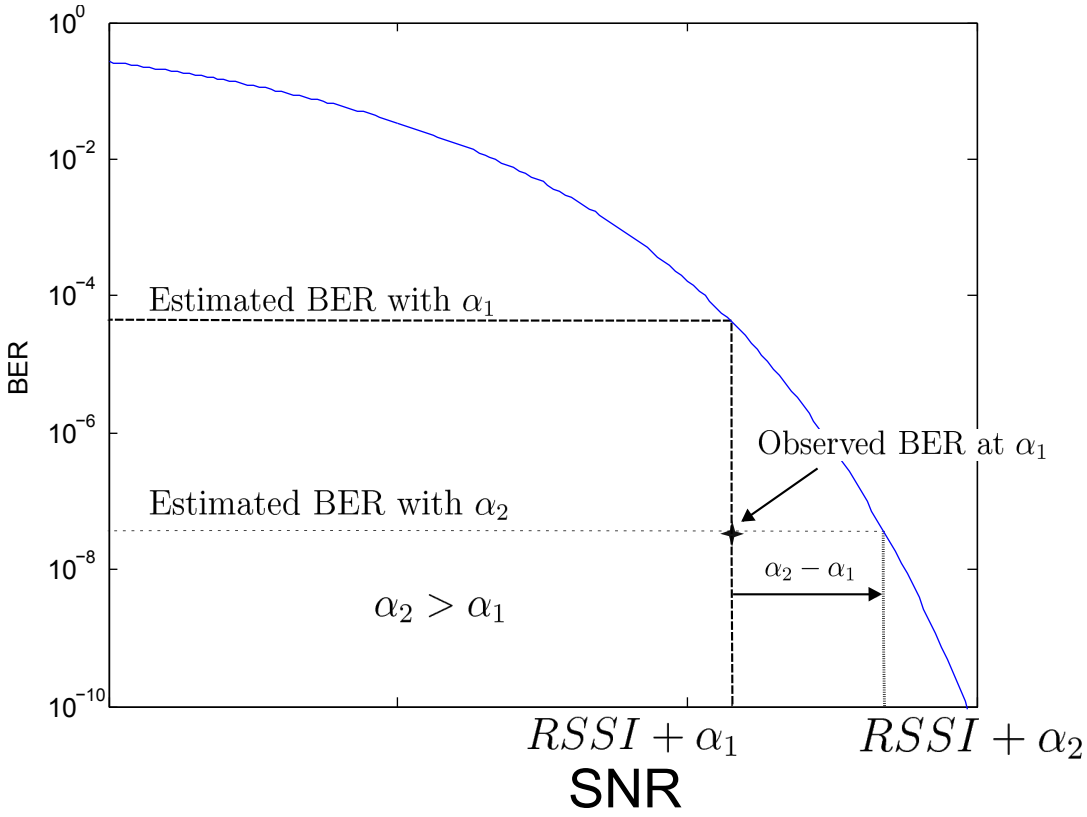


Figure 3.3: Hypothetical situation where the observed BER is better than the estimated BER when using an adjustment factor of α_1. Increasing the adjustment factor to α_2 returns a more realistic estimate.

by verifying CRC checks. A rudimentary binary search was selected for its simplicity and capability of identifying a good selection for α within $\log_2(N)$ iterations where N is the potential number of values that α can take which is defined in the form of a single array. The algorithm initially sets α to the value in the middle of the array. The GA then proceeds with normal operation where performance estimations are made using the current value of α. The GA identifies a selection of input parameters based on calculation of an overall fitness and then implements the decision. After the decision is made, the observed PER is compared to the $P\hat{E}R$.

Each iteration, the algorithm needs to determine two things: 1) whether the difference between the observed PER and $P\hat{E}R$ is within a pre-defined range, and 2) whether the observed PER is higher or lower than the $P\hat{E}R$. If the observed PER is lower than the $P\hat{E}R$ then the algorithm selects an α halfway between the current value of α and the maximum value defined in the array. Conversely, if the observed PER is higher than the $P\hat{E}R$, then the algorithm moves in the opposite direction and selects an α halfway between the current value and the minimum value in the array. The algorithm continues in this manner until the difference between the $P\hat{E}R$ and observed PER falls within the pre-defined range, or there are no more available values in the selection array.

Figure 3.4 illustrates a potential scenario and the resulting selection of α during adjustment. Upon initiation of the adjustment routine, α is first set to the midpoint of available values. The observed PER after a decision is made is less than the $P\hat{E}R$ and the difference is greater than the defined range. Therefore, α is increased to halfway between the midpoint and the maximum. After a decision is identified based on the current α, the same conditions are observed and the value is increased to halfway between this current point and the maximum. However, after this change, the observed PER is greater than estimated and α is decreased. Finally, there are either no more options available or the difference between observed and $P\hat{E}R$ falls within specification and the adjsutment process ends. Pseudocode for the algorithm is shown below.

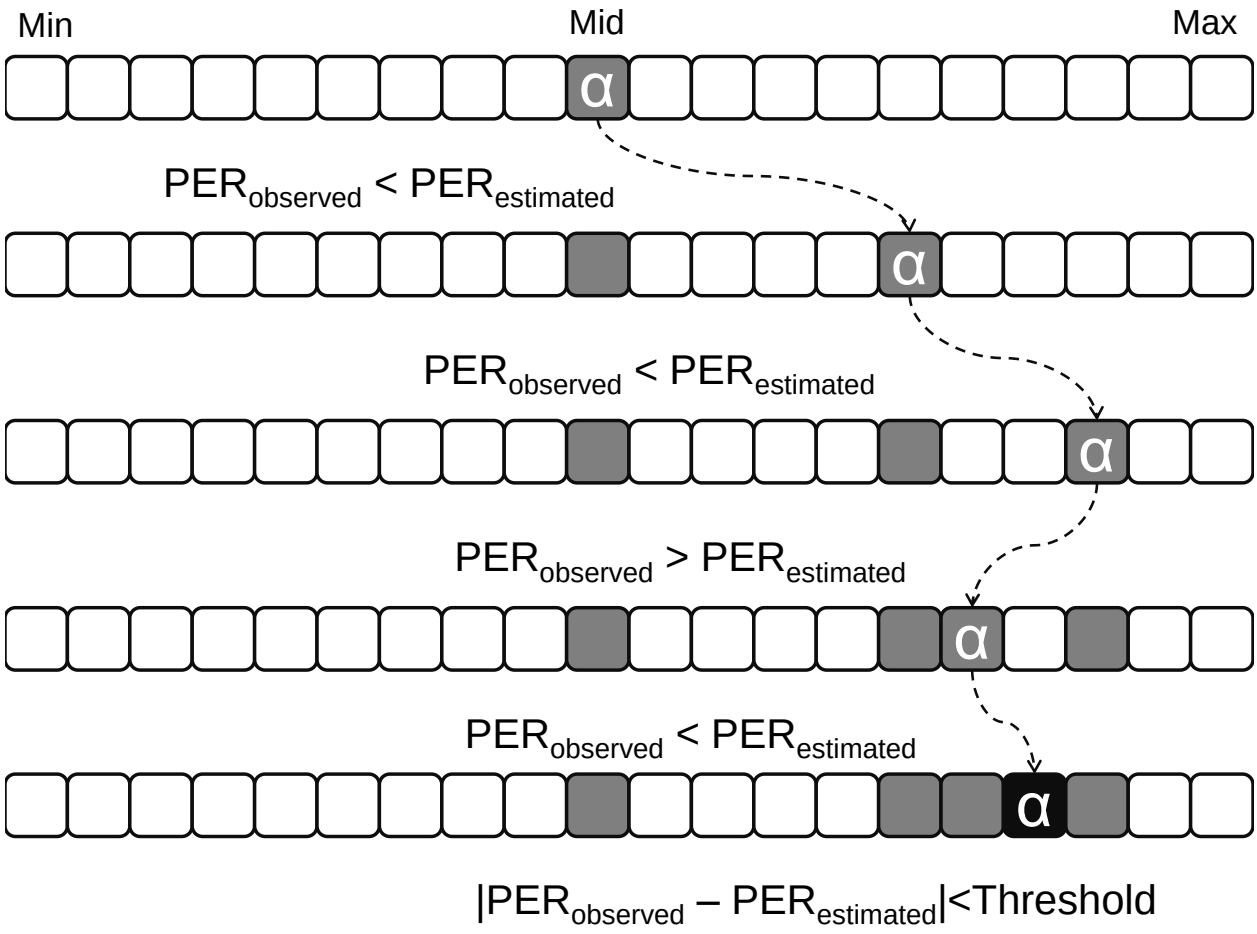


Figure 3.4: Example progression of dynamic adjustment algorithm

```
if GA is triggered
  if abs(PER_observed - PER_estimated) < PER_Range and iterations <6
     %% Increasing alpha
        last_good_alpha=current_alpha
        current_alpha=(last_bad_alpha - last_good_alpha)/2 + last_good_alpha
            if alpha already evaluated
             exit adjustment procedure
         else
         continue to GA to identify new input settings and collect new observations
         return to top
           end
  else
      if already in adjustment mode
           %%decrease alpha
         last_bad_alpha=current_alpha
         current_alpha=(last_bad_alpha - last_good_alpha)/2 + last_good_alpha
            if alpha already evaluated
             exit adjustment procedure
           else
             continue to GA to identify new input settings and collect new observations
             return to top
            end
      else
          if abs(PER_observed - PER_estimated) > PER_Range
              enter adjustment mode
              iteration =iteration+1
              set current_alpha=(alpha_max-alpha_min)/2  % set to middle of possible range
              last_good_alpha=alpha_min % initialize boundaries
              last_bad_alpha=alpha_max
          else
            exit adjustment
      end
   end
end
```

The range for α is selected such that ambient conditions, with no noise and interference, yield a value at the higher end of the range. Hence, as noise or interference appear, there is a larger range to adjust α. Experience with the USRP radios in several environments identified a range between [0,64] to perform well in a variety of environments. This range enabled the binary search to converge on a suitable selection of α in six iterations where $\log_2(64) = 6$.

3.4.2 Examples of Adjustment Procedure

Figure 3.5 illustrates the progression of the adjustment algorithm by comparing the value of α at each iteration to the corresponding difference in PER between observed and estimated. This example was taken from an ambient environment. During the first iteration, α was set to 32 and the corresponding difference in PER was negative because the observed value was lower than the estimated value. To correct this, the algorithm increases α and reruns the GA. After the GA's decision is implemented, the difference in PER between observed and estimated is still below 0. Therefore, the procedure continues to increase α. This repeats until the difference between observed and estimated falls within specification.

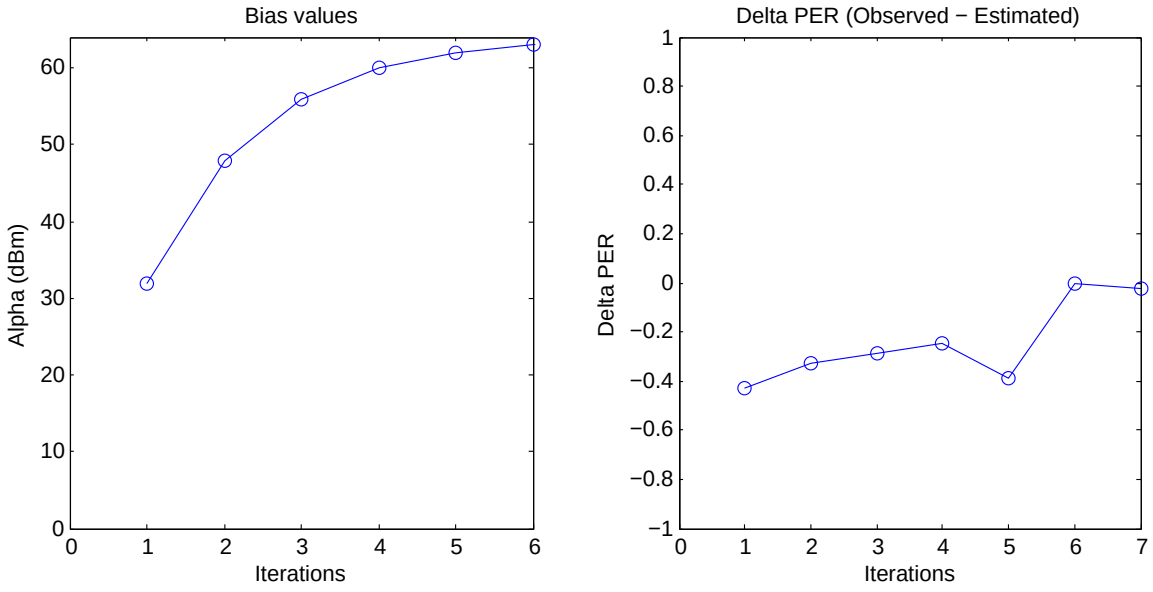


Figure 3.5: Example change in α and corresponding ΔPER

Another example is shown from the perspective of the software gain decisions made by the GA during the adjustment procedure. Figure 3.6 was taken in ambient outdoor environment. The range for α was set to [0,64], and the acceptable difference between observed and $P\hat{E}R$ is 5%. The initial software gain was set low and the engine's first decision was to increase power to maximum. At this point the adjustment routine engages by setting $\alpha = 32$ and continuing through the GA

decision process. This resulted in a small decrease in software gain. At this point, the observed PER is lower than estimated. This indicates that the decision made by the GA is actually too aggressive. The high software gain selected when $\alpha = 32$ is not necessary. The algorithm increases α to 48 which is halfway between the current value of 32 and the maximum value of 64. The GA's selects to decrease software gain, however the difference between observed and $P\hat{E}R$ still exceeds the defined range of 5%. This behavior is repeated the next three iterations and α is subsuquently increased to 56, 60 and 62 leading to corresponding decreases in software gain. However, at this point the observed PER was greater than $P\hat{E}R$. Therefore, the next selection was to decrease α to 61 which was also the last available value between the current bounds.

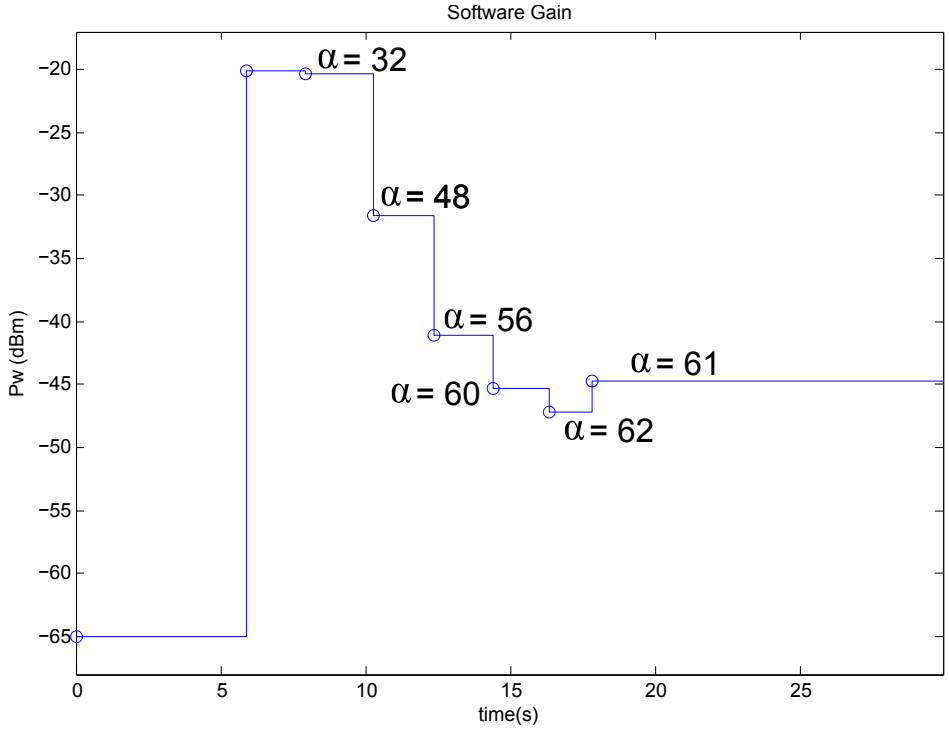


Figure 3.6: Software gain decisions made during adjustment process

Section 4.4 describes several conditions in which the cognitive engine was tested in a lab environment. Table 3.1 lists the environment and the corresponding α identified after six iterations of the adjustment routine. Results indicate that the algorithm identified a lower setting for as the interference conditions increased. A dynamic interference condition, which consisted of randomly moving a narrow-band interference signal, resulted in an α value similar to the wide-band inter-

ference condition. This resulted in slightly diminished overall fitness performance as detailed in Chapter 4.

Table 3.1: Calibration settings for varied environments

Environment	α (dBm)
Ambient	63
Wide-band interference	39
Narrow-band interference	35
Dynamic narrow-band interference	39
Artificial Fading	61

3.4.3 Limitations of Adjustment Algorithm

Use of the algorithm in practice revealed several limitations. The algorithm exhausts the possible range of values after six iterations. There may be better solutions available if there were more resolution to the range of α. However there is a trade off in how much improvement might be gained and taking the extra time to continue the procedure. The process flow was designed to force the procedure to exit after a fixed number of iterations. This decouples the procedure from normal operations to ensure that it is not called if packet error estimations degrade. This too is a trade off as a balance is required between stopping operations in order to recalibrate. Dynamic conditions create a limitation for the GA, as a whole, and this adjustment process. It was decided to perform the adjustment once in a specific environment and then proceed through testing rather then recalibrate continually even though the level of α may not be the ideal.

3.5 Framework for Statistical Response Profiling

In summary, there are limitations to the current approach of using implicit algorithms to estimate performance. Ideally, one could draw upon theoretical models that better matched the situation. However, it is difficult to identify the channel type in order to apply more customized algorithms [7]. Practical improvements can be obtained by modifying existing models through adjustment procedures. Nevertheless, the process requires past experience and must be performed in every new situation.

In light of these hurdles, there is potential for an experimental design approach to compliment the existing strategies. The experimental techniques do not require knowledge of the channel conditions, or past experience with the environment. Performance models developed from statistical approaches are founded upon empirical evidence and not theoretical formulas. Therefore, experimental design has potential to support new and unique methods for identifying solutions in cognitive radio applications. This section provides a guide for incorporating statistical testing methodologies within communications system. The framework is intended to be independent of application space and radio platform. First a statistical screening approach is introduced as a tool to filter out insignificant input parameters. Next, response surface methods provide empirically generated estimation models for identifying recommended parameter settings. An example of implementation of a Box-Behnken and central composite design (CCD) is presented where a single metric is used for evaluation for simplicity. Finally, the Taguchi method is applied to the same wireless link to compare to the RSM methods.

3.5.1 Defining the Input and Output Bounds

The first step defines the specific system based on controllable input parameters, measurable outputs, and a specific environment. Consider m configurable parameters denoted by x and n measured outputs denoted by y. A key aspect of an empirical approach is that the results are environment specific, hence ϕ is indicated as the environment. Once inputs and outputs have been identified, the next step is to define the concept of operations under which the system is bounded. These bounds include the minimum and maximum values that each controllable parameter can take. The typical approach defines minimum, maximum, and a nominal values: $\min(x_i)$, $\max(x_i)$ and $\text{nom}(x_i)$. These definitions provide the basis for normalizing the parameters onto a scale of [-1,0,1].

Note that DOE and RSM assumes that input parameters are continuous. This is straightforward with parameters, such as transmit power and packet size. Parameters, such as modulation and coding, require the user to create an arbitrary translation onto the encoded scale of [-1,1] [39]. This requires ordering the available choices along some continuum.

Attention is now turned to the outputs. In order to define good or bad performance, the user must define the goals of each output. In general, this is performed by matching each metric against the

basic goals of higher is better (HIB), or lower is better (LIB). This fundamental definition drives the overall shape of the utility function and whether it is increasing or decreasing. These functions normalize output metrics onto similar scales in order to aggregate them into a combined value for defining overall performance. These basic definitions drive the formation of either utility functions [7]or desirability functions, as described in Section 2.12.2.

3.5.2 Screening

With the system configuration parameters and the output metrics defined, a statistical screening is performed, as indicated in Figure 3.7, to identify the configuration parameters that have the most bearing to the outputs. One goal of the screening is to identify insignificant parameters so they can be dropped in order to simplify future testing and the estimation models. The screening experiment involves use of a basic statistical design such as a 2^m factorial design, as detailed in Section 2.3. In summary, a factorial design consists of a set of individual tests. Each test consists of a unique combination of input parameter settings encoded onto the [-1,+1] scale. For example, a system consisting of a two configuration parameter system might have a test defined as [-1,-1]. This translates to both the configuration parameters being set to their minimum. A full factorial test would include every possible combination of the minimum and maximum of the input parameters.

To identify factor significance, an ANOVA process, as described in Section 2.8.1, is followed. This analysis identifies the statistical significance of each main configuration parameter in conjunction with each output. In addition, the interaction effects of combinations of parameters and their significance is quantified. This is an important issue, as the main parameter might not produce significant affect on an output. However, changing it in combination with another parameter may induce significant effect. Therefore, that main parameter should still be evaluated in further testing. With significant and insignificant factors identified, one can move on to identifying estimation models of the output.

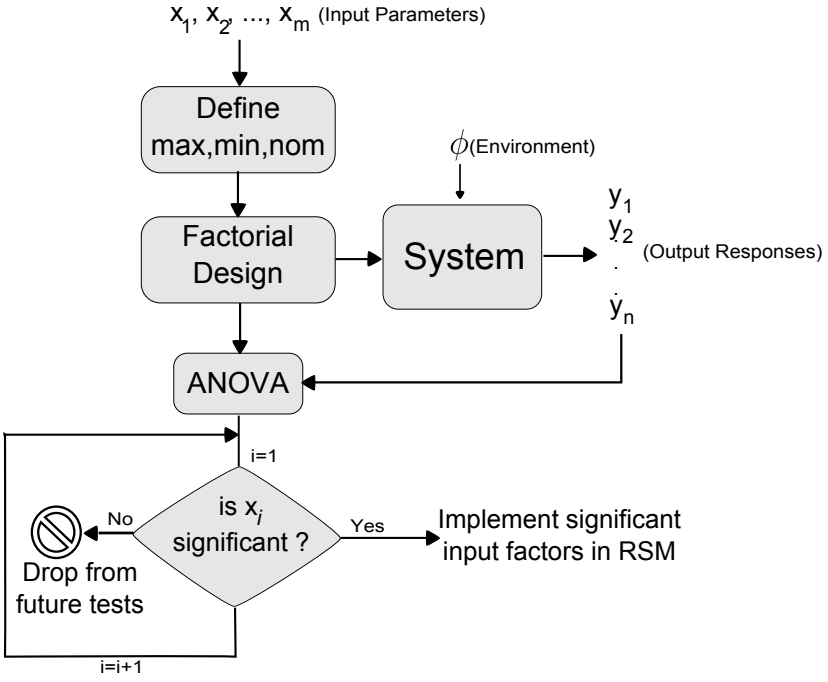


Figure 3.7: Screening Flowchart

3.5.3 Framework for Response Surface Methodology

At this point, one could take the result of the factorial design and follow a linear regression and least squares estimation to identify a linear model of the output profiles. Solutions for these models can be found and implemented on the radios at this time. However, RSM models create more realistic estimations of behavior as they incorporate quadratic model and curvature of the output.

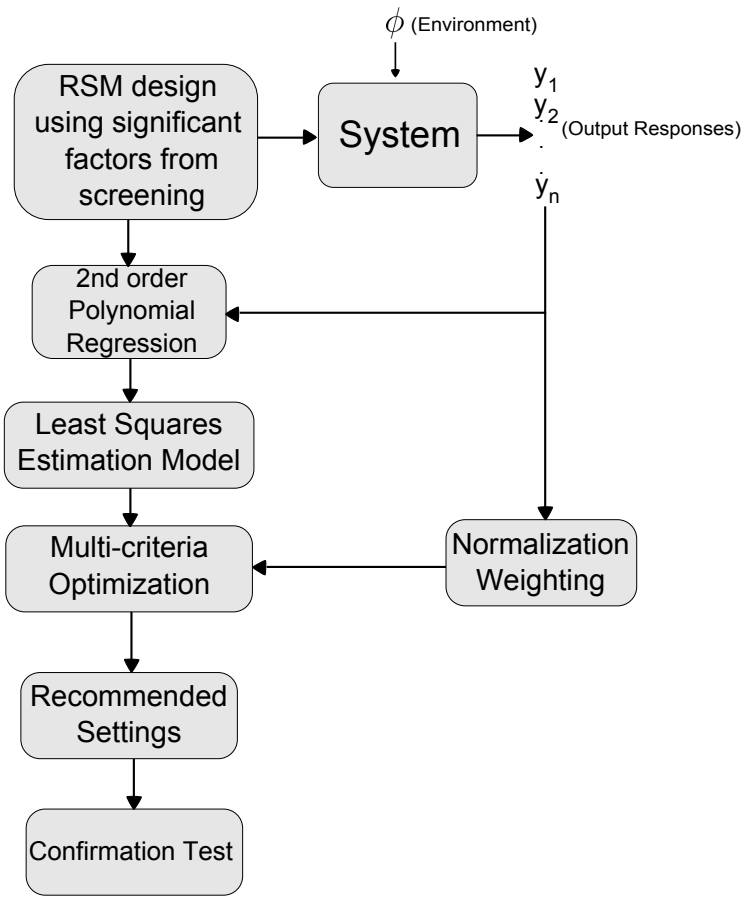


Figure 3.8: Flowchart of Statistical Framework

The RSM process, as detailed in Section 2.5, follows the basic steps of DOE by implementing a specific design across the system. The results are processed through 2nd order polynomial regression and least squares estimation of model coefficients. The result is an estimation model of each output as a factor of each configuration parameter, as described by (2.22). At this point, some method of multi-criteria optimization is required.

Recall that the GA approach also begins with models of system behavior. At this point the procedure for solving the statistical models can follow a GA approach. The core difference is where the performance models came from. The empirical approach developed the performance models based solely on experimental data collected by exercising the system with defined inputs. Regardless of the specific method, the goal is to identify a recommendation for the configuration parameters that yields a good solution as originally defined by the user defined goals for each output. The solution is comprised of an array of the suggested configuration parameters. It is unlikely that this set of configurations was one of the actual tests performed in the design. Therefore, it is essential to implement this set of parameters across the system. This final test is often called a 'confirmation test'.

3.6 Applying Experimental Design to Wireless Application

This section illustrates examples of applying the statistical techniques on a wireless platform. First, the designs are implemented on a simulated wireless links in order to validate design selection and model complexity related to (2.22). Next, an over-the-air implementation of statistical designs is performed where one output metric is considered. The Box-Behnken and CCD RSM methods are performed followed by a Taguchi design. Each is used to identify a recommended input parameter setting that maximizes the goodput across a wireless link. The system model used is described in detail in Section 4.2 where a wireless file transfer is sent between a single transmitter / receiver chain without artificial interference. Note that the current software digital signal processing used for file transfer does not include the capability for retransmission of packets.

3.6.1 Simulation setup

Two simulation platforms were developed to verify the use of RSM estimation models. The first platform used the *liquid*-digital signal processing simulation capabilities [44]. The *liquid* open-source library contains all the necessary processing elements for transmitting signals across a USRP wireless link. In the simulation mode of operations, all of the same steps from encoding, modulation, decoding and demodulation are performed exactly as if the signal was going to be sent over the air. Table 3.2 lists the configuration parameters for the simulation that are defined within *liquid*. Transmit power, modulation, coding, and packet size are all variable and defined for each specific test within the experimental design.

To enable greater flexibility and transition of the research, a basic simulation framework was also implemented on MATLAB. In the similar fashion, each row of a design defines a unique configuration of input parameters prior to simulating a transmitted signal on channel with additive white Gaussian noise (AWGN). In the future more complicated channel environments can be added to the simulation. The transmitted signal consisted of a random generation of bits for each packet with a total of 1000 packets. Based on the defined coding and modulation, each packet was encoded and modulated and then combined with noise. Bit errors and subsequent packet errors were calculated after demodulating and decoding the noisy signal.

3.6.2 Simulation Results

ANOVA performed on the simulated implementation of the Box Behnken and CCD RSM designs show that several second-order and quadratic terms are statistically significant to the output metrics. Similarly, the R^2 measure of fit is presented for three estimation models shown in (3.10), (3.11), and (3.12). The simplest model considers only main terms, while the next incorporates cross terms, and the final model includes squared main terms.

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i \tag{3.10}$$

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum \sum_{i<j} \beta_{ij} x_i x_j \tag{3.11}$$

Table 3.2: *liquid* dsp simulation configuration parameters

Configuration	Setting
total subcarriers	28
null	6
pilot	6
data	16
cyclic prefix length	16
CRC scheme	16 bit
frame assembled	yes
payload decoded bytes	20
payload encoded bytes	22
Total OFDM symbols	28
S0 symbols	3@28
S1 symbols	1@44
header symbols	18@44
payload symbols	6@44
bandwidth	200kHz

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum \sum_{i<j} \beta_{ij} x_i x_j \tag{3.12}$$

Table 3.3 shows the coefficients for estimation models of PER developed using CCD and Box Behnken RSM designs implemented in a MATLAB simulation. The three main input parameters are transmit power (Tx Power), modulation and coding (M/C), and packet size. In this case modulation and coding were changed together as a pair to reduce the number of input variables to three. In addition to the estimate of the coefficient for each term, the statistical significance of each term is indicated by the $p - value$. Values $< .05$ indicate that the terms of statistically significant to the output. These results suggest that some cross terms and squared terms are significant to the output. This validates the need to pursue designs and estimation models that incorporate these terms. Similarly, Table 3.4 lists the coefficients developed from a four variable implementation of RSM designs in simulation using *liquid*-dsp. In this case, modulation and coding were considered as separate input parameters. This table also shows that several cross terms and squared terms are considered statistically significant to the estimation of throughput.

Further justification for developing estimation models based on (3.12) is given by comparing the R^2 values from models of varying complexity. Table 3.5 and Figure 3.9 show the R^2 values of model

Table 3.3: Three Input Variable Matlab Simulation Results for PER

	CCD		BB	
Term	Estimate	p-value	Estimate	p-value
Intercept	0.202	$< .0001$	0.000	1.000
Tx Power	-0.298	$< .0001$	-0.306	$< .0001$
M/C	0.297	$< .0001$	0.311	$< .0001$
Packet Size	0.001	0.981	0.133	$< .0001$
Tx Power*M/C	-0.249	$< .0001$	-0.250	$< .0001$
Tx Power*Packet Size	-0.001	0.974	-0.138	$< .0001$
M/C*Packet Size	0.001	0.978	0.128	0.000
Tx Power*Tx Power	0.189	0.002	0.120	0.001
M/C*M/C	0.186	0.003	0.130	0.000
Packet Size*Packet Size	-0.304	$< .0001$	0.243	$< .0001$

Table 3.4: Four Variable *liquid*-DSP Simulation Results for Throughput

	CCD		BB	
Term	Estimate	p-value	Estimate	p-value
Intercept	5.85E+08	$< .0001$	5.80E+08	$< .0001$
TxPower	2.21E-09	1	0	1
Mod	2.87E+08	$< .0001$	3.15E+08	$< .0001$
Coding	1.07E+08	$< .0001$	9.17E+07	$< .0001$
Packet Size	-3.20E+08	$< .0001$	-3.41E+08	$< .0001$
TxPower*Mod	-5.71E-08	1	0	1
TxPower*Coding	0	1	0	1
Mod*Coding	7.70E+07	$< .0001$	7.26E+07	$< .0001$
TxPower*Packet Size	-2.48E-09	1	0	1
Mod*Packet Size	-2.27E+08	$< .0001$	-2.88E+08	$< .0001$
Coding*Packet Size	-8.6E+07	$< .0001$	-7.48E+07	$< .0001$
TxPower*TxPower	-6.03E+06	0.7696	2.93E+06	0.818
Mod*Mod	6.00E+07	0.0048	6.45E+07	$< .0001$
Coding*Coding	-9E+07	$< .0001$	-9.28E+07	$< .0001$
Packet Size*Packet Size	3.36E+07	0.1067	4.14E+07	0.0017

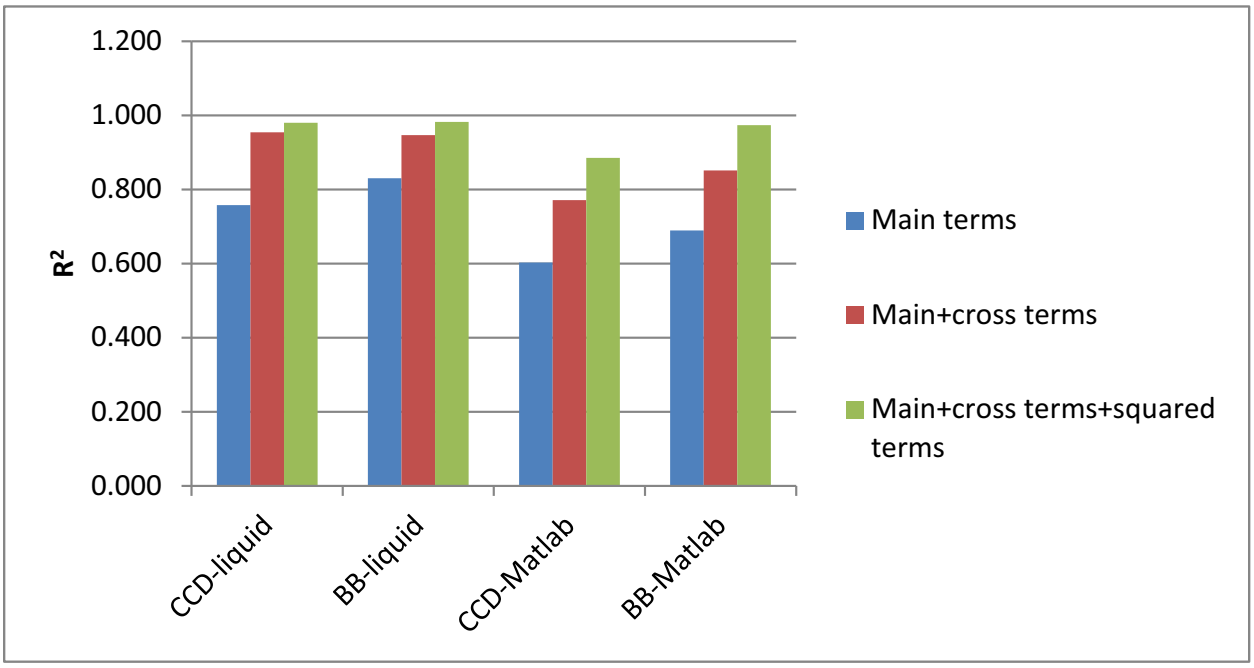


Figure 3.9: R^2 values from models of varying complexity

fit obtained from simulation results in both liquid and Matlab platforms. The results indicate that a model that is constrained to just main terms has limited statistical fit. As additional terms are added to the model, R^2 improves. The statistical fit is highest for models that include both cross terms and squared terms. This further justifies the use of RSM designs.

Table 3.5: R^2 measures of statistical fit

	CCD-liquid	BB-liquid	CCD-Matlab	BB-Matlab
Main terms	0.758	0.830	0.603	0.690
Main+cross terms	0.954	0.947	0.772	0.851
Main+cross terms+squared terms	0.980	0.982	0.885	0.974

3.6.3 Over-the-air Box-Behnken and CCD Goodput Example

Table 2.4 lists the designs for the Box-Behnken and the CCD. Each column represents an input parameter that is changed during each tests. One of the key difference between the designs is the exclusion of vertice points in the Box-Behnken design. Each design was implemented on the wireless link, where a data file was sent across during each test. The goodput (bps), as defined by the total bits contained in payload packets that passed cyclic redundancy checks, was recorded. Following the analysis defined in Section 2.8.2, estimation models for goodput were developed for each design as listed in Table 3.6. These terms correspond to estimations of the coefficients for (2.22) which defines a performance model. The *intercept*, or β_0 term, corresponds to the value of the model if all inputs are set to zero.

The surface profiles from both, are shown in Figures 3.10 and 3.11. These figures illustrate how both methods show the general performance trend for the metric. With a goal of maximizing goodput, both models point to a solution provided by transmit power, modulation/coding (M/C), and packet size all equal to the encoded value of 1, as summarized in Table 3.8. However, the CCD surface profile shows that goodput drops when transmit power is low and packet size is high. This was determined because the CCD incorporates observation points at the vertices of the design space. Therefore, it has higher accuracy for estimation in those regions. In contrast, the Box-Behnken was unable to capture that degradation in goodput in that region. Therefore, there is a trade off in overall accuracy across the entire design space versus just being able maximize the final result.

3.6.4 Taguchi Goodput Example

The Taguchi method differs from the RSM methods in several aspects, as discussed in Section 2.6. First, the approach only considers discrete solutions. Therefore, the convention for defining the designs uses nomenclature to indicate a specific discrete level of an input variable. Typically, a design would be listed in terms of designated levels, such as “1”, “2”, and “3”. These correspond to the RSM coded values of [-1,0,+1]. In practice, the nomenclature of Taguchi designs does not follow the normalized [-1,+1] scale. Table 3.7 shows the $L9$ design and the observed values of goodput

Table 3.6: Estimation of goodput model coefficients

Term	β	BB Estimate	CCD Estimate
Intercept	β_0	280058	297603
Tx Power	β_1	44913	65924
M/C	β_2	83475	79551
Packet Size	β_3	134231	97313
Tx Power*M/C	β_{12}	90331	82509
Tx Power*Packet Size	β_{13}	-218	76767
M/C*Packet Size	β_{23}	82154	50646
Tx Power*Tx Power	β_{11}	-30742	-23177
M/C*M/C	β_{22}	-73984	-47589
Packet Size*Packet Size	β_{33}	-23957	-79975

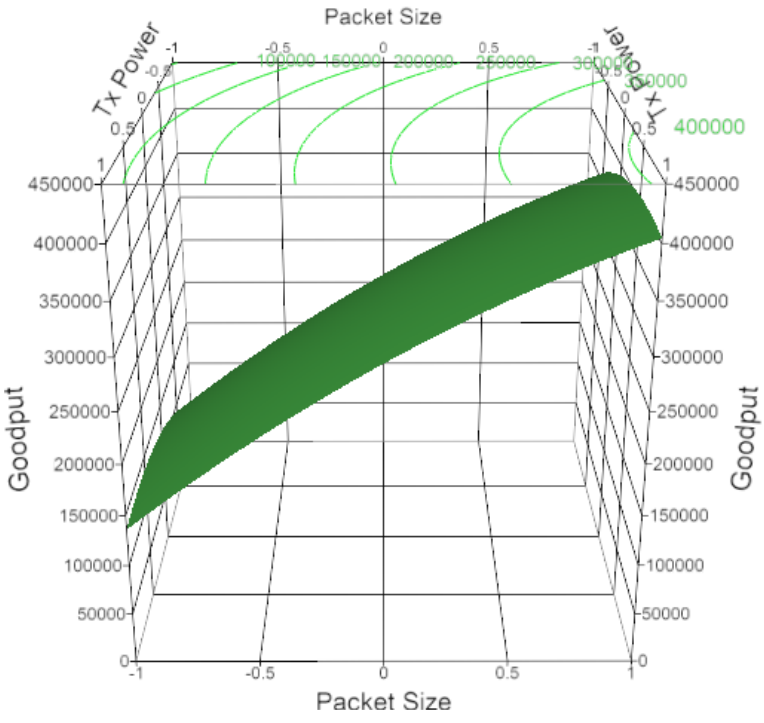


Figure 3.10: Surface profile of goodput in Box-Behnken design

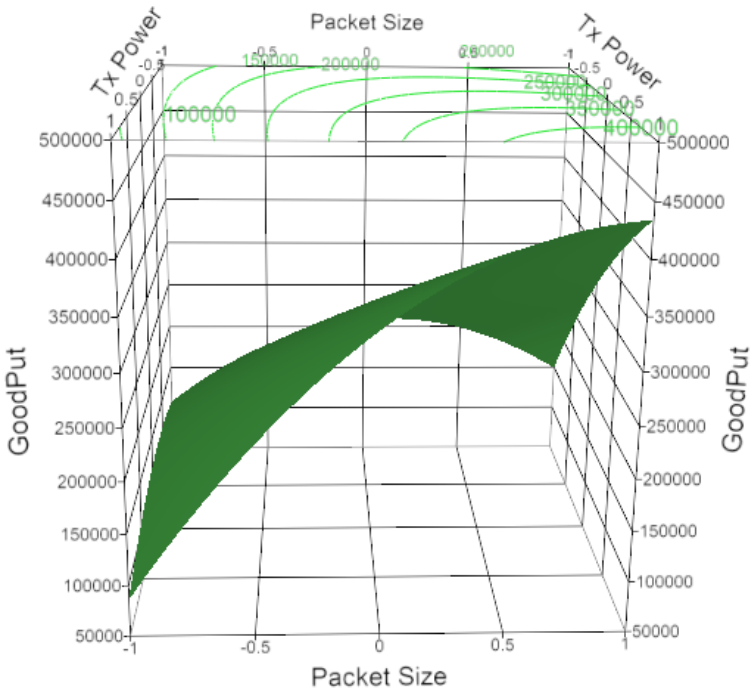


Figure 3.11: Surface profile of goodput in CCD

from four repetitions of the design and the mean for each row of the design. Because of the focus on discrete values, there is no continuous estimation model developed as with RSM. Instead, the approach develops a simplistic model of the output as only a function of the inputs and does not incorporate any interaction terms or quadratic terms. Any solutions identified will be from some combination of the discrete levels initially defined.

Second, the Taguchi method considers variation of metrics around a mean as an additional performance metric. Therefore, the table lists the observed goodput values for each run, the mean, and the Taguchi signal-to-noise-ratio as defined in Section 2.6.1. The Taguchi-SNR is considered similar to any other metric as described in Section 3.2. Therefore, the mean of the response under consideration and Taguchi-SNR are converted to utility values, weighted and combined into a single fitness function in order to identify a solution. A solution to this Taguchi example is one that maximizes both the mean of goodput and the Taguchi-SNR. Figure 3.12 illustrates the JMP prediction profiler, which is an interactive tool to identify solutions from multiple metrics and input parameters. The first three columns represent the three input variables: transmit power,

Table 3.7: Raw Data from Taguchi Example

TxPower	M-C	Packet Size	Run 1	Run 2	Run 3	Run 4	Mean	Tag-SNR
1	1	1	50473	50339	50493	50413	50429	94
1	2	2	373231	371264	377743	377040	374820	111
1	3	3	13124	53854	13449	13412	23460	83
2	1	2	106640	106723	107264	107125	106938	100
2	2	3	374322	373626	374573	365740	372065	111
2	3	1	78953	81648	82967	82926	81624	98
3	1	3	377487	377472	373779	377574	376578	111
3	2	1	74435	74516	74196	74340	74372	97
3	3	2	409174	423492	424616	413898	417795	112

modulation/coding, and packet size. The last column shows simple utility functions that define the objective of maximizing both the goodput mean and Taguchi-SNR. A direct search of the available discrete values of the input parameters yields a recommendation for input settings that achieve the defined goal.

3.6.5 Comparison

The recommended solutions for maximizing goodput from each design were implemented directly across the system model. The resulting goodput provided a basic single-metric frame of reference for comparing these designs. These are called confirmation tests in order to empirically confirm the decisions. Table 3.8 lists the decisions made and resulting goodput. Both the Box-Behnken and CCD recommended maximum transmit power, the most aggressive modulation, and largest packet size. The Taguchi method recommended a more conservative modulation and coding and smaller packet size. The reason behind this was because of the limited observation points of the Taguchi array combined with variation in the observations. For example, the third row of the $L9$ array set the transmit power to the lowest level, and modulation and packet size to the highest. There was more variation in the observations because this combination is not ideal for the wireless link. Hence, the overall Taguchi-SNR went down. This affected the final decisions by decreasing the overall fitness of a decision with higher modulation. The same issue applied to packet size. Therefore, the final decision by the Taguchi design was more conservative compared to the RSM designs. Figure 3.13 compares the performance of the RSM and Taguchi designs in terms of final

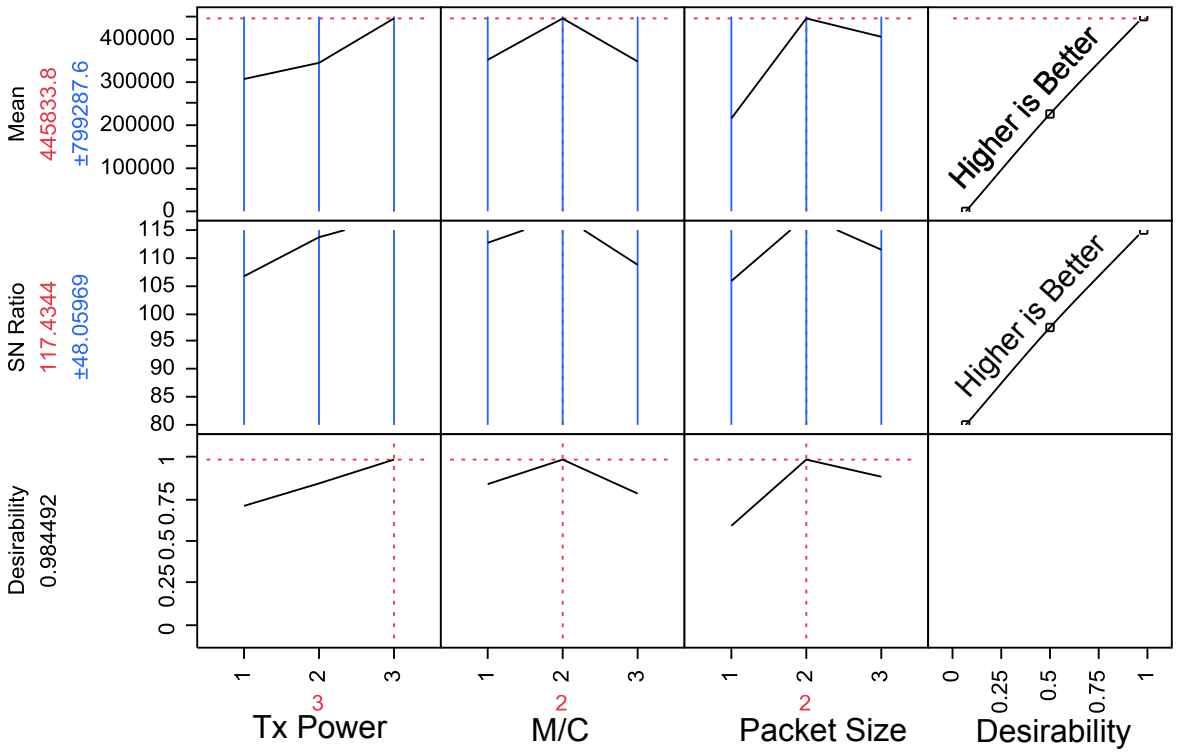


Figure 3.12: Profiler of Taguchi for goodput

goodput and the number of tests required in the design. In this case, there was a significant trade off in decision quality compared to number of test required.

Table 3.8: Recommended Solutions from Example Designs

Method	Software Gain (dB)	Modulation	Coding	Packet size	Goodput (bps)
BB	-19.5	64-QAM	No Coding	300	632304
CCD	-19.5	64-QAM	No Coding	300	674211
Taguchi	-19.5	16-QAM	Hamming (7,4)	140	281464

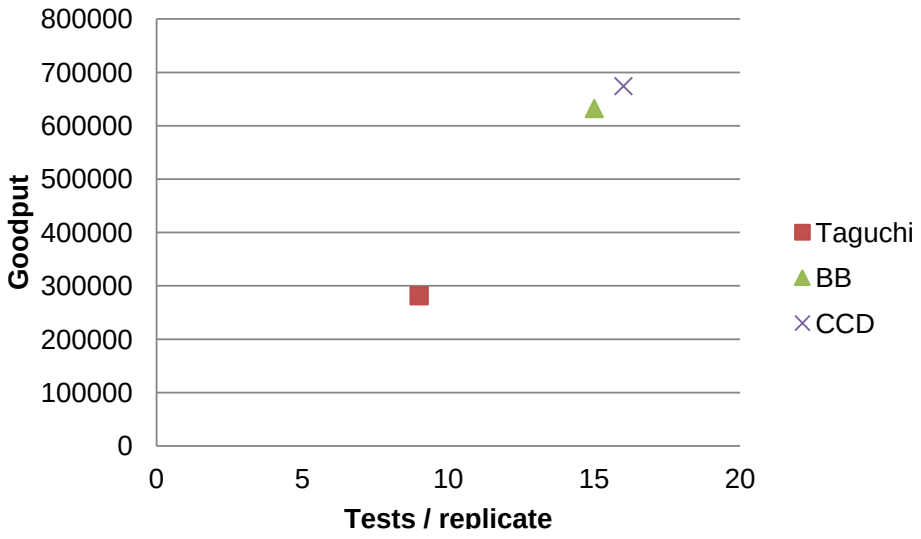


Figure 3.13: Comparison between RSM and Taguchi for goodput

3.7 Drawbacks and Limitations Statistical Experimental Design

The statistical methods described show the capability to model system behavior through a limited number of observed data points. There are drawbacks and limitations that require consideration when implementing these methods. These include lack of formal proof for optimality from the model, assumptions associated with multiple linear regression, potential for anomalies associated with configurations outside of the tested data points, extending results outside of the environmental conditions in which the design was run, and difficulties in scaling the methods to larger distributed systems.

3.7.1 Limitations of Factorial and RSM Designs

- **Domain knowledge required:** One of the keys for applying DOE and RSM to any system is some level of domain knowledge [45]. The user requires an understanding of how to define the minimum and maximum ranges for each parameter. Additionally, the methods are valid for quantitative factors that are placed on a numerical scale. Dummy variables can represent qualitative factors only if they can be arranged in some relative order of magnitude.

- **No learning capability:** Unlike CBR, there is no learning capability in the empirical

methodology. The developed estimation models are tied to the environment in which they were collected. GA suffer this limitation as well. The statistical methods are suitable to augment learning by initializing or feeding the new results into a case base. This can serve as a training tool for CBR.

- **Solutions not formally optimal:** Models developed by RSM and related methods such as DOE provide an estimation of system behavior. These models can feed into techniques, such as method of steepest ascent and multivariate optimization with desirability functions, which provide recommendations for input settings corresponding with favorable outputs. These solutions are not optimal in the formal sense and should be considered as tools for identifying behavior trends and general ranges for factor settings. This issue is often faced within cognitive radio. For example, GAs are commonly implemented for decision making, however limitations on how long they can run limit the true optimality of their output.

- **Always trade offs in arriving at the solution:** There are two steps in the optimization process that have limitations. The first is the normalization process that places all the outputs onto some type of similar scale, such as the use desirability functions as described in Section 2.12.2. There will always be inherit trade-offs in selecting the functions and weights [10]. The second step is the choice of optimization method. There are many choices available, again all with trade offs. Heuristics, such as the GA, have seen extensive use in cognitive radio. The Nelder Simplex method is another optimization method often utilized in analysis of RSM [46]. The simplex method offers an alternative to heuristics when improvement is emphasized more than optimization. The algorithm can achieve improvements in a relatively small number of iterations.

- **Difficulty with discrete parameters:** The RSM approach is designed with continuous input parameters in mind. For nearly continuous parameters, such as transmit power and packet size, it is fairly easy to round to the nearest discrete value. However, with qualitative discrete parameters, such as modulation and coding, there is always an arbitrary definition of how to encode the elements. This is an area requiring further research to identify better combinations of modulations and codings to use as discrete values in the empirical design.

- **Environmental specific results:** The outputs measured during implementation of the

design are tied to the environmental conditions in which the tests occurred. For example, Chapter 4 performs multiple replicates of designs within a fixed and somewhat controllable environmental situation and while the results do show consistency across these conditions, the specific results are not generalizable. The methodologies can be applied in general, but the results are specific to the environment in which the test was conducted. This criticism is also levied against other experiential-based decision making, such as case based reasoning (CBR). CBR cognitive radios may not be able to adapt to a new environmental situation without having experienced it previously.

- **Cannot predict anomalies:** While the results often show strong statistical fit, as indicated in Section 4, the model cannot predict highly anomalous outputs which could result from certain unique combinations of factors. This is a trade-off that must be considered in exchange for efficient designs. One of the most important benefits of statistical methods is their capability to draw conclusions on system behavior from a limited number of test cases. This efficiency is of great benefit in situations where testing and system implementation incur high costs of manpower and resources.

- **Multiple regression assumptions:** Analysis of RSM and DOE statistical designs utilize multiple linear regression to draw statistical conclusions. Success of these techniques requires certain assumptions of the relationships between factor variables and on the measured data. First, the methods rely on linear relationship between the factor variables. In general, minor deviations can be tolerated and do not significantly impact results. When analyzing data, visual checks can quickly identify major violations. Graphs of the observed versus predicted values and residuals versus predicted readily require symmetry around a diagonal line in the former and around the horizontal axis in the latter. This might not be feasible in real-time radio control, but could be used to diagnose network issues in non-real time. The literature supports the use of these methods in complex systems, such as wireless communications, and wireless networking where these linear relationships between factors can be questionable. The second assumption focuses on the distribution of the errors between the observed data and the estimated data. The distribution of these residuals is assumed to fit a normal distribution. This is due to the least squares estimation procedure which drops the error term from the

estimation model under this assumption.

- **Trade offs on large variable systems:** One of the benefits of the statistical methods is efficient testing designs which can exercise a system without testing every combination of variables. The problem is that even a 2^k factorial design can grow unwieldy quickly. This can be mitigated somewhat through fractional factorial designs, but there will always exist a trade off between number of tests and information gain. While the initial factorial design may have a high cost, the goal is that some parameters will surface as insignificant and be dropped form further testing.

- **Solving the model incurs a cost:** In this implementation, model development and solving the model equations is performed off-line after data collection. The process flow and mathematics is straightforward and adaptable for automation given sufficient processing power [41]. However, there is a processing cost in each step and solving the model equations as a multi-variate optimization problem is virtually the same hurdle that GA's face. Methods warranting further exploration for improving the speed of identifying solutions to empirical models include non-linear programming methods or possibly field programmable arrays (FPGA) or graphics processing units (GPU) realization.

- **Potential limitations on scalability for complex systems:** The methodologies supported in the literature and in this paper are focused on straightforward system models that lend themselves to a 'black box' approach of controllable inputs and observable outputs. In question is the scalability of the methods to large cascading systems which can be considered as multiple black box systems. Considering that one system feeds into several others, questions arise as how to control the input parameters of all the systems as well as the timing of when to set the parameters. Future research should consider applying experimental design across multiple networks of SDR radios.

3.8 Summary

This chapter discussed limitations observed in operations of current cognitive architectures. Inaccuracies in estimation methods make it difficult identify correct decisions. An adjustment procedure

that leverages existing work provides a simplistic method for improving estimation accuracy. The procedure is further enhanced through a dynamic algorithm that uses a binary search capable of identifying a setting within less than six iterations. Experimental design techniques were introduced for wireless applications that address the need for knowledge of the environment and limitations in facing new environments. A generic framework provided guidelines for applying the techniques to a variety of systems and applications. A basic example using a single metric compared performance of RSM and Taguchi designs. Finally, the chapter reviews limitation and hurdles of the statistical methods.

Chapter 4

Real-world Testing

4.1 Introduction

A current architecture for decision making in cognitive radio is based on genetic algorithm (GA) optimization. The GA relies on estimation formulas in order to calculate the potential success of a solution generated during the heuristic process. Theoretical estimations rarely match the actual environment in which the radio is operating. Consequently, estimations are inaccurate and require some type of adjustment to accommodate the current environmental conditions. Another limitation is that the bit-error-rate (BER) estimation is used as the foundation for estimating several other metrics. Therefore, if this estimation is imprecise, its effect is magnified during the calculation of an overall fitness which incorporates several metrics.

Chapter 3 introduced an abstract framework for incorporating experimental design within cognitive radio. The techniques provide a systematic approach to probing a cognitive radio search space to develop an empirical based estimation model. Statistical techniques have the potential to compliment existing architectures by providing a unique approach to conditions not previously encountered. In contrast to the GA, the empirical approach requires no knowledge of the current conditions or calibration for a new environment. Furthermore, an estimation model is developed for each metric individually without dependency on any other metric.

This chapter explores the utility and practicality of the experimental design approach to real-world

scenarios and compares performance to an existing GA cognitive engine [3]. Two response surface methodology (RSM) designs are implemented on an over-the-air wireless link under five distinct environments. The recommended solutions identified from the statistical models are compared to a benchmark GA cognitive engine. Chapter 3 discuss the calibration procedure necessary for the GA engine to operate properly. Both an unadjusted GA and adjusted GA were tested in comparison to the statistical approach. In all environments tested, decisions from the unadjusted GA were unable to establish a connection and therefore resulted in a fitness score of 0.

The structure of this chapter is as follows. The universal system radio peripheral (USRP) software-defined radio (SDR) hardware system model is presented. The methodology section then details the approach to implementing the statistical design on this platform. This includes defining the system inputs, operational ranges, translation onto the experimental design normalized scale, and metric definitions. In order to compare performance on a unified scale, utility functions and an overall fitness function are defined that convert multiple performance metrics into a single quantification of fitness. Next, the specific testing environments are described on which this methodology was repeated. Tabulated results for each environment include statistical fit for each output metric, confirmation tests of recommended solutions, and overall comparison of solution fitness. The chapter concludes with a summary and discussion of benefits the approach can provide existing architecture.

4.2 Hardware System Model

The system model is comprised of three SDR nodes each with a controlling laptop, as shown in Figure 4.1. The transmitter's laptop is also the master controller that instructs the interference node. The RF front end for the transmitter and the receiver nodes is a USRP-1 with a Flex 900 daughter board. The interference node is a USRP-2 with a WBX daughter card with a 50MHz to 2.2GHz frequency range.

This platform was originally developed for a cognitive radio application to support link adaptation in railway applications. In the system model, an assumed wireless feedback link was implemented using 802.11b. This feedback link enabled the receiver to transmit performance metrics back to

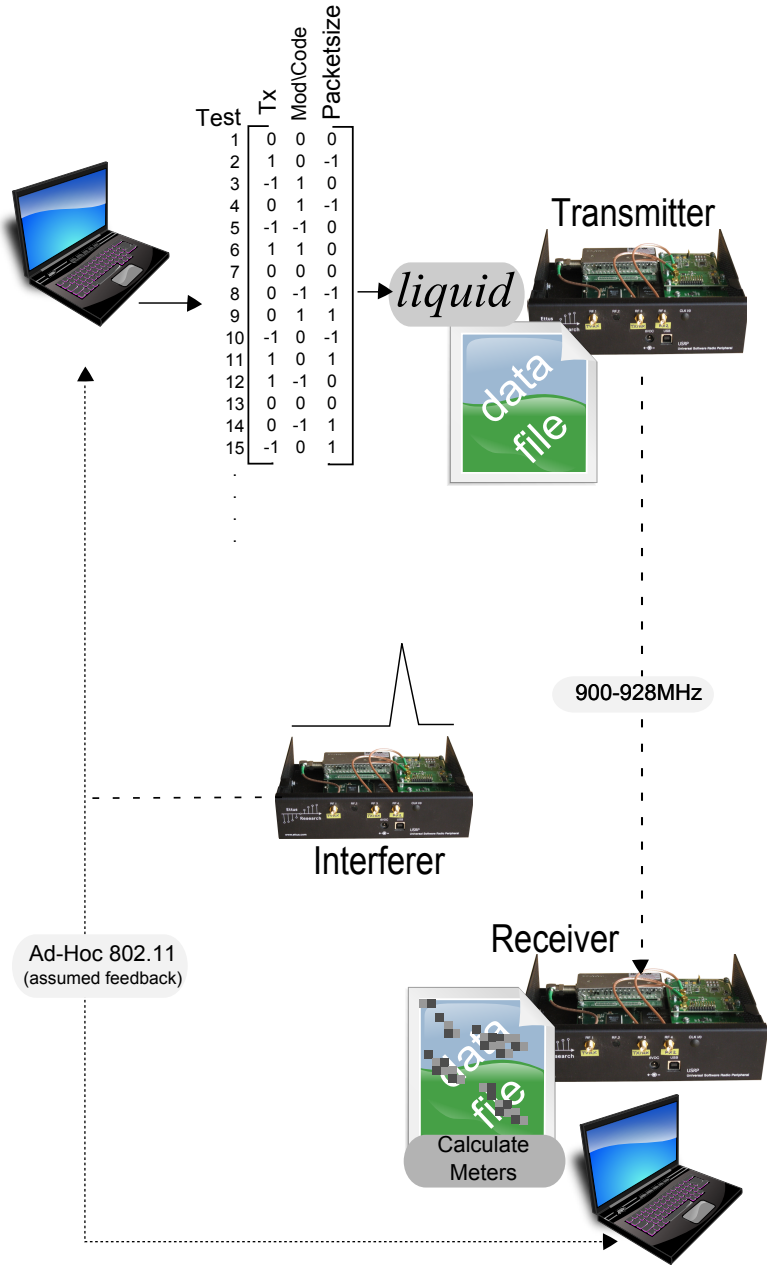


Figure 4.1: System Model

the controlling computer. To implement a dedicated feedback channel in practice, one could use a robust link connection similar to the method that *liquid*, the digital signal processing software, uses to inform the receiver about changes to modulation and coding. The first part of the *liquid* header is sent using a robust modulation and contains information regarding what modulation and coding should be implemented in order to demodulate and decode the following payload. In a similar fashion, a receiver could send a feedback message to the transmitter containing performance metrics needed for cognitive decision making. This 802.11b network also acts as a communications link between the controlling laptop computers for the interferer and receiver.

Each node has a dedicated software thread to interact with the master controller that is implemented via C++ UNIX sockets to send serialized message data to each node's controlling computer. In order to offload some of the processing from each node, a central broker architecture is implemented to synchronize and mitigate message distribution throughout the system. This is a simplified architecture based off of the CORBA/CROSS architecture [47].

USRP RF front ends require software based digital signal processing. This implementation used the *liquid* DSP software library [44]. This software provides flexible processing for sending data across the link. The library has significant functionality in terms of available signals, modulations, and error correcting codes, however a limited set was implemented here. Table 4.1 defines the basic *liquid* configurations used in this experiment.

Table 4.1: *liquid* DSP Configuration Parameters

Specification	Value
Transmission Frequency	910MHz
Transmission Bandwidth	200KHz
Modulation	BPSK, QPSK, 8-PSK, 16-QAM, SQAM-32, 64-QAM
Coding	Hamming(7,4), or No Coding
OFDM Sub-carriers:	28
Cyclic Prefix Length	6
Cyclic redundancy check scheme	CDR 16

4.3 Methodology

The major steps in implementing this experiment include defining the system inputs and output metrics, normalizing the inputs to a [-1,+1] scale, performing the design, analyzing the statistical fit, and developing estimation models of each output metric. This process flow was repeated for each of the testing environments described in Section 4.4.

4.3.1 System Inputs and Outputs

An initial step in experimental design, as described in Section 2.2, is defining the inputs and outputs. The inputs in this platform include configurable parameters that a cognitive engine may typically control. These include software gain, modulation, coding, and packet size. All input parameters are passed into the *liquid* DSP software control of the USRP. Gaeddert's past work in resource management showed that there are combinations of modulation and coding that are redundant or unnecessary [44](page 107). This justified simplification of the system from four input variables to three variables by modifying modulation and coding together. This resulted in a decrease in the number of tests in each design.

4.3.2 Factor Mapping

In order to implement a statistical design, the inputs require mapping to a coded scale, as described in Section 2.9.1. Table 4.2 lists the mapping of the minimum, maximum, and a nominal value for each input. These are coded as -1, 0, and $+1$. Input variables that are discrete in nature, such as modulation, require definition on a range of values. The DSP software has many available modulations and codings; however this implementation was limited to the same modulations and codings that were also implemented in the GA-CE in order to maintain consistency in the performance comparison. The existing GA-CE had BER equations for a smaller subset of modulations than *liquid* had available. The six available modulations ranged from binary phase shift keying (BPSK) to 64-quadrature amplitude modulation (64QAM). A mapping between these six options and the empirical [-1,+1] scale is shown in Table 4.3. BPSK corresponds to a normalized value of -1 and 16-QAM corresponds to a value of 0. When the empirical models are solved for a recommended

setting, a solution may arise that is between -1 and 0 which requires a choice between QPSK and 8-PSK.

Table 4.2: Coded values of input parameters

	Coded Value		
Parameter	-1	0	+1
liquid Software Gain (dB)	-56	-37.5	-19.5
Modulation	BPSK	16QAM	64QAM
Coding	Hamming(7,4)	Hamming(7,4)	No Coding
Packet Size (bytes)	20	140	300

Table 4.3: Defining Encoded Values

Input	CE Mapping	DOE Mapping	Value
Modulation	1	-1	BPSK
	2		QPSK
	3		8-PSK
	4	0	16-QAM
	5		SQAM32
	6	1	64-QAM

Metric Definitions

The metrics reported during testing include packet error rate (PER), signal-to-noise ratio (SNR), received signal strength indicator (RSSI), throughput, goodput, and spectral efficiency. SNR and RSSI are provided directly from *liquid* [44]. Packet errors were detected by verifying the cyclic redundancy check (CRC), where any packet that failed the CRC check upon receipt was considered a bad packet. PER was calculated by dividing the number of packets in error by the total number of packets. The throughput is defined as the total number of bits received divided by the time difference between the receipt of the first packet and the end of the transmission. The measured throughput does not take into account any overhead in the packet or packet errors. Goodput incorporates overhead and errors by only considering payload packets received correctly. Therefore, any packets that failed the CRC check are discounted as well as the bits related to overhead in the *liquid* frame. There are no retransmissions in this implementation. Packet size directly affects the goodput calculation because smaller packet sizes lead to more packets and more overhead. Spectral

efficiency is calculated by dividing the throughput by the fixed bandwidth of the transmitted signal. In this experiment, the transmission bandwidth was 200KHz.

4.3.3 Utility and Fitness Definitions

Section 3.2 describes an approach for comparing performance between architectures or approaches. The method focuses on defining utility functions, fitness functions, and utility weights. The utility definitions corresponding to (3.1) are listed in Table 4.4. This table defines parameters for transmit power and output metrics used in this implementation. Specifically, utilities for transmit power, coding, SNR, PER, spectral efficiency, and throughput are defined. The utility weights used here

Table 4.4: Utility Function Parameters

Parameter	Tx Power	SNR	PER	SE	Throughput
q_0	0.9000	0.9000	0.9000	0.9500	0.9000
$\dot{\beta}$	10.0000	50.1187	0.1000	3.0000	1.00E+06
q_p	0.1000	0.0500	0.0010	0.1000	0.1000
p	9.0000	0.0126	9.0000	0.0333	0.1000

were selected from the perspective of first prioritizing low packet error and then emphasizing lower transmit power. Table 4.5 lists the weights implemented in calculating overall fitness.

Table 4.5: Weights used in calculating fitness

Metric	Weight
Transmit Power	1
SNR	0.5
PER	10
Spectral Efficiency	0.3
Throughput	0.3

4.3.4 Identifying a solution from empirical models

Section 2.8.2 described how estimation models are developed during the analysis of statistical designs. SAS *JMP* 9.0 was used to analyze the results of the design implementation and derive the empirical estimation models. These models are a direct alternative to the theoretical models

derived in the GA. Regardless of the source of the model, in order to identify a solution, some type of search of the solution space and multi-criteria optimization is required. The core GA code used in the CE also provided a tool for identifying a recommended solution from the statistical models. The same heuristic, utility functions and weightings used in the GA-CE were applied to evaluate potential solutions from the statistical models. While this is not the most efficient approach, it did provide a generally good solution for comparative purposes. The goal was to identify a solution to empirical and GA theoretical models without time constraint as a limitation. It also enabled a consistent frame of reference for comparing the performance of the statistical methods to the benchmark-CE

4.4 Testing Environments

The general methodology for implementing a design generalizes to different environments and hardware platforms. However, the developed models, and results are tied to the specific environment in which the design was implemented. Selective use of the interference node, as well as modifying the signal at the transmitter, enabled different testing environments. The baseline SNR as measured in the ambient environment with a spectrum analyzer is estimated at -84dBm. The transmitted signal, shown in Figure 4.2, remained centered at 910MHz. The environments included:

- **Ambient conditions**: Under the ambient environment, the interferer was inactive. The tests were performed in the Wireless@VT graduate student lab with no outside control placed on other activities. This is a live environment and it was not verified whether there were other users in the transmission band during each test.
- **Wide-band interference**: The third USRP was set to a wide-band signal centered at 910MHz and 600kHz wide, as shown in Figure 4.3. The intent is to show a generic interference that is in band with substantial portions of the desired signals. This is also an interference condition that exists in railway communications when locomotives pass underneath alongside and underneath high-voltage power lines[1]

[1]This information was obtained in discussion with railway radio engineers.

- **Narrow-band interference**: The third USRP was set to a narrow-band jamming signal centered at 910MHz frequency with a bandwidth of 65KHz, as shown in Figure 4.4. This simulates the effect of another user or malicious user.

- **Dynamic narrow-band 'hopping' interference**: The narrow-band signal was randomly moved across multiple center-frequencies where the frequency and time-in-band were random variables created from a normal distribution. This selection of this distribution was arbitrary with the goal of creating some type of dynamic interference in contrast to the static nature of the other two interference profiles. The user-defined variables for this environment included the standard deviation of the random distribution, selection of available center frequencies, and the maximum time-in-band. The maximum-time-in-band is the highest length of time that the signal is allowed to stay at any one center frequency. For this experiment it was set to 2 seconds.

- **Artificial fading environment**: Creating a controllable fading environment on demand is difficult in an over-the-air system. It would require moving the radios or placing obstacles in the pathway during transmission. Performing this in a manner that enabled repeatability was unrealistic. Therefore, a crude artificial fading environment is created by adding or subtracting power at the transmitter. A crude slow-fading environment was created at the transmitter by modifying the transmit power based on a log-normal random distribution with a mean of 0 and standard deviation of 2dB generated within *liquid*. The modified transmit power was changed prior to sending each packet.

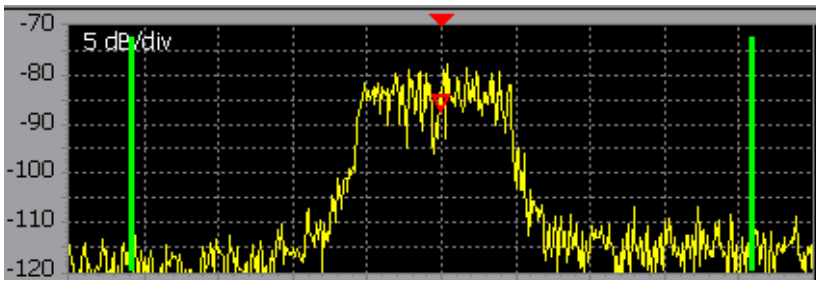


Figure 4.2: Transmitted Signal

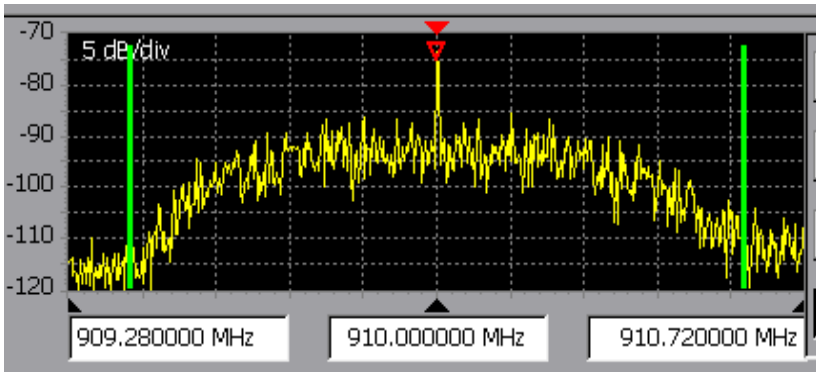


Figure 4.3: Wide-band Interference

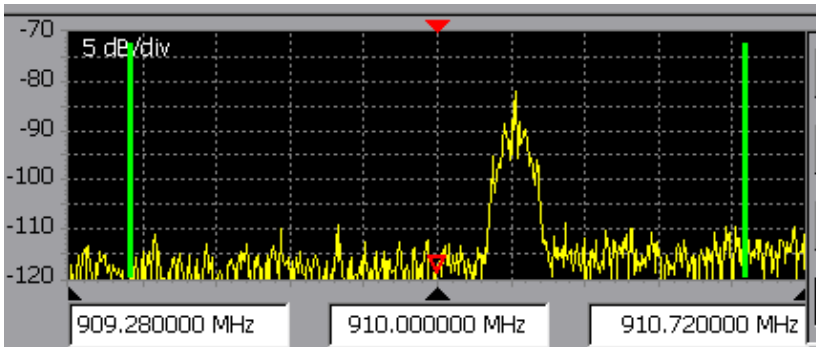


Figure 4.4: Narrow-band jamming signal

An issue with this radio system, and sometimes experienced in deployments, is that lack of connection was observed at low transmit power and high transmit power in certain environments. The lowest transmit power setting sometimes was unable to overcome the interference resulting in loss of connectivity. While the high transmit power setting occasionally saturated the link resulting in a similar connection loss. From a statistical fit and modeling perspective, the key is that even though the behavior is undesirable, it is important that it manifests itself consistently. The empirical method will identify this in the estimation models, however the GA will not because it only uses static estimation models.

4.5 Results and Discussion

The results are presented for each environment and include measures of statistical fit, the recommended input settings, and the results of the confirmation test. For confirmation tests, the result is listed followed by the 95% confidence interval. These were calculated using the *normfit* function in MATALB's statistical toolbox. The performance metrics are aggregated into a single fitness and the statistical design fitness is compared to the GA-CE bench mark decision.

4.5.1 Ambient Environment

In the ambient environment, the third-party interferer was not engaged, however there were no other controls placed on the laboratory environment. Upon completion of each statistical design, the data was analyzed in *JMP* 9.0 statistical software. The analysis yielded estimation models of each output metric as well as measures of the input parameter significance. The contrast between the estimation model and the observed values derives the R^2 measure of statistical fit, as described in Section 2.9.2. Figure 4.5 illustrates the R^2 for each performance metric, where each arm of the plot is the labeled metric and the R^2 value is between [0,1]. The higher the R^2, the better the fit between the model and the observations. Both the Box-Behnken and the CCD showed strong statistical fit with R^2 above 80% for all metrics. Spectral efficiency, throughput and SNR showed near 100% fit. The high fit for SNR is also repeated in other environments due to its sole dependency on transmit power. Therefore, SNR was straighforward to develop an accurate estimation model.

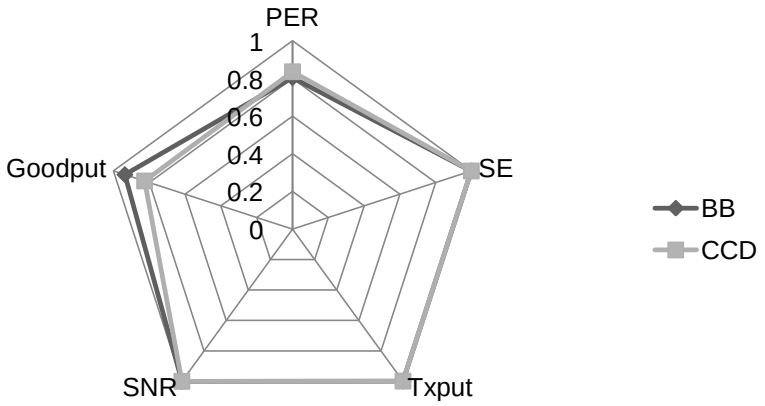


Figure 4.5: R^2 statistical fit of output estimation models for ambient environment

For all environments, the Box-Behnken showed slightly better statistical fit. This is most likely due to the lack of vertice points as described in Section 2.5.2. This indicates that observations of the vertice points showed less consistency, resulting in a lower statistical fit.

The recommended solutions identified by each approach are summarized in Table 4.6 where Software Gain is the value that is passed into *liquid* DSP. For comparative purposes, a decision from the unadjusted GA is also listed. In all environments, the unadjusted GA was unable to make a connection. The unadjusted-GA optimistically selected low transmit power, aggressive modulation and no coding. However, even in this ambient environment it was unable to connect. In contrast, the adjusted GA selected a less aggressive modulation and higher transmit power. Both the statistical methods selected transmit powers similar to the adjusted GA, however were more conservative in the selection of modulation and coding. The observed data from the design implementation showed that design points at low transmit power and aggressive modulation incurred higher PER. Therefore, the solutions to the resulting estimation models trended towards more conservative modulation with coding.

Table 4.6: Recommended input settings in ambient environment

Method	Software Gain	Packet Size	Modulation	Coding
GA-unadjusted	-61.15	278	64-QAM	no coding
GA-adjusted	-56.91	288	16-QAM	no coding
BB	-54.41	300	8-PSK	Hamming (7,4)
CCD	-55.93	203	QPSK	Hamming (7,4)

Table 4.7 summarizes the results of confirmation tests where each recommended input setting was directly implemented on the wireless link. PER is packet error rate, SE is spectral efficiency (b/s/Hz), Txput is throughput (bps), SNR is signal-to-noise ratio, RSSI is received signal strength indicator (dbM), and GP is goodput (bps). The (+/-) column is the 95% confidence interval range for the measurement (in the same units) based on at least 25 repetitions of the confirmation test. The decision from the GA-CE provided the highest throughput and goodput, however suffered from a PER of approximately 10%. The weighting scheme penalized PER more than it awarded high throughput. Conversely, the statistical approaches resulted in lower PER. Additionally, both the Box-Behnken and CCD resulted in lower throughput. These decisions cannot be compared until the combination of selected inputs and performance metrics are aggregated into a single fitness according to Section 4.3.3. The high weighting placed on PER, as shown in Table 4.5, negatively affected the overall fitness of the GA-CE. Even though it was able to transfer more data than then statistical methods, the final fitness was 2% less than the Box-Behnknen and 1.2% than the CCD. These results indicate that the conservative decision by the empirical approach did not negatively affect final fitness under this specific weighting scenario.

Table 4.7: Output metrics in ambient environment

Metric	GA-adjusted	(+/-)	BB	(+/-)	CCD	(+/-)
PER	0.104	0.041	0.002	0.006	0.004	0.006
SE	2.521	0.110	1.478	0.017	0.999	0.017
Txput	504258	22094	298590	3437	199863	3437
SNR	29.383	0.190	35.357	0.008	33.780	0.008
RSSI	-59.106	0.234	-53.110	0.011	-54.663	0.011
GP	453000	35109	295611	3437	199463	3437
fitness	0.7932		0.8627		0.8322	
% diff			2.1%		1.2%	

4.5.2 Wide-band Interference Environment

The wide-band interference environment, as described in Section 4.4, engages the third USRP to transmit a 600kHz wide signal centered at the same frequency as the transmitted signal. In the wide-band interference environment, the statistical fit dropped between 10-15% for goodput,

throughput and spectral efficicency, as shown in Figure 4.6. This was caused by increased variability between the prediction model and observed values. Interestingly, PER statistical fit improved under the interference. This was actually caused because of the way disconnects are dealt with during data collection. If no connection can be established during the file transfer, then PER is listed as 1. However, when a link is established, measured PER always showed some variability. From a statistical fit perspective, any variability in observed data is reflected in a decrease in statistical fit. In the presence of noise, lower transmit powers experienced a higher proportion of disconnects. The variability of PER for those observations was virtually zero because the PER was always 1. Therefore,the fit of the PER prediction model improved slightly. The decisions made by all methods,

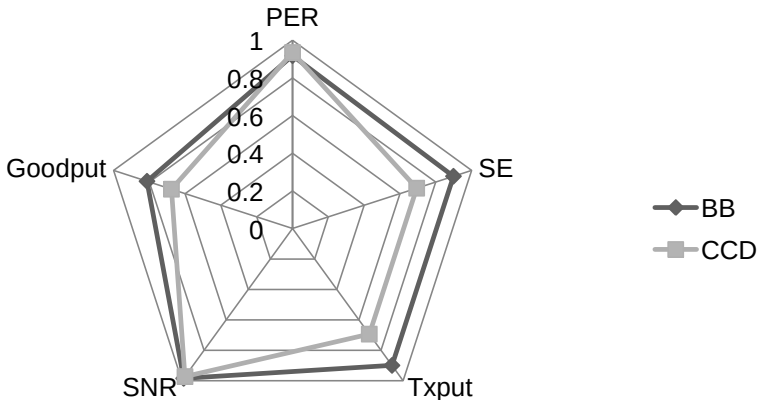


Figure 4.6: R^2 measure of statistical fit of estimation models for wide-band interference environment

as shown in Table 4.8, correctly selected a higher transmit power in order to mitigate the higher interference. Once again, the unadjusted GA decision was still unable to establish a connection. As in the ambient condition, the GA-CE selected a slightly more aggressive modulation scheme of 8-PSK compared to the more conservative QPSK selected by the statistical methods. Similarly, the GA selected no coding while the statistical methods both engaged coding. This was due to the empirical approach observing poorer performance in uncoded settings. In contrast, the GA's decisions are made solely on theoretical formulas with limited knowledge of how the system actually is performing. The resulting performance metrics from the confirmation tests are listed in Table 4.9. As expected, the GA's throughput performance was higher than the statistical methods due to the selection of more aggressive modulation scheme. However, this also incurred a higher PER. It was shown in the previous section, that there is a higher penalty paid with any PER degradation.

Table 4.8: Recommended input settings in wide-band noise environment

Method	Software Gain	Packet Size	Modulation	Coding
GA-unadjusted	-43.30	241	16-QAM	no coding
GA-adjusted	-19.59	184	8-PSK	no coding
BB	-20.16	298	QPSK	Hamming (7,4)
CCD	-19.07	201	QPSK	Hamming (7,4)

Therefore the higher throughput experienced by the GA decision was of relatively little benefit to the overall fitness. Overall, the fitness comparisons show that the GA performed less than 2% better than the statistial methods.

Table 4.9: Output metrics in wide-band noise environment

Metric	GA-adjusted	(+/-)	BB	(+/-)	CCD	(+/-)
PER	0.016	0.011	0.004	0.007	0.004	0.011
SE	1.837	0.007	1.077	0.002	0.987	0.010
Txput	367383	1387	215438	443	197343	1907
SNR	69.058	0.421	69.467	0.010	70.535	0.118
RSSI	-19.428	0.449	-18.999	0.017	-17.914	0.136
GP	362237	4911	215009	1622	196950	2981
fitness	0.6977		0.6537		0.6453	
% diff			-1.63%		-1.95%	

4.5.3 Narrow-band Noise Environment

The narrow-band interference environment simulated a point-interference source or a jamming signal of 65kHz bandwidth at 910.15MHz. The signal was static and did not change frequency. The statistical fit of the empirical methods showed similar performance to the wide-band interference condition, as shown in Figure 4.7. Again, the PER statistical fit improved compared to the ambient condition, as described in Section 4.5.2. The GA-CE recommended 16-QAM with error correcting coding, while the statistical approaches suggested QPSK with coding, as indicated in Table 4.10. The selection of QPSK by the statistical methods was due to disconnects occurring when set to higher modulation. Similarly, tests with uncoded settings also experienced several disconnects. This drove the solution for the statistical designs away from higher modulations and uncoded settings.

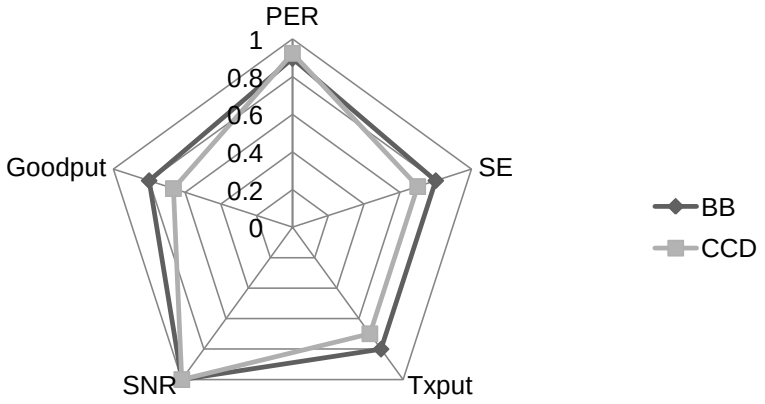


Figure 4.7: R^2 measure of statistical fit of estimation models for narrow-band interference environment

The Box-Behnken and the CCD design both selected similar transmit power, modulation and coding, however they differed in the recommended packet size, where the Box-Behnken recommended the maximum packet size of 300 which the CCD selected 200. The raw data identifies that the best performance for the Box-Behnken design occurred when the packet size was set to maximum, specifically the [1,0,1] data point. In contrast, because the CCD is based on a factorial design, the occurrences of packet size set to a maximum also coincide with the other input settings either set to a maximum or minimum. In this and the wide-band interference environment, minimum transmit power was unable to overcome the interference, and maximum modulation incurred too many errors. The one data point where packet size and transmit power were set to maximum while modulation was conservative did maintain a connection, however throughput and goodput were consequently low. Therefore, there were no occurrences where packet size was set to maximum that yielded strong performance in the CCD. Hence, the solution identified a lower packet size. This is important because lower packet sizes incur more overhead and decrease usable payload. The weighting scheme for this particular fitness calculation did not emphasis packet size, however this may be desired in other scenarios. The results of the confirmation test for the narrow-band interference condition, shown in Table 4.11 resemble the wide-band interference environment. The GA decision again showed higher overall data transfer, however incurred more packet errors. The more conservative modulation scheme observed less packet error at the expense of transferring less data. The final fitness evaluation demonstrate that the statistical methods produced decisions less

Table 4.10: Recommended input settings in narrow-band interference environment

Method	Software Gain	Packet Size	Modulation	Coding
GA-unadjusted	-39.85	238	16-QAM	no coding
GA-adjusted	-20.48	199	16-QAM	Hamming (7,4)
BB	-20.16	297	QPSK	Hamming (7,4)
CCD	-19.22	192	QPSK	Hamming (7,4)

than 2% lower than the GA-CE.

Table 4.11: Output metrics in narrow-band interference environment

Metric	GA-adjusted	(+/-)	BB	(+/-)	CCD	(+/-)
PER	0.016	0.017	0.002	0.006	0.004	0.007
SE	1.604	0.001	1.074	0.000	0.981	0.004
Txput	320757	172	214830	48	196130	892
SNR	68.630	0.092	69.447	0.010	70.392	0.018
RSSI	-19.774	0.064	-19.023	0.019	-18.075	0.020
GP	316268	4651	214830	48	195738	1504
fitness	0.6915		0.6536		0.6452	
% diff			-1.41%		-1.73%	

4.5.4 Dynamic Interference Environment

The first three scenarios involved static environments where the center frequency of the interference was fixed. In static environments, statistical fit for empirical estimation models showed good performance with most R^2 exceeding 70%. A concern is how well the approach will work in a more dynamic environment in terms of statistical fit and more importantly, the fitness of the final decision. For this reason, the dynamic narrow-band interference scenario, described in Section 4.4 was developed.

The R^2 results for the this dynamic interference scenario are listed in Table 4.12. Degradation in statistical fit between 5-40% occurred where the CCD incurred the most prounouced decrease. The reasons for the decrease relate back the raw data where repetitions of specific tests showed inconsistency. This translates into lower statistical fit of the model. This is not unexpected given the dynamic nature of the environment. A remaning question is how will this affect the recommended

solution and the confirmation test result.

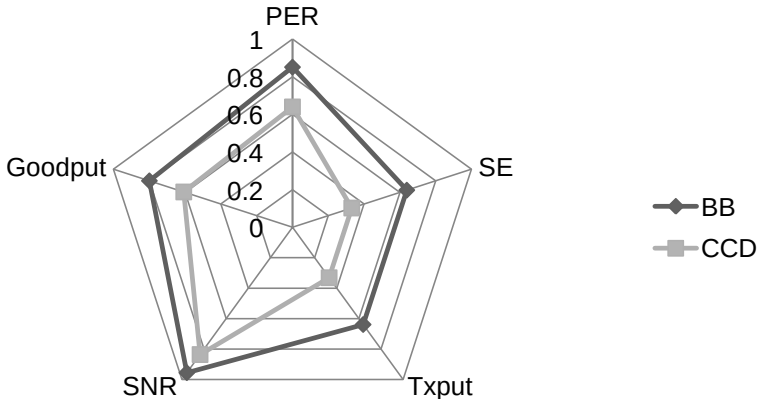


Figure 4.8: R^2 measure of statistical fit of estimation models for dynamic interference environment

The scenario was run on both the reference GA and the statistical designs. As expected, the decision from the unadjusted GA was unable to initiate a connection. In contrast to the previous environments, the adjusted GA selected a more conservative decision than the statistical methods. In this case, the GA selected BPSK while both statistical designs recommended QPSK. One reason behind the selection is the adjustment required to assist the theoretical formulas for this environment as introduced in Section 3.4.1.

In contrast, the statistical approaches used only the observed data collected when the design was exercised to develop estimation models for each metric individually. The models developed from the collected data suggested that QPSK was capable of maintaining a connection that produced higher throughput without more degradation in PER, as listed in 4.12. The results of confirmation tests, shown in Table 4.13, show that the estimates from the statistical models were valid in determining that QPSK was able to produce higher throughput without higher PER. The overall

Table 4.12: Recommended input settings in dynamic interference environment

Method	Software Gain	Packet Size	Modulation	Coding
GA-unadjusted	-44.29	193	16-QAM	no coding
GA-adjusted	-19.59	294	BPSK	Hamming (7,4)
BB	-19.43	299	QPSK	Hamming (7,4)
CCD	-19.79	295	QPSK	Hamming (7,4)

fitness tabulation demonstrates that the decisions from the statistical approach improved upon the GA by 2-3% based on the current fitness weighting definition.

Table 4.13: Output metrics in dynamic interference environment

Metric	GA-adjusted	(+/-)	BB	(+/-)	CCD	(+/-)
PER	0.083	0.070	0.068	0.064	0.086	0.080
SE	0.557	0.022	1.028	0.050	1.005	0.065
Txput	111403	4410	205690	10032	201026	13039
SNR	67.299	1.639	67.931	1.530	66.914	1.854
RSSI	-19.558	0.193	-19.211	0.039	-19.763	0.161
GP	110819	4457	203504	9900	200184	13454
fitness	0.5516		0.6317		0.6045	
% diff			3.38%		2.29%	

4.5.5 Artificial Fading Environment

The artificial fading environment was created by adding a random component to the software gain prior to the transmission of each packet. Statistical fit, as shown in Figure 4.9, was similar in performance to the ambient environment. Similarly, the decisions made by the GA and the statistical methods matched closely to the ambient environment with the exception of transmit power. Table 4.14 shows that primary reaction to the environment was to increase the software gain of the transmission signal. The GA selected 16-QAM modulation, while the statistical methods both selection 8-PSK. The observed data taken during the design implementations showed that higher

Table 4.14: Recommended input settings in artificial fading environment

Method	Software Gain	Packet Size	Modulation	Coding
GA-unadjusted	-52.20	273	64-QAM	no coding
GA-adjusted	-24.03	271	16-QAM	no coding
BB	-23.70	285	8-PSK	Hamming (7,4)
CCD	-25.29	292	8-PSK	Hamming (7,4)

packet error occurred at the more aggressive modulations. The decision made by the statistical approaches reflected this and were verified in confirmation tests as listed in Table 4.15. The GA's decision did incur slightly higher error, however it also enabled higher throughput and goodput

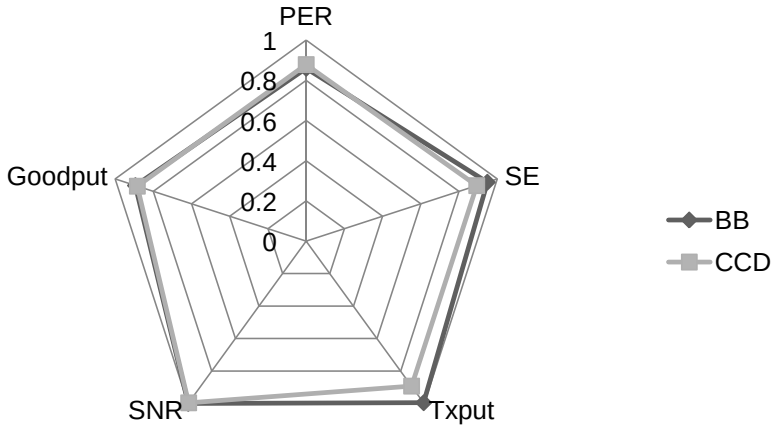


Figure 4.9: R^2 measure of statistical fit of estimation models for artificial fading environment

than realized by the statistical method's recommendations. The fitness tabulations show that overall, the statistical methods were less than 2% lower than the GA's decision. A summary of

Table 4.15: Output metrics in artificial fading environment

Metric	GA-adjusted	(+/-)	BB	(+/-)	CCD	(+/-)
PER	0.024	0.021	0.010	0.009	0.012	0.011
SE	2.496	0.027	1.450	0.017	1.448	0.025
Txput	499185	5339	289967	3310	289606	5025
SNR	41.365	1.264	43.026	0.156	39.258	0.145
RSSI	-47.597	1.411	-45.379	0.416	-49.586	0.020
GP	486935	7175	288199	1775	287246	5061
fitness	0.7313		0.6929		0.6989	
% diff			-1.35%		-1.13%	

the overall fitness results, shown in Figure 4.10, illustrates that the statistical approaches were within 4% of the GA for all environments. This illustrates that this unique approach is capable of similar performance to traditional approaches without the benefit of prior calibration for a specific environment. This provides a complimentary methodology when facing a new situation where past calibration of the GA or past experience in a case-based reasoner is unavailable.

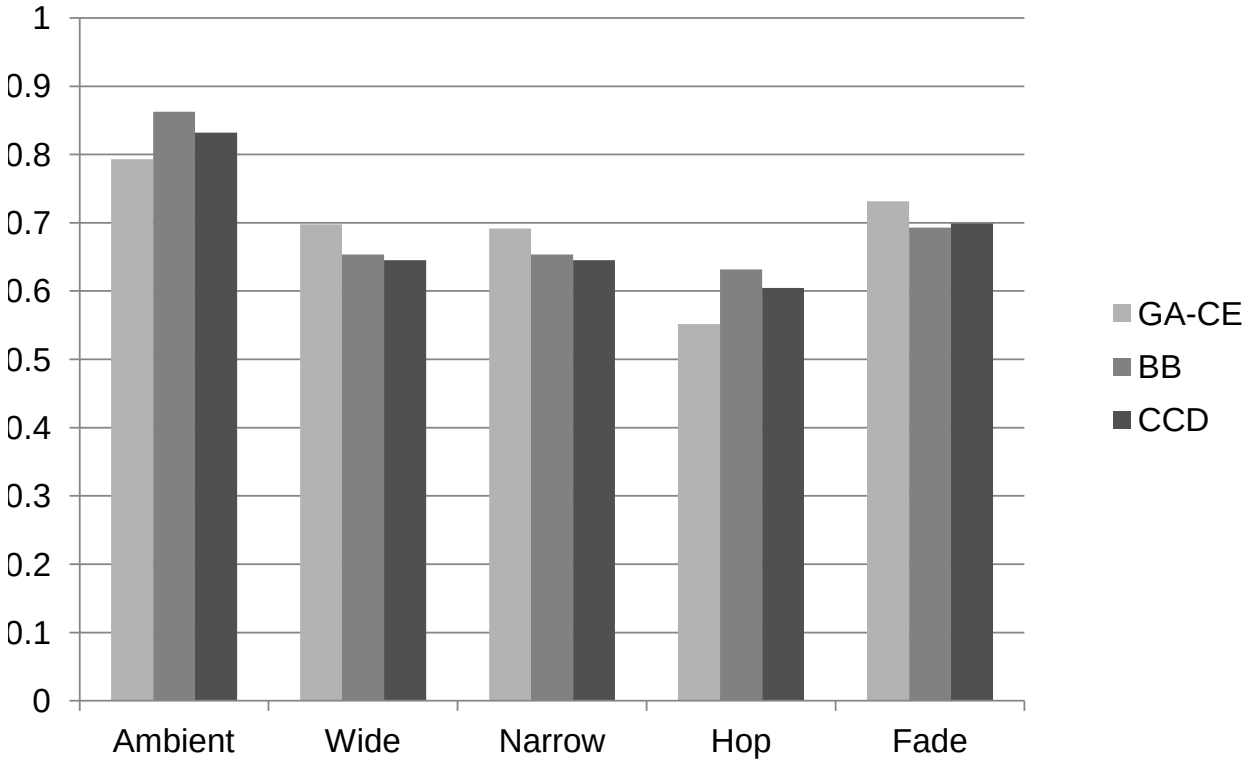


Figure 4.10: Overall comparison of fitness between RSM statistical designs and GA-CE

4.6 Summary

This chapter demonstrated the utility and practicality of the experimental design approach in real-world scenarios. The methodology compared the Box-Behnken and CCD RSM statistical designs against a standard approach of a GA cognitive engine. The empirical designs and the GA were implemented in five distinct environments to identify recommended input parameter settings based on pre-defined goals. The results were compared on a common fitness evaluation and showed that the statistical designs were capable of performing within 4% of the GA. In a dynamic interference environment, the statistical fit of the empirical estimation models dropped compared to static situations. However, the identified solution was still capable of performing similar to the GA.

In contrast to the GA's reliance on theoretical estimation formulas and *a priori* knowledge of the environmental conditions, the RSM designs were able to construct empirical-based estimation models with no previous knowledge of the system. The approach compliments traditional approaches

and has implications for approaching new environments and for existing CBR architectures. The solutions to the statistical approach can be viewed from the perspective of training a CBR. The implementation of a design on a specific environment provides an experience from which the cognitive engine can draw upon later. Designs can be implemented on known situations that an engine will encounter in order to populate a case library. In addition, the models derived from performing a design might provide a way of checking the relevancy of existing case libraries.

Chapter 5

Taguchi Based Parametric Optimization[1]

5.1 Introduction

Previous chapters introduced experimental designs and developed a framework for applying the concepts to cognitive radio. Specifically, Chapter 3 introduced a unique approach for identifying input parameter settings in cognitive radio using experimental designs. Chapter 4 extended these concepts by applying them under real-world conditions to perform wireless link adaptation. While factorial and response surface methodology (RSM) provide powerful tools to identify representative knowledge and empirical estimation model of output performance, the cost to benefit ratio diminishes as the number of input parameters grows. For example, a four-variable configuration with three levels each requires $3^4 = 81$ individual test cases at the minimum. The Taguchi approach, introduced in Section 2.6, has potential to address this trade off between knowledge gain and number of input parameters.

This chapter applies the Taguchi method as an alternative to traditional factorial and RSM designs. Specifically, the Taguchi method is used to identify initialization settings of a genetic algorithm

[1]Portions of this chapter were originally published in [48]

(GA) cognitive engine module. The Taguchi approach incorporates fractional factorial designs that significantly reduce the number of tests while maintaining the capability to identify recommended settings of input parameters. The trade offs include requirement for using discrete value inputs. Additionally, there is no information gained on input significance, interaction effects or estimation models. When compared to the RSM method, the key difference is in the estimation models. The RSM approach develops estimation models that are continuous across the inputs and includes coefficients for combinations of terms. The Taguchi approach does not account for higher order interactions. This approach is strictly discrete and only provides estimations for the discrete setting of the input.

The GA requires setting of crossover rate, mutation rate, population size, and maximum generations prior to normal operations. Applying statistical approaches to identifying these parameters illustrates a new use for the experimental design framework presented in Section 3.5 by extending the concepts beyond the traditional SDR inputs of a cognitive radio. Furthermore, this effort differs from past literature associated with identifying GA parameter optimization by performing the assessment within a complete cognitive radio framework. Existing GA performance assessments focus on convergence of commonly researched objective functions. In this application, the analysis takes into account the performance of cognitive radio decision making as well as how well the engine uses the case-based reasoner (CBR) in conjunction with the genetic algorithm.

To the best of our knowledge, this is the first use of Taguchi analysis to identify GA initialization parameters and the first use of Taguchi methods within a cognitive radio application. The concept of Taguchi signal to noise ratio, as described in Equations 2.8, 2.9, and 2.10, also provides a non-traditional approach to performance analysis by tracking variation of metrics in addition to their mean. Other contributions include using concepts from computer performance testing, not previously explored, to assess cognitive radio. These include using figure of merit (FOM) to aggregate multiple performance measures into a single quantification and Kiviat graphs to visualize cognitive radio performance. These differ from traditional radar plots, as seen in Chapter 4, by emphasizing balanced operations.

The remainder of this Chapter is structured as follows. Background on the use of GA with cognitive radio and efforts to identify ranges for GA parameter settings is discussed. This GA is used within

a cognitive radio architecture, therefore background on the engine architecture is presented. Refer to Section 2.6 for background on the Taguchi method and Section 3.7 for discussion of specific limitations of the method. The methodology section includes defining the Taguchi approach on this application and discussion of the performance metrics and FOM. Results include identification of recommended parameter settings and tabulation of estimates of performance. Kiviat graphs provide graphical representation of performance. A confirmation experiment is run using the predicted best parameter settings and compared against the calculated performance. A summary and suggestion for further research concludes the chapter.

5.2 GA use in Cognitive Radio

Cognitive radio architectures have gravitated towards GA as potential decision-making algorithms given their capability of solving complex spaces based on multi-objective definitions [8]. Refer to Section 3.3 for background information on the GA and details on applying it to cognitive radio. In summary, utilization of GA within wireless communications application space requires modeling the physical (PHY) layer traits of the radio within the context of a genetic chromosome. PHY layer characteristics such as BER, modulation, and frequency are represented as variable bit representations of genes. Non-linear utility functions are used to convert PHY layer meters into values between [0,1]. These utilities are aggregated into weighted fitness functions that could be tuned to emphasize specific radio missions such as minimizing transmit power or maximizing throughput [49]. A typical process flow for the use of a GA within cognitive radio is as follows:

1. The radio parameters each represent a gene which are encoded together to form a chromosome.
2. The initial population is created either from random generation, or from the output of other modules of a cognitive engine, such as a CBR [3].
3. During each generation, the chromosome's genes are decoded to identify the suggested radio parameters.
4. The radio parameters are used to estimate performance meters. Both the parameters and estimated meters are normalized using utility functions and combined into a single measure

of fitness.

5. The next generation is created by crossing over genes from the parents with the highest fitness.
6. Each bit of the population is randomly mutated with a fixed probability.
7. The algorithm repeats the process for a defined number of generations.

While powerful, the heuristic nature of the algorithm was plagued with slow operations. To enhance decision-making speed, the GA architecture was hybridized with experientially-based decision making such as CBR. Experiential databases provide a faster first attempt to match the current situation with a successful decision made in the past and is executed in place of calling the GA. If a sufficiently similar case is not found, the top retrieved cases can act as partial seeds into a GA with the goal of improving performance by providing a better starting point [10].

5.2.1 Identifying GA Parameters

The selection of GA configuration parameters will have great bearing on the success of the algorithm. While there are many variations of the GA, the key configuration parameters are crossover rate, mutation rate, population size, and maximum generations. Definitions for each are detailed next.

1. **The Crossover rate**, or probability of crossover, is configuration parameter that affects the rate at which crossover between parents occur. A higher crossover rate, increases new strings into the population faster. Too low a crossover rate will limit the exploration rate due to lower number of potential solutions.
2. **Mutation rates** affect the speed of searching. Mutation rate is the probability that each bit of the string undergoes a random flip after the selection of a new parent. Too high a rate makes the search similar to a random search which can be inefficient. Too low a rate limits the diversity of the population making finding the best solution harder.
3. **Population size** controls the amount of chromosomes that the GA has in each generation. Too big a population size requires a longer search of the current generation to identify the

best ones of the generation which will become the parents of the next generation. However, too small a population again limits future diversity.

4. **Maximum generations** limits the number of iterations that the GA is allowed to make before a final solution is identified. GA's are known to converge to near-optimal solutions, however the time it takes to reach a solution is an important consideration. In cognitive radio, the environment may change before the GA has had a chance to fully converge, therefore maximum generations cannot be too high. This leads to having to make some concessions into the final fitness of the solution as it may not have converged on the overall best solution. In this case, a 'good-enough' solution maybe the best case.

Since the inception of GA, there have been several efforts to identify ideal parameter settings. De Jong's key work evaluated four parameters: population size, crossover rate, mutation rate, and generation gap [50]. His conclusions provided recommended ranges that are considered default settings for basic GAs. The suggested guidelines were population size: 50-100; crossover rate: 0.6; and mutation rate: 0.001. Work by Schaffer et al. investigated interaction effects between parameter settings and suggested an inverse relationship between population size and mutation rate [51]. This effort led to recommendations for parameter ranges of population size: 20-30; mutation rate: 0.005-0.1; and crossover rate: 0.75-0.95. The Taguchi method has been applied to the problem of identifying GA parameters using the same theoretical test objective functions as utilized by DeJong [52]. The results indicated that GA parameter settings were dependent on the specific test application.

With regard to interaction effects between parameters, Rezende performed statistical analysis using design of experiments methodology to identify these relationships between population size, number of generations, crossover probability, and mutation probability, as well as the qualitative factors of crossover type and mutation type [53]. Results indicated that crossover type had the most effect on performance, and interaction effects were most prevalent between population size, maximum number of generations, crossover rate and type of crossover, and mutation rate and type of crossover. The statistical value of the interaction effect between population size and maximum number of generations was 0.03 where p=0.05 is the typical cutoff. Therefore, this interaction is considered relatively weak and not an impediment to using the Taguchi method here. The Taguchi method

typically assumes that there are no interaction effects between the input parameters under study.

5.3 CE Architecture

This section overviews the CE architecture that provided the platform for the Taguchi analysis of GA configuration parameters. Figures 5.1 illustrate the CE architecture [9]. The CE is tethered to a software-defined radio such that it observes system parameter information from the radio and implements new configurations. The engine maps the configuration parameters of the radio (commonly known as 'knobs') as well as radio performance metrics (commonly known as 'meters') into a vector representation. This vector enables the use of similarity calculation for CBR-based engines to compare different situations against each other. CBR is founded on the belief that solutions to new situations can be identified based on solutions used in similar situations in the past. This is discussed in more detail in [3, 9, 11]. The decision process first attempts to use the CBR. If a past decision does not fall within a defined similarity threshold, then the CE calls the GA. Only GA configuration parameters are changed between test runs, as indicated in Table 2.6.

When a CE engine is initialized without a case base, it must rely on the GA to make decisions. Each time a good decision is made, such that performance improves after the decision, that case is added into the past history as a successful decision. CBR case retrieval is based on a combination of similarity to the current situation as well as the resulting success of the past decision. Therefore, past decisions that also have a high fitness have a better chance of being retrieved as a potential solution to a new situation. Ideally, this case can be retrieved in the event that a future situation matches this past situation.

5.4 Experimental Approach

This method strives to assess the performance of the GA initialization parameters in terms of overall CR performance and not just the fitness function utilized within the GA. The hybrid architecture of the GA and CBR contains several indicators of good performance that might not be considered in a traditional assessment. Typically, metrics such as BER or throughput are considered when

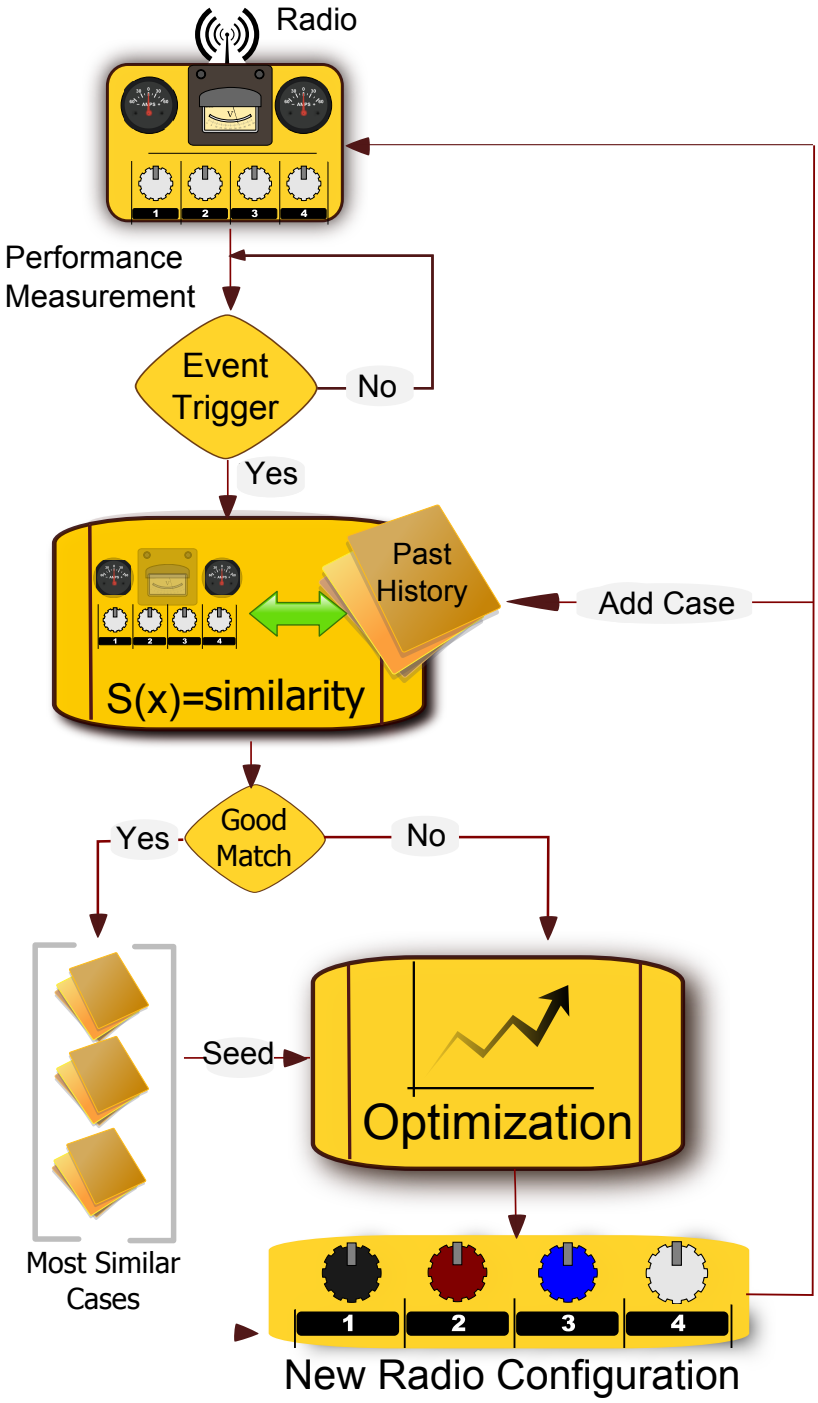


Figure 5.1: Basic CE Architecture (used by permission from [3] ©2011 IEEE)

assessing performance. However, to truly evaluate the engine performance one must consider other factors related to the cognitive performance. For example, as the GA engages to mitigate the interference, new cases are appended to the case library. Ideally, the next time the engine is engaged, a recently appended case will be recalled instead of engaging the GA. Therefore, another metric of performance might include the percentage that the CBR is used compared to the GA. Similarly, CBR performance is faster than the GA, hence overall time-to-action which tracks the amount of time from engine engagement to implementing a decisions, is an indicator of performance. The more times the GA is engaged, the more opportunities there are to add new cases. The problem with this, is that the higher the number of cases, the longer it takes an engine to search the case library. Therefore, the case addition rate is an indicator of performance.

A 'test run' can be defined as a repeatable transfer of a large data file across a link in the presence of an interferer. The run is started with an empty case base in order to measure how well the GA's decisions are appended to the case base and retrieved. In order to measure this in the same test run, the engine operations was set to a 'forced decision' mode. Typically, the engine is only engaged when a defined metric falls outside of specification. The problem is that after one good decision the engine would leave the decision loop. In a forced decision mode, after a decision is made, the engine is required to engage and go through the decision loop continuously. This enabled the generation of several data points during each test run. Under this type of operations, the engine can make the same decision as the previous iteration. For example, the engine might retrieve and implement the same case as the prior decision. The metrics will reflect this as desired behavior as the case addition rate will not increase, the percentage use of the CBR increases while the percentage use of the GA decreases, and the average time-to-action will decrease because the CBR use is faster than the GA.

5.4.1 Metrics

The performance metrics that define the overall measure of success, as shown in Table 5.1, include BER, number of decisions made using the GA, average fitness after a decision is made, average throughput, number of decisions made using CBR, and average time to action. The architecture will operate with faster decision cycles the more that it can use the CBR over the GA, and operate

with higher fitness performance if better decisions are made. Therefore, the GA parameter settings are judged with regard to improvements of fitness post-decision, how fast decisions are made, and how often the engine is able to rely on the CBR for decisions. These metrics are broadly categorized as higher is better (HIB) and lower is better (LIB). BER can be viewed as a LIB metric in its standard form, or converted to HIB by dividing the BER by a target-goal BER. In this case, BER is converted to HIB. The use of the Merrill's figure of merit(FOM) is suggested to combine alternating HIB and LIB metrics into a single quantified value as shown in (5.1) [54]. In the FOM, performance is defined by a vector = $x_1, x_2, x_3, \cdots, x_{2n}$ where each x is a metric and odd values are HIB and even values are LIB. The alternating of HIB and LIB metrics also supports kiviat graphs by highlighting balanced behavior. When metrics are shown in a spider plot in an alternating form, balanced performance is quickly illustrated by a symmetrical star shape.

The FOM is not without its drawbacks. It considers all axes and metrics of equal weighting, and extreme values are viewed as favorable. This may not always be the case. Equal FOM values between two different systems do not mean that the systems are equally as good. There could be imbalances in the system which the FOM calculation does not show.

$$FOM = \left[\frac{1}{2n}\sum_{i=1}^{n}(x_{2i-1} + x_{2i+1})(100 - x_{2i})\right]^{\frac{1}{2}} \tag{5.1}$$

Table 5.1: Performance Metrics

Metric	Type	Justification
BER goodness	HIB	Inverse of BER. Higher value shows better performance
GA use	LIB	% use of GA. Better performance when GA is used less
Ave. Fit	HIB	Ave. fitness. Better decisions lead to higher fitness
Added cases	LIB	Increasing cases will make search times go up
Ave Txput	HIB	Higher throughput is desired
Case add rate	LIB	The rate in which cases are added indicates more use of the GA
CBR use	HIB	% use of the CBR shows better performance if higher
Ave TTA	LIB	Ave. time-to-action decreases the more the CBR is used over the GA

5.5 Experimental Design

A Taguchi design based on the L9 orthogonal array, as described in Section 2.6, was implemented here. The design is listed in Table 5.2 where A, B, C, and D represent input variables. The system

Table 5.2: L9 Orthogonal Array

Test ID	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

model was comprised of a transmit-receiver chain and external interference source, as shown in Figure 5.2. This model was realized in MATLAB simulation. A data file was transmitted to the receiver utilizing BPSK modulation, forward error correction coding, and additive white Gaussian noise injected into the system. The CE was required to make a decision regarding whether to change transmit power, or coding in response to changes in system performance. Feedback of received BER, presence of and magnitude of external interference power were assumed.

A decision cycle consisted of: 1) observation of received BER and interference power, 2) making a decision using either the CBR or the GA to change or maintain existing settings of either the transmit power or coding, 3) observing the resultant performance metrics after a decision, and 4) augmenting the existing case-based history if a decision resulted in improved performance. After the tenth decision cycle, a simulated external interference source was engaged. The interference was turned off after the twentieth decision cycle and the test ended after 30 decision cycles. Each decision cycle provided performance measures in terms of overall system fitness and time-to-action for each decision whether the CBR or the GA was used.

Figure 5.3 illustrates the results from one example test run. The x-axis represents the decision index. Stars indicate that the GA was utilized to make the decision, while circles indicate that

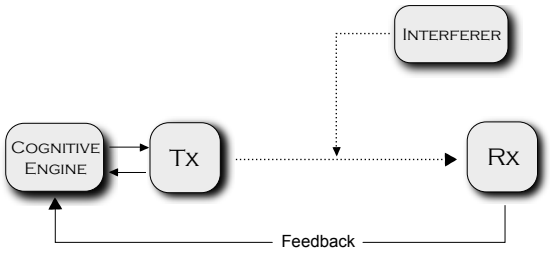


Figure 5.2: System Model

the CBR made the decision. Successful decisions made by the GA are saved into the case base history for retrieval by the CBR based on similarity calculations. The resulting metrics across 30 decision indexes are averaged and then combined into the FOM as defined by (5.1). This example resulted in a $FOM = 0.7833$ and is illustrated by the Kiviat graph shown in Figure 5.4. The arms of the spider chart represent normalized values of the metrics list in Table 5.1 where TTA is average time-to-action, C. Add Rate is the case addition rate, %CB is the percentage of decisions that use the CBR, %GA is the percentage of decisions that used the GA, and added cases is the normalized value of the number of new cases added during the run.

Typically, a CE would operate in an event-driven cycle such that a decision process is entered only if predefined metrics go outside of defined specifications. In this testing protocol, the engine was operated in a forced decision mode such that decisions were required every cycle to generate a higher number of decisions per test. Even if the engine was selected to not make a change to existing knob settings, this was still considered a decision.

The GA parameters manipulated between test runs were A) crossover rate; B) mutation rate; C) population size; and D) allowable maximum generations. The range of values these parameters take was selected based upon findings discussed in Section 5.2.1. The values of each of these parameters were mapped into three discrete levels, as shown in Table 5.3. This mapping is used by first referring to a specific row of the experimental design shown in Table 5.2. Each row indicates the level that each input variable (A,B,C, or D) should be set to in terms of a a value between 1 and 3. Next, lookup the mapped value for each input variable in Table 5.3, and set the engine configuration accordingly. Note that the Taguchi approach maps inputs onto a discrete scale defined by [1,2,3,...]. This is different than the factorial design and response surface methodology that use

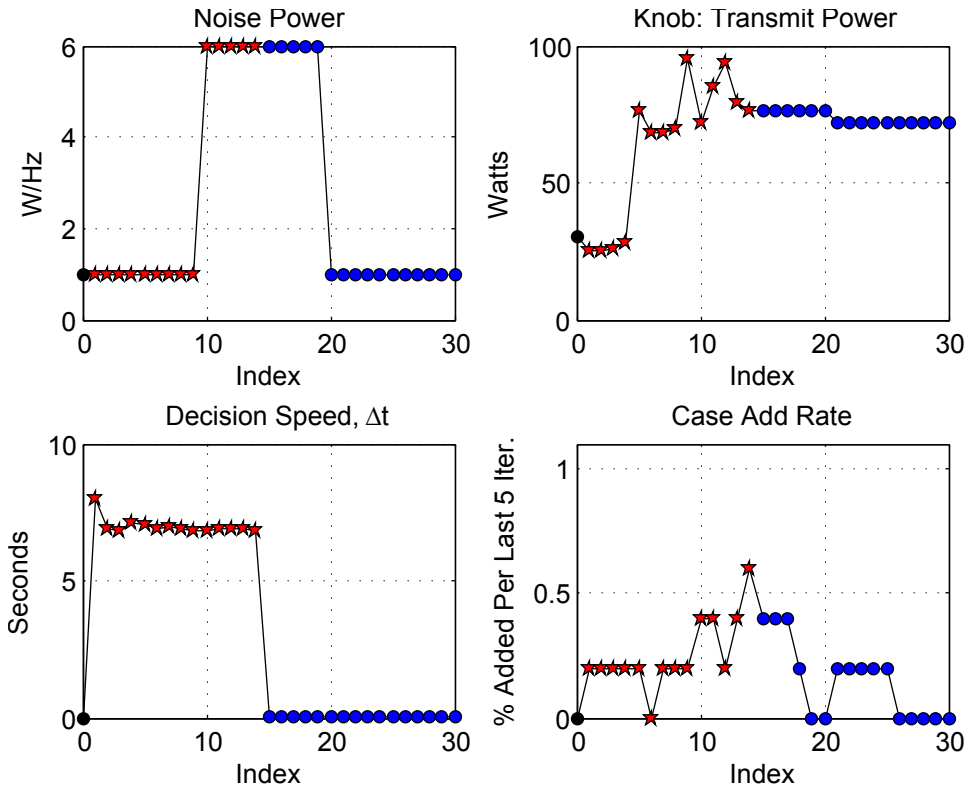


Figure 5.3: Example Test Run

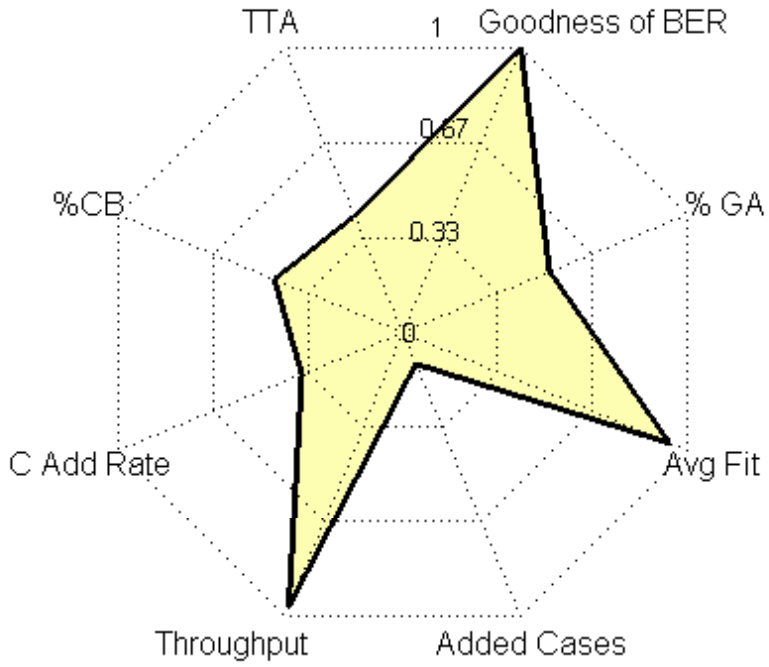


Figure 5.4: Kiviat Graph with a FOM of 0.7833

a scale of [-1,0,1].

Table 5.3: GA configuration parameter ranges

Parameter Level	1	2	3
A:Crossover Rate	0.55	0.675	0.95
B:Mutation Rate	0.005	0.0075	0.01
C:Population Size	50	75	100
D:Maximum Generations	25	50	75

5.6 Results

Nine separate testing configurations of GA parameters were tested. Each configuration test was repeated five times. The overall FOM for each replicate of the test was recorded, as shown in Table 5.4. *SAS JMP 9.0* statistical software package was used for the analysis of the Taguchi testing matrix. The SNR_{Tag} utilizing an HIB goal, per (2.9), was calculated for each test configuration along with a calculation of desirability, indicated by d as discussed in (2.35).

Desirability functions are a common tool for assessing optimization of multiple outputs in statistical analysis, as described in Section 2.12.2. Desirability functions are similar to utility functions common in cognitive radio because they map inputs to a [0,1] scale where 0 is least desirable and 1 is most desirable. In this situation, the goal is to take the two outputs, FOM and SNR_{Tag} and aggregate them into one desirability value. Both are weighted equally. The JMP analysis performs

Table 5.4: Collected Data from L9 Testing Matrix

ID	A	B	C	D	run-1	run-2	run-3	run-4	run-5	mean	SNR_{Tag}	d
1	1	1	1	1	0.6488	0.6465	0.6504	0.6535	0.6613	0.6521	-3.7145	0.0762
2	1	2	2	2	0.7546	0.7574	0.7586	0.7590	0.7557	0.7571	-2.4175	0.5035
3	1	3	3	3	0.7284	0.7282	0.7308	0.7330	0.7285	0.7298	-2.7363	0.3922
4	2	1	2	3	0.8433	0.8436	0.8451	0.8483	0.8438	0.8448	-1.4648	0.8746
5	2	2	3	1	0.6700	0.6691	0.6760	0.6820	0.6753	0.6745	-3.4212	0.1690
6	2	3	1	2	0.8393	0.8180	0.8418	0.8428	0.8451	0.8374	-1.5432	0.8430
7	3	1	3	2	0.8603	0.8604	0.8606	0.8612	0.8630	0.8611	-1.2990	0.9423
8	3	2	1	3	0.7736	0.7750	0.7757	0.7749	0.7749	0.7748	-2.2160	0.5777
9	3	3	2	1	0.7346	0.7229	0.7753	0.7382	0.7423	0.7427	-2.5913	0.4432

an iterative search of parameters based upon the estimation profiles. Desirability for every combination of the full-factorial set of parameters is generated. Table 5.5 lists parameter values and predicted desirability of the top ten most desirable combinations. The parameter configuration of [A=3, B=1, C=2, D=2] was identified as the predicted best combination to achieve the desired output.

Table 5.5: Top ten configurations based on sorting on highest desirability

ID	A	B	C	D	Predicted mean	Predicted SNR_{Tag}	d
1	**3**	**1**	**2**	**2**	**0.8742**	**-1.1363**	**0.9865**
2	2	1	2	2	0.8713	-1.1890	0.9790
3	3	3	2	2	0.8626	-1.2121	0.9587
4	2	3	2	2	0.8597	-1.2648	0.9456
5	3	1	3	2	0.8522	-1.4090	0.9017
6	3	1	1	2	0.8519	-1.4147	0.8998
7	2	1	3	2	0.8494	-1.4616	0.8853
8	2	1	1	2	0.8490	-1.4674	0.8834
9	3	1	2	3	0.8432	-1.4671	0.8706
10	3	3	3	2	0.8406	-1.4848	0.8613

5.6.1 Confirmation Experiment

The predicted best testing configuration was not one of the original configurations of the L9 orthogonal matrix. Therefore, a confirmation experiment was run implementing the suggested configuration in order to verify performance. Table 5.6 shows the results of these tests and comparison to predicted performance. The confirmation experiment indicated that resulting mean FOM was 1.50% less than predicted, SNR_{Tag} was 14.52% worse than predicted, and desirability was 4.58% less than predicted. The predictive capability of the model is most limited with regards to the SNR_{Tag}. This maybe attributed to the inherent random nature of GA. One of the criticisms of the GA for cognitive radio applications is that it can produce different results every time.

The solution recommended from the Taguchi analysis produced a FOM of 0.8612 and is graphed in Kiviat form in Figure 5.5. This chart shows the balanced performance compared to the Figure 5.4. The combination of the qualitative assessment of this chart along with the overall FOM metric provide an additional approach to comparing performance. While a single measure, such as fitness or FOM, provides an frame of reference for comparing, the visual cues provided by the Kiviat provide a way to quickly identify deficiencies or imbalance in performance. While the literature provides some general guidelines for initializing the GA configuration parameters, this exercise showed that there is dramatic performance differences within the suggested ranges. The Taguchi method was able to identify a generally good setting of these four configuration parameters from just 9 unique test cases. To test all combinations it would have required 81 specific test cases. The systematic nature and efficient use of test cases makes this an attractive alternative to the typical ad-hoc nature or brute force approach to setting initialization parameters.

Table 5.6: Confirmation experiment results

ID	A	B	C	D	run-1	run-2	run-3	run-4	run-5	mean	SNR_{Tag}	d
Predicted	3	1	2	2						0.8742	-1.1363	0.9865
Confirmation	3	1	2	2	0.8602	0.8619	0.8607	0.8604	0.8611	0.8609	-1.3014	0.9413
Difference										-1.50%	-14.52%	-4.58%

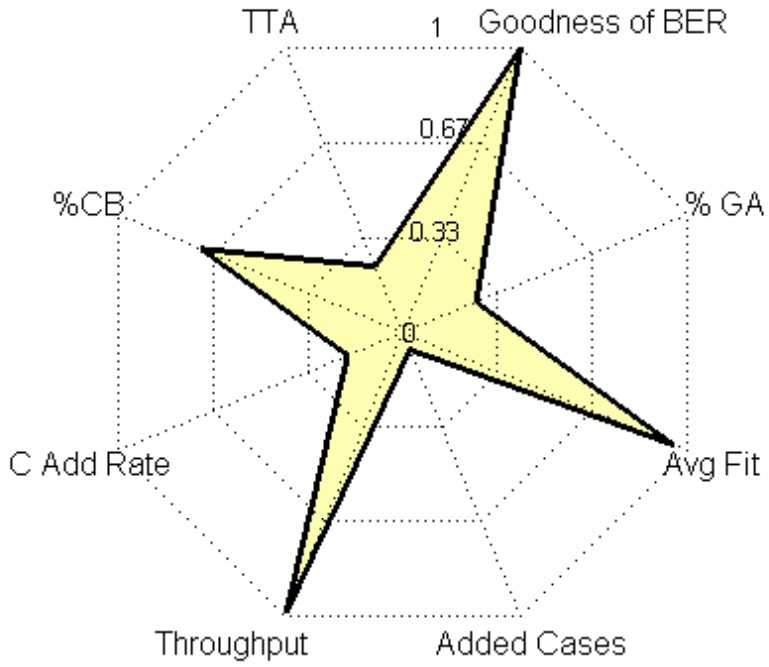


Figure 5.5: Kiviat Graph with FOM=0.8612

5.7 Summary

This chapter presented an application of the Taguchi method that utilized fractional factorial testing matrices for identifying near-optimal configuration parameters of a GA utilized within a cognitive radio engine. Testing methodologies like these provide an efficient means for assessing performance without testing across the entire range of potential configurations. Results indicate that a nine fold decrease in the number of required experiments elicited a predicted configuration parameter set that fell within 1.50% of the actual FOM performance metric. While unsuitable for drawing conclusions on true optimality, these methods provide general trends of performance and provide a tradeoff between minimizing testing costs and knowledge gained on system performance.

Further research suggested includes comparing this analysis to one utilizing RSM to develop an overall model of system performance and confirming the state of interaction effects between parameters. Additionally, this effort focused on a single module within the cognitive architecture. A similar experimental design that considers interaction and performance effects of varying configuration parameters across multiple modules simultaneously, such as the CBR and the GA, will provide

even greater insight into performance prediction. In general, I would suggest taking the concept of metric SNR and incorporating them into a more traditional RSM design as an additional output metric. While there is an increase in cost, in terms of number of test, the ability to identify factor interactions, factor significance, and identify a more accurate performance model offset the increase testing requirement.

Chapter 6

Grey Systems Theory Application to Wireless Communications[1]

The main focus of this dissertation has been experimental design application to cognitive radio that includes factorial design, response surface methods, and the Taguchi method. This emphasis on testing and experimental design was pursued based on the Committee's recommendation at the preliminary exam. However, prior to the preliminary exam, my research investigated similarity algorithms in case based reasoning. The grey relational analysis (GRA) algorithm was identified as a unique approach to quantifying vector similarity. Unlike traditional similarity approaches, which quantify each case's similarity independently of the others in the database, GRA incorporates a relational element between a case and the entire database. My research developed a unique approach to automatic modulation classification (AMC) using GRA that led to a conference paper [55] and invited journal paper [1]. It is acknowledged that this chapter on a GRA-based AMC approach is peripheral to the main focus of this dissertation. Nevertheless, there is a synergy between the core concepts of GRA and experimental design.

Further review of literature revealed a connection between experimental design and GRA. At its

[1]This work was originally published in [1]. With kind permission from Springer Science+Business Media: Analog Integrated Circuits and Signal Processing, Grey systems theory application to wireless communications, vol. 69, pp. 259-269, ©Springer Science+Business Media, LLC 2011

core, GRA is method for comparing a vector against a reference vector. The goal is to quantify similarity between the two. This capability was identified for use within the Taguchi method to assist with robust parameter design [17, 18]. GRA's comparative abilities provided a unique mechanism for performing multi-criteria optimization to identify a recommended solution based on results of a Taguchi analysis. As described in Chapter 5, Taguchi analysis yields several metrics including mean and Taguchi signal-to-noise ratio. Some method of combining multiple responses is required to identify a single solution. This dissertation used utility approaches in Chapter 4 to approach the problem of combining multiple outputs.

GRA provides an alternate method for combining multiple outputs from the Taguchi analysis. In this case, each Taguchi output was normalized to between 0 and 1, and the ideal reference values were set to 1. For example, consider a configuration input that results in a pair of Taguchi responses that have been normalized to between 0 and 1. Together they make a vector in the form of (x,y). An ideal reference is then set at (1,1). As described in Section 6.3, the GRA method identifies the configuration input that yields a result that is most similar to the reference.

6.1 Introduction

As the demand for spectrally efficient and intelligent communication systems such as software-defined radio and cognitive radio grows, the automatic modulation classification becomes an important task for recognizing the modulation format of the received signal in an intelligent receiver. AMC algorithms are mainly composed of two steps: signal preprocessing and classification. The classification algorithms are generally classified into either likelihood-based or feature-based [56]. The feature-based algorithms use features of a signal that are distinct for each modulation type. The pattern recognition or the decision theory-based algorithms are then used for classification [57, 58, 59].

One of the problems with existing approaches, such as neural networks, is the reliance on training. A new approach is presented here that does not require training or *a priori* knowledge in order to perform the modulation classification. This chapter describes a GRA approach to AMC that uses the alpha profile (α-profile) as a signal feature. The proposed algorithm then employs GRA

to identify the highest similarity against a library of reference signals. The modulation type of the signal is then classified based on this similarity measure. This proposed algorithm is similar to Lin's algorithm [60] used for electrocardiogram (ECG) heartbeat recognition which showed high accuracy and fast processing time.

The structure of this chapter is as follows: First an overview of the grey systems theory and the grey relational analysis algorithm is presented. This includes customizing the algorithm with normalization and weighting techniques. Additionally, some limitations of the approach are discussed. Next, the methodology is described and results reviewed. Finally, the paper summarizes with recommendations for future research.

6.2 Overview of Grey Systems Theory

This section overviews Grey Systems Theory and the Grey Relational Analysis (GRA) algorithm that will be utilized as one of the methods for quantifying similarity in case retrieval. This section also introduces a synergistic application of GRA towards automatic modulation classification (AMC). Grey Systems Theory is family of theoretical concepts and practical algorithms developed in China over 20 years ago [61]. The practical applications extend across many disciplines, including economics, ecology, transportation, and wireless communications [62]. The epicenter of the work in this area is in the Far East, however it is seeing growth among the global research community. The originator, Professor Deng, developed the theories to investigate problem sets which contain few data points. These poor information systems are known as Grey Systems. The information theory perspective is that a totally known system is considered white and a completely unknown system is considered black.

The theories created a method for studying problems with few data points and poor information, essentially known as Grey information where a totally known system would be considered white and a completely unknown system black [61]. Utilizing the Grey Systems algorithms to manipulate the original observed data is said to whiten the data by philosophically moving information about the data set towards a more known state. The theoretical aspects of Grey Systems Theory are not as mathematically founded like other theories such as Hidden Markov Models and its use has

been much more empirically based. However, an indicator of it success has been adoption within products and manufacturing.

The Grey Theory family includes GRA which is designed to provide a quantification of multidimensional distance between an observed data vector and a reference vector. This distance quantification provides an indicator of similarity. GRA sets itself apart from other distance measures due to a relational aspect where the measure of similarity is directly related to all the data in the library. This relational aspect of the algorithm sets it apart from other non-statistical vector methods designed to quantify similarity by enabling the ability to merge dissimilar signals into the reference libraries. GRA has been used for clustering, pattern recognition, and matching a signal to known references.

6.3 Basic GRA Algorithm

This section describes the GRA algorithm. It can be used in CBR as a measure of similarity between a new situation and past history in the case library. In a modulation classification scenario it is utilized once features have been extracted from a waveform. This section presents a version developed in [63]. Several variations of the basic algorithm also exist [64]. Figure 6.1 describes the basic methodology. First, let the observed test sequence consist of n sample points $\varphi_i(0), i = 1, 2, 3, \ldots, n$ and a library of K reference vectors $\Phi(k) = [\varphi_1(k), \varphi_2(k), \cdots, \varphi_n(0)], k = 1, 2, 3, \ldots, K$. Equations (6.1) and (6.2) show Φ_{test} as the observed vector and Φ_{comp} as the comparative set of reference vectors. For this project, each vector contains feature(s) extracted from the test and reference signals.

$$\Phi_{test} = [\varphi_1(0), \varphi_2(0), \cdots \varphi_i(0), \cdots, \varphi_n(0)] \tag{6.1}$$

$$\Phi_{comp} = \begin{bmatrix} \Phi(1) \\ \Phi(2) \\ \vdots \\ \Phi(k) \\ \vdots \\ \Phi(K) \end{bmatrix} = \begin{bmatrix} \varphi_1(1) & \varphi_2(1) & \cdots & \varphi_i(1) & \cdots & \varphi_n(1) \\ \varphi_1(2) & \varphi_2(2) & \cdots & \varphi_i(2) & \cdots & \varphi_n(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \varphi_1(k) & \varphi_2(k) & \cdots & \varphi_i(k) & \cdots & \varphi_n(k) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \varphi_1(K) & \varphi_2(K) & \cdots & \varphi_i(K) & \cdots & \varphi_n(K) \end{bmatrix} \tag{6.2}$$

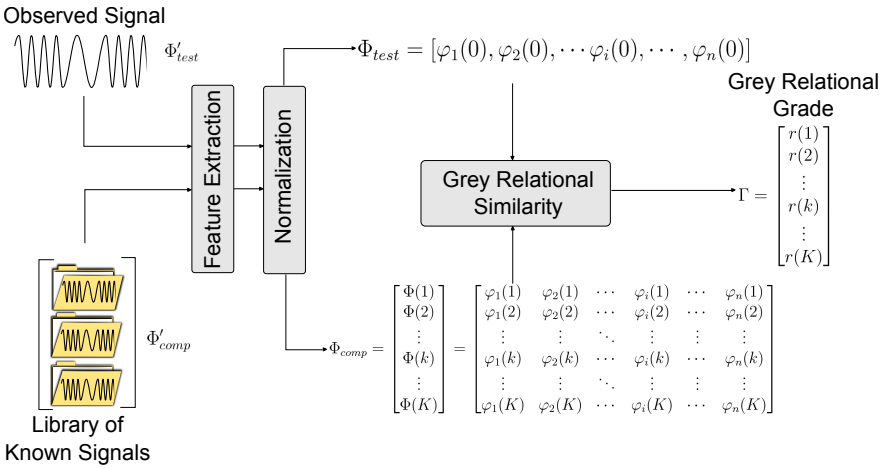


Figure 6.1: In the GRA, an observed signal is processed through a defined feature extraction. A library of reference signals is run through the same extraction and both are processed through a normalization. The GRA similarity measure quantifies and ranks the vector distance between the observed signal and all the signals in the reference library. The Grey Relational Grade provides the ranking of which signal the observation is closest to.

The absolute value of the difference between the test sequence and the k^{th} row of the comparable matrix of reference vectors is calculated in (6.3). The Euclidean distance between the test vector and each comparable vector is calculated and shown in (6.4). Most implementations of GRA utilize Euclidean distance, however an open area of research is utilizing other distance metrics listed previously in place of Euclidean distance.

$$\Delta\varphi_i(k) = \mid \varphi_i(0) - \varphi_i(k) \mid \tag{6.3}$$

$$ED(k) = \sqrt{\sum_{i=1}^{n} (\Delta\varphi_i(k))^2} \tag{6.4}$$

The next step is to calculate quantification of rankings known as the grey relational grade, as shown in (6.5),where the coefficient $\xi \epsilon [0, 1]$. This coefficient is used to minimize or weaken the effect of $\Delta\varphi_{max}$ as it grows large. In the literature, ξ is typically set to $\frac{1}{2}$. We will evaluate variable rates of this element to see the affect. The calculations for $\Delta\varphi_{max}$ and $\Delta\varphi_{min}$ are shown in (6.6) and (6.7).

$$r(k) = \exp\left[\xi \left(\frac{ED(k)}{\Delta\varphi_{max} - \Delta\varphi_{min}}\right)^2\right] \tag{6.5}$$

$$\Delta\varphi_{min} = \min_{\forall k}[\min_{\forall i} \Delta\varphi_i(k)] \tag{6.6}$$

$$\Delta\varphi_{max} = \max_{\forall k}[\max_{\forall i} \Delta\varphi_i(k)] \tag{6.7}$$

The Grey Relational Grades, $r(k)$, provide a quantification of similarity between the test vector and the library of vectors in the case history or library of reference vectors. The higher the relational grade, the higher the similarity and the closer the vector is in terms of distance from the reference vector. A relational grade of 1 corresponds to an exact match. Recall, that this similarity metric is a representation of distance away from the reference signal. The power of the GRA is that it provides a multidimensional quantification of distance combined with a relational element encompassing all the signals in the reference library. These grades are placed into a matrix, Γ, shown in(6.8).

$$\Gamma = [r(1), r(2), r(3), \ldots, r(K)] \tag{6.8}$$

6.3.1 Weighted GRA

Each column within the representation vector has the potential to influence the similarity more or less than another. A weighting process can be introduced through refining the rankings with a weighting factor w_{kj}. The weighting factor can take on a binary representation of a 1 or 0 or it can take on fuzzy values. One of major research questions relates to the methodology for the weighting factors. Methods range from equal weighting of each element to distance based weights, correlated weights, linear, nonlinear and maximal weighting [65]. Other methods in the literature utilize optimization through heuristics like genetic algorithms [64] or particle swarm optimization [66].

6.4 Limitations of GRA

The GRA algorithm presents a unique tool for analyzing similarity. Like any theory it has its limitations that are important to note. The GRA method requires a preprocessing step that normalizes the elements in the matrix. There are several methods for normalization and these will need significant investigation. Finally, the GRA methodology is not as well researched as other

artificial intelligence methods such as artificial neural networks (ANN) and hidden markov models (HMM) and lacks some of the mathematical rigor that HMM's are founded upon.

6.5 Methodology

6.5.1 Feature Extraction

Alpha profile (α-profile) will be selected as the tool for feature extraction because it is capable of discerning differences between digital modulations. The α-profile is defined in (6.9).

$$\alpha - \text{profile} = \max_{f}[\mathrm{C}_X^{\alpha}(f)] \tag{6.9}$$

The term $\mathrm{C}_X^{\alpha}(f)$ is the spectral coherence function (SCF) of the function $x(t)$. This function is defined in(6.10)

$$\mathrm{C}_X^{\alpha}(f) = \frac{S_X^{\alpha}(f)}{\sqrt{S_X^0(f+\alpha/2)S_X^0(f-\alpha/2)}} \tag{6.10}$$

$S_X^{\alpha}(f)$ is defined as the spectral correlation density (SCD) of the $x(t)$. This is given as

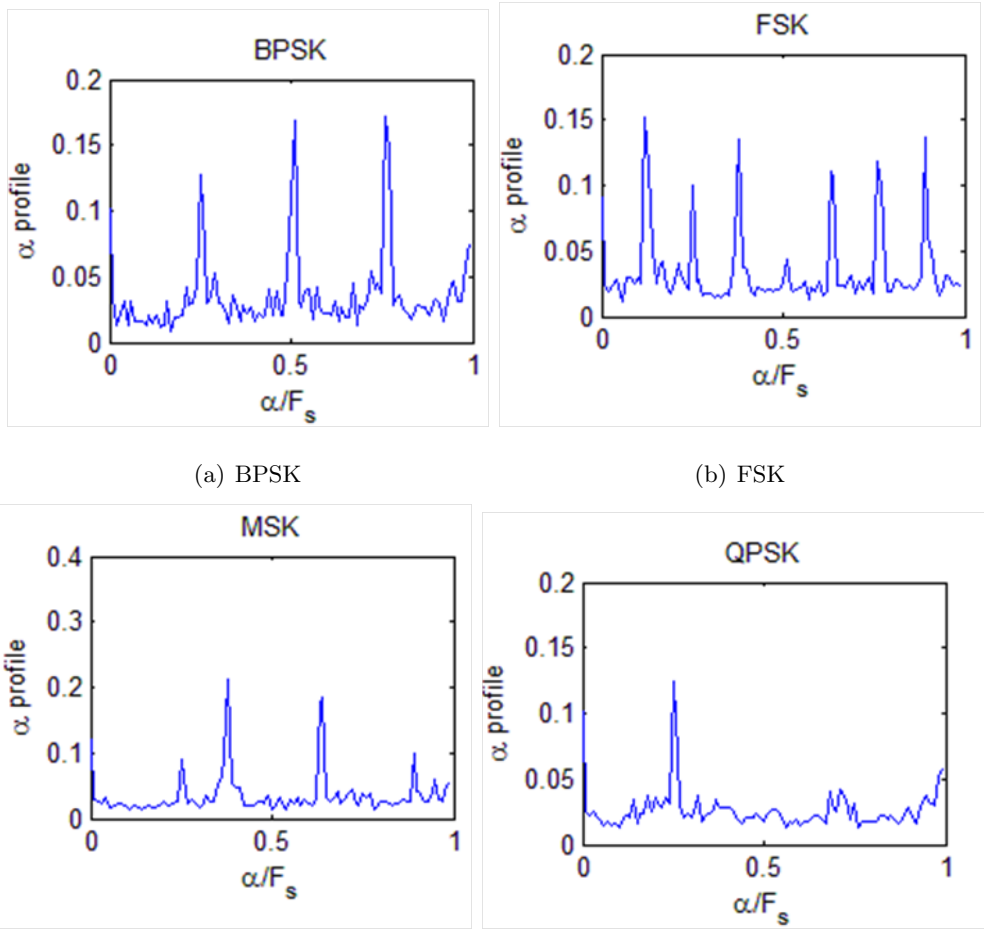
$$S_x^{\alpha}(f)_{\Delta t} = \frac{1}{\Delta t}\int_{-\Delta t/2}^{\Delta t/2}\frac{1}{T}X_T(t, f+\frac{\alpha}{2})X_T^*(t, f-\frac{\alpha}{2})dt \tag{6.11}$$

Where $X_T(t, f)$ is designated as the local Fourier transform of $x(t)$ and α is a cycle frequency or amount of shift.

6.5.2 Enhanced GRA Classification

This section describes the enhanced GRA classifier which utilizes the GRA algorithm presented in [63] together with normalization and weighting methodologies. Other variations of the GRA algorithm are also available [64]. The calculation of α-profiles was performed using MATLAB scripts used in [67].

First, the α-profile of each modulated signal with no noise is generated and converted to a data vector. The library of these profiles, called reference vectors, is used to compare with the α-profile of the received noisy signal, called an observed test sequence, for the classification. Due to



(a) BPSK (b) FSK

(c) MSK (d) QPSK

Figure 6.2: The α-profile provides a method for feature extraction that enables some level of differentiation between the tested modulation types.

limitation in the a-profile where the higher order QAM and higher-order PSK exhibit the same features as QPSK [25], four modulation types are considered: binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), frequency-shift keying (FSK) and minimum-shift keying (MSK). Figure 6.5.2 shows the α-profiles of BPSK, QPSK, FSK, and MSK. These modulation schemes were simulated using MATLAB with 2000 symbols at a baud rate of 100 Hz, sampled at 500 Hz. The α-profiles were calculated with 100 discrete cycle frequency values and a carrier frequency to sampling frequency ratio of $F_C/F_S = 0.2$.

The set of a-profiles given in Figure 6.5.2 is the library of reference sequences, Φ'_{comp} as given in

(6.2). The α-profile of the received noisy signal is the observed test sequence, Φ'_{test} as given in (6.1). Next, each sequence is normalized and weighted as discussed below.

For each normalized vector (α-profile), the weighting vector is then calculated and multiplied with the corresponding normalized vector as shown in (6.12) and (6.13).

$$\Phi_{test} = W(\Phi(0)_{norm}) \times |\Phi(0)_{norm}| \tag{6.12}$$

$$\Phi_{comp} = W(\Phi(k)_{norm}) \times |\Phi(k)_{norm}| \begin{bmatrix} \varphi_1(1) & \varphi_2(1) & \cdots & \varphi_i(1) & \cdots & \varphi_n(1) \\ \varphi_1(2) & \varphi_2(2) & \cdots & \varphi_i(2) & \cdots & \varphi_n(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \varphi_1(k) & \varphi_2(k) & \cdots & \varphi_i(k) & \cdots & \varphi_n(k) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \varphi_1(K) & \varphi_2(K) & \cdots & \varphi_i(K) & \cdots & \varphi_n(K) \end{bmatrix} \tag{6.13}$$

The normalization and the weighting methodology are pre-processing techniques used to quantify the importance of each element in the α-profile based on its magnitude. This allows the distinguishable characteristics (peaks) of the α-profile to have more influence over the final decision. Maximum Value (MV), "Higher is Better" (HiB), and statistical normalization, are normalization techniques used in GRA where $x_i^0(k)$ is the original sequence. MV normalization is shown in (6.14) and HIB shown in (6.15). Statistical normalization is shown in (6.16) where μ is the mean and σ the standard deviation of the sequence.

$$x_(k) = \frac{x_i(k)^0}{max_i(x_i^0(k))} \tag{6.14}$$

$$x_i(k) = \frac{x_i^0(k) - min_i(x_i^0(k))}{max_i(x_i^0(k)) - min_i(x_i^0(k))} \tag{6.15}$$

$$x_i(k) = \frac{x_i^0(k) - \mu}{\sigma} \tag{6.16}$$

A weighting function defined by (6.17) and shown in Figure 6.3 enhances the importance of peaks by relating the weight directly to the magnitude of the α-profile. In this instances, a threshold value of 0.25 was chosen based on observations indicating that unimportant elements within the α-profile has magnitude underneath 0.25. However, a statistical threshold is more appropriate if the receiver is meant to be flexible and adaptable to the variable radio environment. Equation

(6.17) is also very close to a hard-decision threshold so a statistical threshold of $\mu + \sigma$ was chosen. If a peak appears above this threshold, it will be given a weight of 1, otherwise the value of the α-profile at that frequency will be 0.

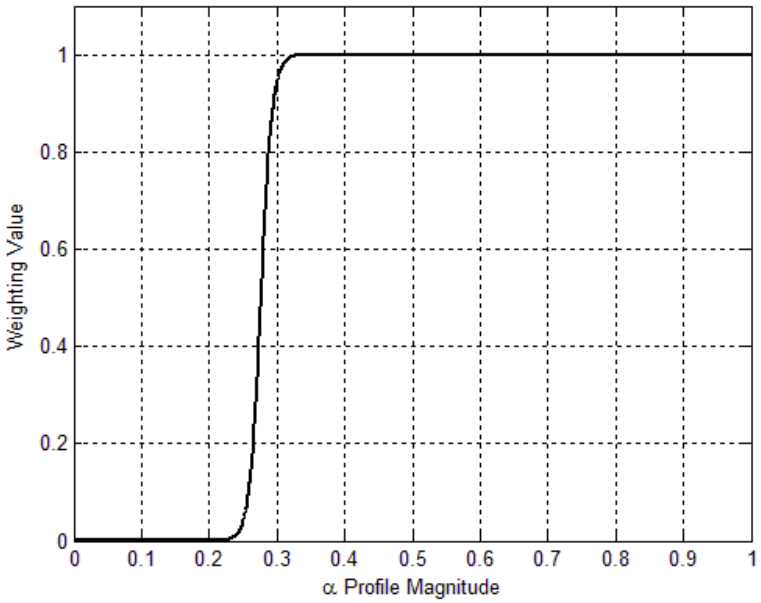


Figure 6.3: Weighting Function

$$W(x) = \frac{1}{2}\left\{\tanh\left((|x| - 0.25) \cdot 60 - 1\right) + 1\right\} \tag{6.17}$$

The final step is to calculate Grey relational coefficients and Grey relational grades as given in Equations (6.3) through (6.8). The classification output of the proposed GRA-based AMC is the modulation type which corresponds to the reference sequence that yields the highest value of the Grey relational grade (the highest similarity between the α-profile of the received noisy signal and the library of the α-profiles shown in Figure 6.5.2.

The following section discusses the results of the GRA-based AMC for all of the previously-discussed normalization and weighting methods in an effort to determine and select the combination that yields the best GRA-based AMC performance.

6.6 Results

The proposed enhanced GRA classifier outputs the modulation type after it compares the normalized and/or weighted α-profile of the received noisy signal with the normalized and/or weighted α-profiles of those four modulation formats. The computer simulations performed to test the GRA-AMC assumed a zero-mean AWGN channel with no fading, no frequency or phase error, perfect synchronization and rectangular pulse shaping with *a priori* knowledge of baud rate and carrier frequency. Each signal consisted of 10,000 data samples corresponding to 2,000 sent symbols. Each simulation consisted of 1,200 Monte-Carlo trials (300 for each modulation) for each SNR value ranging from -20dB to +15dB in 5dB increments. A confusion matrix was then produced for each SNR. The received signal, for which the modulation is to be classified, is modeled as (6.18) where $n(t)$ is additive white Gaussian noise with power σ_N^2. The signal-to-noise ratio (SNR) is then defined by (6.19), where E_S is the signal energy.

$$r(t) = s(t) + n(t) \tag{6.18}$$

$$SNR = 10\log_{10}\left(\frac{E_S}{\sigma_N^2}\right) \tag{6.19}$$

In real-world environment, coarse estimates of the signal characteristics such as carrier frequency and signal bandwidth can be determined from the estimated power spectral density. Baud rate can be estimated via a tracking loop, and symbol timing can be recovered with fractional sampling schemes [68].

The first element in each of the four α-profile reference vectors used in this work had the highest magnitude when the noise was high. This high peak occurs at cyclic frequency $\alpha = 0$ where there is an autocorrelation of the signal, according to (6.10) and (6.11). So to further focus on the differences between the α-profiles, the effect of ignoring this first peak and normalizing and/or weighting, the remaining abridged α-profile was investigated. The average of probability of correct classification P_{cc} was taken over all four modulation types to summarize the performance of the four-modulation classifier.

The results of using the complete α-profile (including $\alpha = 0$) for the three previously mentioned normalization schemes are shown in Figure 6.4. Figure 6.5 illustrates the results of using an abridged α-profile where the first ($\alpha = 0$) element of the α-profile vector is omitted. This shows that the

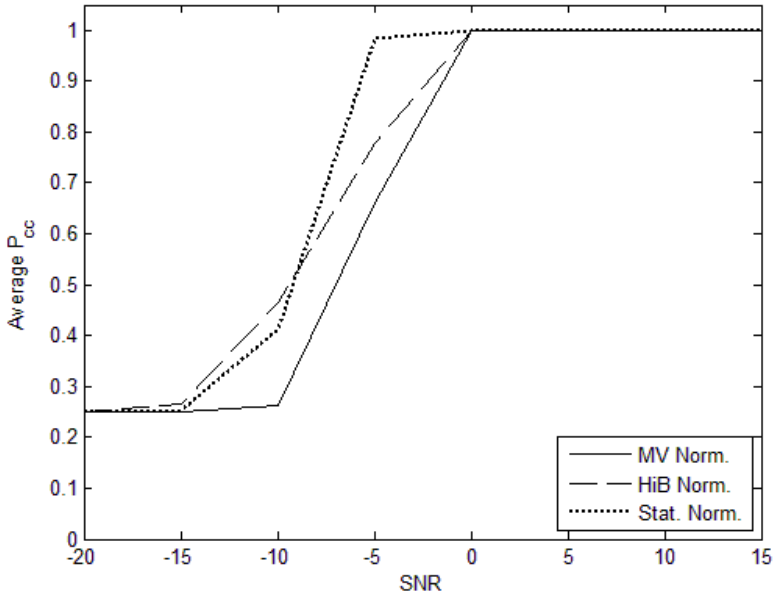


Figure 6.4: Average P_{cc} Normalization Comparison Using Complete α-profile

abridged α-profile results in a better performance for all three normalization schemes. Of the three normalization schemes, statistical normalization also consistently has the best performance.

Weighting the α-profile yielded better results for MV and HIB normalization as shown in Figure 6.6, however weighting was detrimental to statistical normalization performance, as shown in Figure 6.7. Again, using the statistical normalization without weighting had the best result of all combinations.

Statistical normalization with no hard-decision weighting had the best performance overall. However, Figures 6.4 through 6.7 do not address a case of false positives when a signal is classified incorrectly.

To obtain a complete picture of the performance of the classifier, the probability of a false positive should also be tabulated. Since the GRA-AMC must classify the observed received signal as one of four modulations, presenting the overall probability of a false positive will not be useful as it will simply be $P_{fp,Total} = 1 - P_{CC,Total}$, where $P_{CC,Total}$ is the average P_{CC} over all SNR values. Even though a decrease of $P_{fp,Total}$ is important, it is desirable that every modulation does not end up being classified as one particular modulation when the noise floor is raised. Therefore we

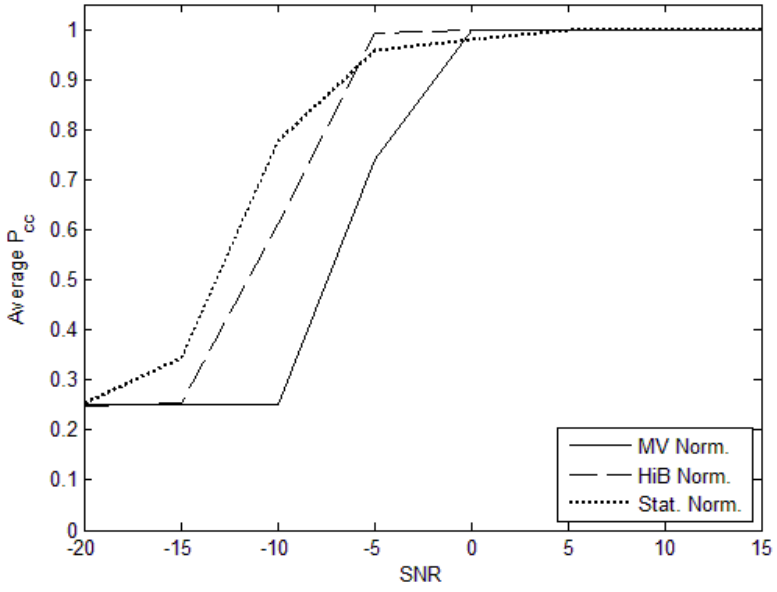


Figure 6.5: Average P_{cc} Normalization Comparison Using Abridged α-profile

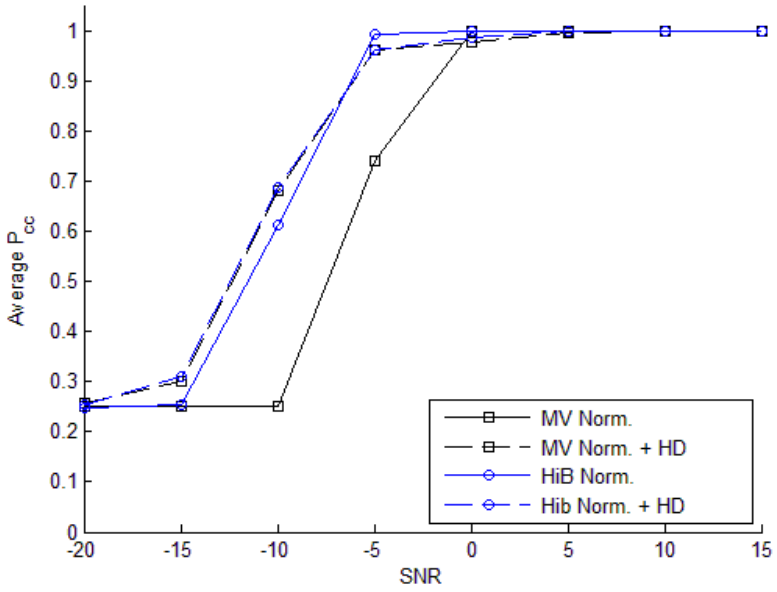


Figure 6.6: MV and HiB Hard-Decision Weighting Comparison

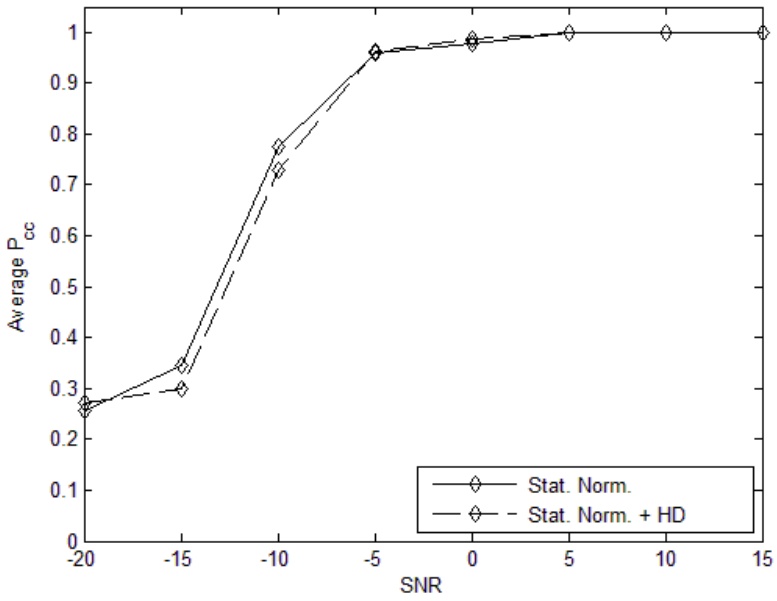


Figure 6.7: Statistical Normalization Hard-Decision Weighting Comparison

instead present the average of the maximum P_{fp} for each modulation, $P_{fp,max}$, out of all tested SNR scenarios. These performance measures of the proposed GRA AMC are all extracted from the tabulated confusion matrices and Receiver Operating Characteristic (ROC) curves.

The ROC curve plots P_{fp} against P_{CC}. The point (0,1) represents perfect classification of the modulation with no false positives while the point (1,1) indicates always classifying the received signal as one particular modulation, whether or not it is correct [69]. The performance of the GRA-AMC is better if the ROC curve falls faster. ROC plots often have a curve that rises steeply and moves towards $P_{fp} = 1$ and $P_{CC} = 1$ [70]. However, this is a characteristic of a receiver that indicates the presence or absence of a signal. Since the proposed GRA-AMC always assumes that there is a signal present, it will always select one of the four possible modulation types, creating a false positive for one of the other three. So, as the P_{CC} of one modulation decreases, P_{fp} of one or all other modulations will increase. The goal is to increase $P_{CC,Total}$ while keeping the maximum P_{fp} of all four modulations as low as possible.

The confusion matrix is a simple tool that can quickly show insight into the classifier behavior. For an ideal classifier all results, each 100%, are contained in the diagonal matrix elements [70]. Figure

6.8 and Figure 6.9 compare the P_{CC} of each modulation against the probability that the GRA modulation choice is a false positive and another modulation was actually transmitted (P_{fp}). To illustrate non-favorable results, Figure 6.9 and Table 6.1 show the performance results of using MV normalization only, with no weighting. For this data preprocessing combination, the probability of classification for BPSK, FSK, and MSK drops off quickly after 0-dB SNR. However, QPSK P_{CC}=100% for all SNR (see Table 6.1). This is only because the GRA determines QPSK to be the most likely choice of the four modulations. QPSK has the fewest number of distinguishing peaks in its α-profile and therefore will have the smallest amount of total distance between the ideal QPSK profile and random noise with no distinguishing peaks. These P_{CC} results are misleading and this is the reason why Figure 6.8 must be shown in juxtaposition. Although it appears that QPSK is always being classified correctly in Table 6.1, the ROC curve indicates the probability of a QPSK false positive approaching 100% at the same time.

In contrast, Table 6.2 indicates worse QPSK classification at low SNR. However, not only do the other modulations have a higher probability of classification, but the P_{fp} values in Figure 6.9 are not dominated by a single modulation as in Figure 6.8.

The ideal plot should show all four modulation type curves as close to 100% P_{CC} as possible while maintaining a low P_{fp}. A stark contrast can be seen between MV and Statistical normalization. MV Normalization, as shown in Figure 6.8, shows QPSK as having a 100% probability of being correctly classified, but as noise increases, QPSK is still selected as a false positive when a different modulation type is transmitted. An incorrect decision automatically indicates a false positive for another modulation type. As noise increases the probability of a QPSK false positive approaches 100%. QPSK becomes the most likely choice for the GRA since it has the smallest number of distinguishing peaks and therefore the smallest Euclidean distance when compared to an α-profile that is completely dominated by noise.

Statistical normalization significantly decreases the probability of a false QPSK result, as shown in Figure 6.9. Since this decision engine has only four choices, it will be impossible for all four curves to have 0 false positive unless each modulation P_{cc} is also 100% for every trial SNR.

The 1200-trial Monte Carlo simulation results for the proposed GRA-based AMC are shown in Tables 6.1 and 6.2 for different SNRs varying between -5 dB and 15dB. The tables also show the

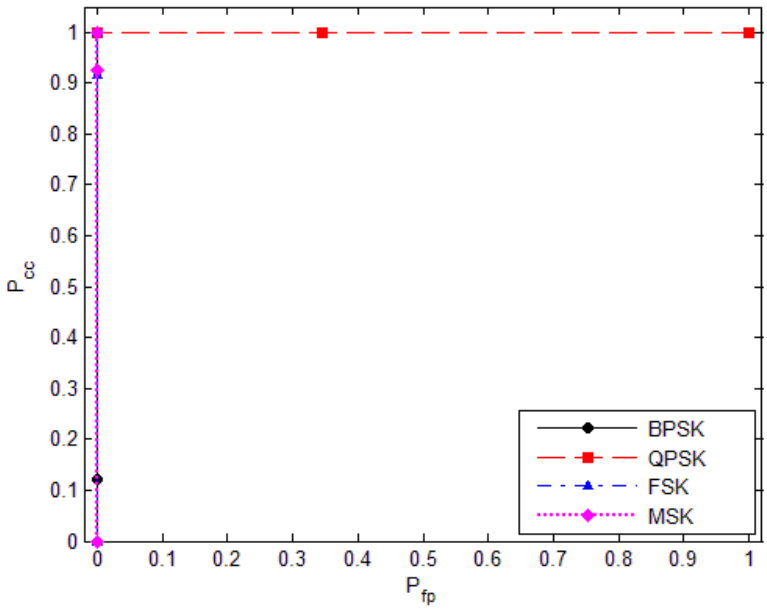


Figure 6.8: P_{cc} vs. P_{fp} - Abridged α-Profile, MV Normalization, No Weighting

modulations that were chosen when a misclassification occurred.

Table 6.1: MV - No Weighting Confusion Matrix

Rx \ Tx	BPSK	QPSK	FSK	MSK
BPSK	1236	-	-	-
QPSK	264	1500	25	22
FSK	-	-	1475	-
MSK	-	-	-	1478

6.7 Discussion/Limitations

We can see in Figure 6.6 and Figure 6.7 that the results are promising especially for SNR above -5 dB. The weighting improves the results for P_{cc} of BPSK, FSK, and MSK at lower SNR when

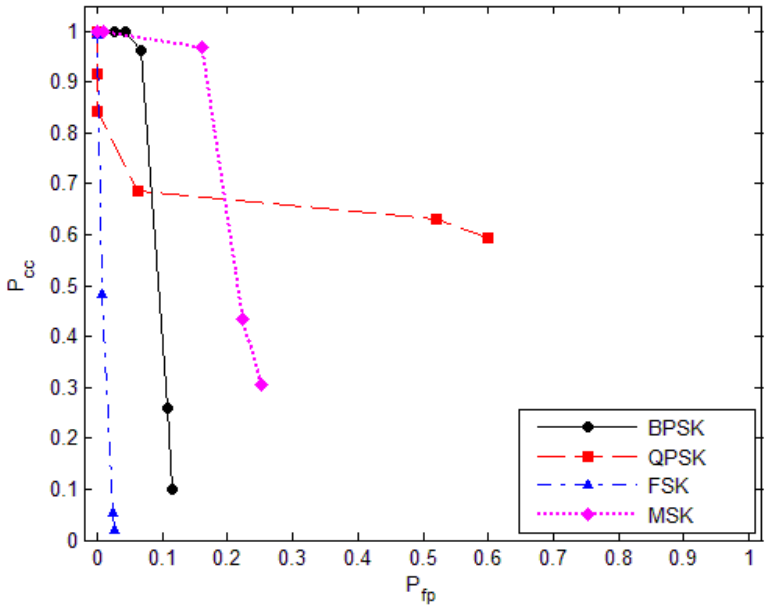


Figure 6.9: P_{cc} vs. P_{fp} - Abridged α-Profile, Statistical Normalization, No Weighting

Table 6.2: Statistical Normalization, No Weighting Confusion Matrix

Rx \ Tx	BPSK	QPSK	FSK	MSK
BPSK	1500	66	-	-
QPSK	-	1427	-	-
FSK	-	-	1475	-
MSK	-	7	2	1500

MV or HIB normalization is used. However, P_{cc} of QPSK was negatively affected in these cases. It is likely that when SNR is low, noise-induced peaks in the α-profile of the received QPSK signal will make it appear to be that of a different modulation type. Statistical normalization with no weighting had the most favorable results.

The decision to use the α-profile as the entire signal feature vector produced the additional need for extensive data preprocessing, yielding a secondary investigation. Normalization methods, weighting schemes and feature abridgment were presented as methods for improving the signal feature before comparison. Combinations of these techniques were then tested. Figure 6.9 illustrated improvements over Figure 6.8 by showing the spreading of false positives across all modulations, instead of one modulation always being classified correctly at the expense of the other three. A successful classifier should not have a bias towards one particular modulation scheme and it was found that using profile abridgment and statistical normalization with no weighting yielded the most favorable results. The false-positive rate must be kept even across the modulations so that a true correct classification can be performed. The current design of the GRA-AMC forces the final decision to be between one of the four modulation types. It is assumed that a signal is present. This means that if a signal is incorrectly classified, the false positive rate automatically increases for one of the other three modulations.

Other methods such as changing the distance metric used by the GRA could be used to increase the P_{cc} while evening out the P_{fp} for all modulations or, a representative GRA choice for channel noise with no signal could be designed to represent that absence of a signal and break apart the linear connection between correct classification and false positive rate.

As SNR decreases, the peaks of the α-profile of the received signal may decrease or change. Distinguishable peaks may still be noticeable, but the Euclidean distance between the observed profile and each reference profile will be large, and therefore a small difference in the noise can easily force the GRA to incorrectly classify the signal. The first step in pre-processing should therefore be normalization in order to have both the reference profile and the observed profile on the same scale. This allows a focus on the relative shape of the profile, and not only the magnitude. A weighting function increases an importance of the peaks by applying a weight that eliminates unimportant elements and maintains peaks of the α-profile.

The simulation results in Table 4 are comparable to those obtained in [57] when a signal carrier and bandwidth can be correctly obtained. The algorithm of Fehske et al. uses the α-profile of signals with the artificial neural network for classification. The proposed GRA-based AMC requires no training and no configuration of architecture and parameters as required in the neural network. However, additional preprocessing algorithms must be incorporated for cases when there is no prior knowledge of the signal or when there are offsets in carrier frequency and symbol rate which make the α-profile different from ones in the library of references especially peak locations.

6.8 Summary

A specific application of GRA towards AMC was presented. The proposed AMC has shown promising classification performance in using the enhanced GRA algorithm to classify four modulation types based on the α-profiles. There are clear advantages to normalization and weighting. Future efforts will include investigating other normalization methods that can improve P_{cc} at low SNR and investigating the effect of substituting different similarity metrics in place of Euclidean distance in the GRA algorithm itself, creating a"modified GRA". The number of modulation types in this chapter is restricted due to the limitation of the α-profiles. Other signal features and their combinations will be investigated in order to classify more modulation types. The combination of several other signal features can be used with few additional computations in the GRA algorithm. Future efforts will also include integrating other preprocessing algorithms and/or signal features for a case when there are offsets in carrier frequency and symbol rate.

Chapter 7

Conclusion

7.1 Summary

The models used in cognitive radio decision making drive the ultimate success of the engine. Current architectures are limited by incomplete mathematical models, reliance on key environmental knowledge, or lack of past experiences. Consequently, there is a need for better approaches to learning system performance in order to make improved decisions, especially in situations not encountered previously. As such, this dissertation investigated improved calibration of existing methods and a statistical experimental design framework that estimates performance models based on empirical observations. The demonstrated benefits of the framework include: 1) enabling an alternate approach to decision making based on empirical estimation models of system performance, 2) providing a systematic method for initializing cognitive engine configuration parameters, and 3) facilitating a deeper understanding of system behavior by quantifying parameter significance and interaction effects.

Chapter 2 provided a tutorial background on experimental design. Starting with basic factorial designs, the chapter introduces the concepts of fractional factorial, Taguchi, and response surface methodology (RSM) designs. The analysis and information afforded by the procedures are summarized including an overview comparison between designs. A reference example provides a data set and resulting analysis to illustrate the concepts.

Chapter 3 discussed new techniques for improving cognitive radio operations. A unique approach for implementing a calibration factor improved performance of existing theoretical estimation formulas for GA operations. Without this calibration, decisions made by the GA in an over-the-air environment were unable to establish a wireless connection. An algorithm was developed that identifies calibration levels within six iterations. Next, statistical methods were presented for application to wireless networks. Examples of response surface methodology and Taguchi design were given and compared using a single performance metric. Results indicate that there is a trade off between the decision quality and the number of tests in a design. A recommended strategy is to screen out any insignificant parameters in order to reduce the number of variables under consideration to three or four in order to perform a response surface design.

Chapter 4 compared the performance of RSM designs and calibrated GA engine under real-world testing conditions. The Box Behnken and central composite design are compared to a reference GA-cognitive engine architecture under five separate interference environments. Results show that statistical fit in static environments is strong, however, it degrades in a dynamic environment. The statistical methods perform within 4% of the adjusted GA approach without reliance on channel knowledge or previous experience.

Chapter 5 implemented the Taguchi method for the initialization of configuration parameters of a GA cognitive engine. In this case, the GA itself is the black box system whose configurable parameters include the crossover rate, mutation rate, population size, and maximum generations. The fractional factorial design were able to reduce the number of tests compared to full factorial design by a factor of nine. The Taguchi estimation was accurate to within 1.5% of actual performance.

Chapter 6 aligned with earlier research on similarity methods for quantifying vector distance in case bases. A unique approach to automatic modulation classification was presented utilizing feature extraction and the grey relational analysis algorithm. A new method for approaching automatic modulation classification was developed. At -5dB or better SNR, the results showed that the GRA approach required no training and only one observation. It was capable of producing similar performance to a neural network approach requiring over 25 observations.

7.2 Significance and Long Term Implications

The methodology identified provides a new way to develop performance estimation models for decision making in cognitive radio. The process also identifies and quantifies significance of parameters, and interaction amongst parameters leading to a better understanding of the overall system. The framework and methods have application within any system that can be distilled to controllable input parameters and measurable outputs. This is a contribution towards general cognitive radio and network systems by providing additional tools for identifying desirable parameter settings.

These findings have a number of key implications to future practice. From a deployment perspective, these methods provide a systematic way for setting up cognitive radio engines in the field. Each deployment site will require customized tuning of cognitive engine configuration parameters. The methodology presented here enables an automated procedure for identifying ideal engine parameters for each deployment. The GA adjustment technique is especially suited for operations in varied environments.

Additional implications tie to simulations of wireless and cognitive networks. Simulations, like the GA, rely on algorithmic models of system behavior. It has been shown that these models do not always match the situation. The performance models developed from empirical testing on over-the-air systems provide potential contribution to improving existing simulations by providing more realistic models for specific situations.

Similarly, the performance models have implications towards the development of traditional rule-based reasoners comprised of 'if then' statements. It was shown that the statistically generated performance models produce basic equations of a performance in terms of each configuration parameter. Unlike traditional theoretical models, these models are defined on a normalized coded scale. This coding facilitates easier interpretation of the predictive models, as the model coefficients provide an indication of the relative impact of each parameter on the output. This provides a roadmap for planning rules on how to change parameters in order to elicit a desired change in the output. Weingart illustrated a cognitive radio reconfiguration algorithm with rules based on DOE generated models of performance [42].

7.3 Summary of Limitations

Finally, a number of fundamental limitations need to be considered as detailed in Section 3.7. While the method develops strong models, one must keep in mind that the models are not perfect and are only estimated models. Once the models are developed, the same limitations incurred from any multi-criteria optimization hold. These include the computational and time costs associated with solving the models.

Additionally, the results identified by the methods are tied to the environment in which the data was collected. Therefore, the specific models and solutions identified from one run of a design are not generalizable. Furthermore, the approach is not designed for exhaustive testing or finding anomalies caused by unique combinations of parameters.

The dependency on number of configuration parameters incurs additional limitations as the number of configuration variables increases. The number of tests goes up by a power of 2 for every variable added into the design. Mitigation strategies include performing screening experiments for the sole purpose of identifying and dropping insignificant parameters. Additional strategies include fractional factorial and Taguchi methods which reduce the required tests. Trade offs include replacing continuous values of parameters with discrete values. Therefore, the models will not have the same resolution as with RSM methods.

The limitations realized from increased complexity bring up additional issues with overall scalability. Consider the fundamental system model of a black box of inputs and outputs. Imagine feeding the outputs of several black boxes as inputs into another black box? This creates a difficult problem in how to manage the controllabilty and observability requirements from cascading black box systems. The methods rely on the capability of having control of the input parameters and the ability to measure the output parameters. Changing input parameter at the beginning of the chain will have dramatic effects farther down the line. Furthermore, what happens if one cannot access the metrics at certain points in the network? This makes for a difficult problem, not solving it could have a very important impact on highly scalable cognitive network design and evaluation.

7.4 Recommendations for Future Work

Areas for future research include implementation methods for automating the analysis so that decision can be performed in line with the statistical probing. This will require emphasis on the multi-criteria optimization methods of desirability or utility functions.

Another area warranting further investigation is the scalability issues, as discussed above, in terms of multiple wireless systems or networks. This statistical approach provides guidance for setting specific link performance. However, this does not insure that the overall network will be operating well. Applying the methods at an overall network level maybe challenging, given the large number of variables and internal dependencies between networks. This may present a tie to game theoretic approaches to synergize the selfish goals of each link to the overall network goals [71].

The Taguchi method was implemented in a configuration parameter initialization approach in Chapter 5. An area of open research is applying Taguchi methods to the same problem of profiling wireless link performance in Chapter 4. The Taguchi emphasis on variation around a mean and the concept of Taguchi Signal to Noise Ratio would present different metrics for performance assessment.

A next step would take situation specific empirical models and use them as a direct replacement for the theoretical models within the GA. Similarly, there is an opportunity to tie statistical results into an existing CBR by feeding the statistical solutions into a case library.

This seeding of the case based can be viewed as a type of cognitive radio training. A case entry is fundamentally a description of the environment, the past decision made, and resulting fitness. The results of the statistical analysis of Chapter 5 produced this exact information and can act as a direct input into the library of a case based reasoner as an environmentally specific learning experience.

The results from the statistical profiling become experiences from which the cognitive engine can draw upon. A compelling experiment would be to hand code these suggested configuration parameters into the case library and then rerun the same scenario on a cognitive engine. Ideally, the mapping of the current situation would produce a high similarity to one of these new cases and

draw upon it to provide a new set of configuration parameters to meet the situation.

7.5 Final Thoughts

Hands-on implementation of the USRP software-defined radios has shown that live operations do not always match simulation and theory. Each radio platform, environment, and application create unique situations that are difficult to simulate. In some situations, one can only say, 'It is what it is,' and try and find a way to work around the issue. The statistical tools presented here provide a systematic method for understanding real operations within deployed settings. Even though as researchers we prefer optimal solutions, in practice, "good enough" usually provides performance adequate to meet the current need.

Bibliography

[1] A. Amanna, M. J. Price, and R. Thamvichai, "Grey systems theory applications to wireless communications," *Analog Integr. Circuits Signal Process.*, vol. 69, pp. 259–269, December 2011.

[2] D. C. Montgomery, *Design and Analysis of Experiments.* John Wiley & Sons, 2001.

[3] A. Amanna, M. Gadhiok, M. J. Price, J. H. Reed, W. P. Siriwongpairat, and T. K. Himsoon, "Railway cognitive radio," *IEEE Vehicular Technology Magazine*, vol. 5, no. 3, pp. 82–89, 2010.

[4] G. Hutto, "Design of experiments: Meeting the central challenge of flight test," *American Institute of Aeronautics and Astronautics*, vol. 7607, pp. 6–8, 2005.

[5] "Sdrf cognitive radio definitions," Tech. Rep. SDRF-06-R-0011-V1.0.0, SDR Forum, 2007.

[6] T. W. Rondeau, *Application of Artificial Intelligence to Wireless Communications.* PhD thesis, Virginia Tech, Blacksburg, 2007.

[7] T. Rondeau and C. Bostian, *Artificial Intelligence in Wireless Communications.* Artech House Publishers, 2009.

[8] T. W. Rondeau, B. Le, C. Rieser, and C. Bostian, "Cognitive radios with genetic algorithms: Intelligent control of software defined radios," in *SDR Forum Technical Conference*, (Phoenix), 2004.

[9] A. Amanna, M. Ghadiok, M. Price, J. Reed, W. Siriwongpairat, and T. Himsoon, "Rail-cr: Cognitive radio for enhanced railway communications," in *Joint IEEE ASME Railway Conference*, (Urbana, Illinois), April 2010.

[10] A. He, J. Gaeddert, K. Bae, T. Newman, J. Reed, L. Morales, and C. Park, "Development of a case-based reasoning cognitive engine for ieee 802.22 wran applications," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 37–48, 2009.

[11] A. Amanna, M. Price, S. Bera, M. Ghadiok, and J. H. Reed, "Cognitive architecture for railway communications," in *Proceedings of 2010 ASME Rail Transportation Division Fall Technical Conference*, (Roanoke, VA), Oct 2010.

[12] Y. Posherstnik, M. Totaro, T. Mai, and J. Molnar, "Testing of policy-based dynamic spectrum access radios," in *Military Communications Conference*, 2010.

[13] S. K. Jones, T. W. Phillips, H. L. V. Tuyl, and R. D. Weller, "Evaluation of the performance of prototype tv- band white space devices (phase ii)," Tech. Rep. FCC/OET 08-TR-1005, FCC, 2008.

[14] Y. Zhao, S. Mao, J. Neel, and J. H. Reed, "Performance evaluation of cognitive radios: Metrics, utility functions, and methodology," in *Proceedings of the IEEE*, vol. 97, April 2009.

[15] M. McHenry, E. Livsics, T. Nguyen, and N. Majumdar, "Xg dynamic spectrum access field test results," *IEEE Communications Magazine*, vol. 45, no. 6, 2007.

[16] H. Arslan, "Testing and measurement of cognitive radio and software defined radio systems," in *SDR Forum Technical Conference*, 2007.

[17] J. Kopac and P. Krajnik, "Robust design of flank milling parameters based on grey-taguchi method," *Journal of Materials Processing Technology*, vol. 191, no. 1-3, pp. 400 – 403, 2007. Advances in Materials and Processing Technologies, July 30th - August 3rd 2006, Las Vegas, Nevada.

[18] Y.-M. Chiang and H.-H. Hsieh, "The use of the taguchi method with grey relational analysis to optimize the thin-film sputtering process with multiple quality characteristic in color filter manufacturing," *Comput. Ind. Eng.*, vol. 56, pp. 648–661, March 2009.

[19] M. Proust, *JMP Statistics and Graphics Guide*. SAS Campus Drive, Cary, NC 27513: SAS Inc., 2008.

[20] C. Rao, "Factorial experiments derivable from combinatorial arrangements of arrays," *Supplement to the Journal of the Royal Statistical Society*, vol. 9, no. 1, pp. 128–139, 1947.

[21] A. A. Hulya Bayrak, "On the construction of orthogonal arrays," *Hacettepe Journal of Mathematics and Statistics*, vol. 31, pp. 45–51, 2002.

[22] C. R. Hicks and K. V. Turner, *Fundamental Concepts in the Design of Experiments*. Oxford University Press, 1999.

[23] L. Du, S. Ukkusuri, W. F. Y. D. Valle, and S. Kalyanaraman, "Optimization models to characterize the broadcast capacity of vehicular ad hoc networks," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 571 – 585, 2009.

[24] L. Chen, "Integrated robust design using response surface methodology and constrained optimization," Master's thesis, University of Waterloo, 2010.

[25] J. Royston, "Some techniques for assessing multivarate normality based on the shapiro-wilk w," *Applied Statistics*, pp. 121–133, 1983.

[26] C. Kim and K. K. Choi, "Reliability-based design optimization using response surface method with prediction interval estimation," *Journal of Mechanical Design*, vol. 130, no. 12, p. 121401, 2008.

[27] M. Wang and B. Bushman, "Using the normal quantile plot to explore meta-analytic data sets.," *Psychological Methods*, vol. 3, no. 1, p. 46, 1998.

[28] W. Rosenkrantz, "Confidence bands for quantile functions: A parametric and graphic alternative for testing goodness of fit," *American Statistician*, pp. 185–190, 2000.

[29] "Nist/sematech e-handbook of statistical methods," tech. rep., NIST, June 2010.

[30] T. Newman, *Multiple objective fitness functions for cognitive radio adaptation*. PhD thesis, 2008.

[31] G. Derringer and R. Suich, "Simultaneous optimization of several response variables," *Journal of Quality Technology*, vol. 12, no. 4, pp. 214–219, 1980.

[32] C. Otero, W. Shaw, I. Kostanic, and L. Otero, "Multiresponse optimization of stochastic wsn deployment using response surface methodology and desirability functions," *Systems Journal, IEEE*, vol. 4, no. 1, pp. 39 –48, 2010.

[33] T. Wagner and H. Trautmann, "Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions," *Special Issue: Preference-based Multiobjective Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 688–701, 2010.

[34] T. Beard, M. Temple, and M. Roberts, "An experimental design approach for optimizing smse waveforms to minimize coexistent interference," in *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 5581 –5585, june 2007.

[35] T. W. Beard, *Applicaion of Optimization Techniques to Spectrally Modulated, Spectrally Encoded Waveform Design*. PhD thesis, Air Force Institute of Technology, 2008.

[36] Y. Cai and D. Liu, "Multiuser detection using the taguchi method for ds-cdma systems," *Wireless Communications, IEEE Transactions on*, vol. 4, pp. 1594 – 1607, july 2005.

[37] A. Awada, B. Wegmann, I. Viering, and A. Klein, "Optimizing the radio network parameters of the long term evolution system using taguchi's method," *Vehicular Technology, IEEE Transactions on*, vol. 60, pp. 3825 –3839, oct. 2011.

[38] J. Hui and M. Devetsikiotis, "Transactions letters - the use of metamodeling for voip over wifi capacity evaluation," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 1–5, 2008.

[39] C. Barrett, A. Marathe, M. V. Marathe, and M. Drozda, "Characterizing the interaction between routing and mac protocols in ad-hoc networks," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, (New York, NY, USA), pp. 92–103, ACM, 2002.

[40] T. Weingart, D. Sicker, and D. Grunwald, "A predictive model for cognitive radio," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1–7, 2006.

[41] T. Weingart, D. C. Sicker, and D. Grunwald, "A method for dynamic configuration of a cognitive radio," in *Proc. SDR '06.1st IEEE Workshop Networking Technologies for Software Defined Radio Networks*, pp. 93–100, 2006.

[42] T. Weingart, G. V. Yee, D. C. Sicker, and D. Grunwald, "Implementation of a reconfiguration algorithm for cognitive radio," in *2nd International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, IEEE, May 2007.

[43] C. Doerr, D. Grunwald, and D. Sicker, "Enhancing cognitive radio algorithms through efficient, automatic adaptation management," in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pp. 1 –5, sept. 2008.

[44] J. D. Gaeddert, *Facilitating Wireles Communications through Intelligent Resource Management of Software-Defined Radios in Dynamic Spectrum Environments.* PhD thesis, Virginia Tech, 2011.

[45] K. Vadde, V. Syrotiuk, and D. Montgomery, "Optimizing protocol interaction using response surface methodology," *Mobile Computing, IEEE Transactions on*, vol. 5, pp. 627 – 639, june 2006.

[46] J. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[47] T. R. Newman, A. He, J. Gaeddert, B. Hilburn, T. Bose, and J. H. Reed, "Virginia tech cognitive radio network testbed and open source cognitive radio framework," in *Proc. 5th Int. Conf. Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops TridentCom 2009*, pp. 1–3, 2009.

[48] A. Amanna, D. Ali, M. Gadhiok, and M. J. Price, "Cognitive radio engine parametric optimization utilizing taguchi analysis," *EURASIP Journal on Wireless Communications and Networking Special Issue - Ten Years of Cognitive Radio: State of the Art and Perspectives*, 2011. Submitted May 2011, in review.

[49] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans, and G. J. Minden, "Cognitive engine implementation for wireless multicarrier transceivers," *Wireless Communications and Mobile Computing*, vol. 7, no. 9, pp. 1129–1142, 2007.

[50] K. De Jong, *Analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan Ann Arbor, 1975.

[51] J. Schaffer, R. Caruana, L. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proceedings of the third international conference on Genetic algorithms*, pp. 51–60, Morgan Kaufmann Publishers Inc., 1989.

[52] C.-F. Tsai and K.-M. Chao, "The contingent design for the optimal parameter settings of genetic algorithms," in *Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on*, pp. 217 –222, 2008.

[53] M. Rezende, A. Costa, R. Maciel Filho, P. Bártolo, and R. Rezende, "A systematic procedure to set up the genetic algorithm parameters for large scale systems: application to a three-phase catalytic reactor," *Chemical Engineering Transactions*, vol. 11, pp. p. 827–832, 2007.

[54] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Wiley New York, 1991.

[55] A. Amanna, R. Thamvichai, and M. Price, "Grey systems theory applications to communications," in *SDR Forum Technical Conference and Product Exposition (SDR'10)*, (Hyatt Crystal City, Arlington VA, USA). Submitted June 2011, in review.

[56] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, 2007.

[57] A. Fehske, J. Gaeddert, and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *Proc. First IEEE Int. Symp. New Frontiers in Dynamic Spectrum Access Networks DySPAN 2005*, pp. 144–150, 2005.

[58] K. Kim, I. A. Akbar, K. K. Bae, J.-S. Um, C. M. Spooner, and J. H. Reed, "Cyclostationary approaches to signal detection and classification in cognitive radio," in *Proceedings of IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212–215, 2007.

[59] K. Maeda, A. Benjebbour, T. Asai, T. Furuno, and T. Ohya, "Recognition among ofdm-based systems utilizing cyclostationarityinducing transmission," in *Proceedings of IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 516–523, April 2007.

[60] C.-H. Lin, "Classification enhancible grey relational analysis for cardiac arrhythmias discrimination," *Medical Biological Engineering Computing*, vol. 44, pp. 311–320, 2006.

[61] Y. Lin and S. Liu, "A historical introduction to grey systems theory," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 2403–2408, 2004.

[62] Z. Liu, Y. Cao, and Z. Xu, "Face recognition based on grey relation metric," in *WRI Global Congress on Intelligent Systems*, vol. 4, pp. 307–310, May 2009.

[63] C.-H. Lin, Y.-C. Du, Y.-F. Chen, and T.-S. Chen, "Multiple ecg beats recognition in the frequency domain using grey relational analysis," in *Proceedings of the 28th IEEE EMBS Annual International Conference*, (New York City), 2006.

[64] S.-J. Huang, N.-H. Chiu, and L.-W. Chen, "Integration of the grey relational analysis with genetic algorithm for software effort estimation," *European Journal of Operational Research*, vol. 188, no. 3, pp. 898–909, 2008. doi: DOI: 10.1016/j.ejor.2007.07.002.

[65] C.-J. Hsu and C.-Y. Huang, "Improving effort estimation accuracy by weighted grey relational analysis during software development." IEEE Computer Society, 2007.

[66] C.-Y. Chen, H.-M. Feng, and F. Ye, "Self grey particle swarm optimization patterns clustering algorithm," *Journal of Grey Systems*, vol. 17, no. 4, pp. 327–336, 2005.

[67] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *Circuits and Systems Magazine, IEEE*, vol. 9, no. 2, pp. 27–45, 2009.

[68] A. Swami and B. Sadler, "Hierarchical digital modulation classification using cumulants," *Communications, IEEE Transactions on*, vol. 48, no. 3, pp. 416–429, 2000.

[69] M. Barreno, A. A. Crdenas, and J. D. Tygar, "Optimal roc curve for a combination of classifiers," in *In Advances in Neural Information Processing Systems*, vol. 20, pp. 57–64, 2007.

[70] P. Prakasam and M. Madheswaran, "Digital modulation identification model using wavelet transform and statistical parameters," *Journal of Computer Systems, Networks, and Communications*, vol. 2008, p. 6, 2008.

[71] J. Neel, *Analysis and Design of Cognitive Radio Networks and Distributed Radio Resource Managament Algorithms*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2006.

[72] A. Amanna, D. Ali, M. Gadhiok, M. J. Price, and J. H. Reed, "Statistical framework for parametric optimization of cognitive radio systems," in *SDR Forum Technical Conference and Product Exposition (SDR'11)*, 2011.

[73] T. R. Newman and J. B. Evans, "Parameter sensitivity in cognitive radio adaptation engines," in *Proc. 3rd IEEE Symp. New Frontiers in Dynamic Spectrum Access Networks DySPAN 2008*, pp. 1–5, 2008.

[74] P. Balister and J. H. Reed, "Usrp hardware and software description," Tech. Rep. 9, Virginia Tech, 2006.

Appendix A

Experimental Results

[1] This section includes experimental results from performing a RSM statistical analysis on SDR universal hardware driver gain settings and transmit power software gain. To goal was to identify how significant each was to output power and packet errors during a file transfer. Additionally, the analysis identified the interactions between software gain and UHD transmit power gain.

A.1 Introduction

SDR and cognitive engine systems have several levels of configurable inputs, although most of the focus is on SDR inputs, such as transmit power amplitude, modulation, or coding. These other initialization inputs also need to be defined prior to operation. For example, GA modules require definition of mutation rate, population size and cross over rate. Similarly, an SDR, contains several initialization configuration parameters that are set outside of the cognition loop. On a USRP these include the programmable gain amplifier settings of the transmitter and receiver. Typically, these settings are identified in an ad hoc nature through trial and error.

More sophisticated techniques, such as parameter sensitivity, vary one input parameter while holding others constant in order to identify recommended settings [73]. However, these approaches cannot identify interaction between input parameters or quantify statistical significance to the outputs. In experimental design analysis, an interaction between two input parameters is identified

[1] Portions of this chapter have been previously published in [72]

when the behavior of one factor is inconsistent if the other factor is changed. For example, consider two factors, A and B. When factor B is set to its minimum, factor A produces a downward trend in output as A is changed from its minimum to its maximum. If this same trend was observed when B was set to its maximum then there would be no interaction between A and B. However, if the output were to trend up when B was set to a maximum, then there is an interaction between A and B.

When considering the input configuration parameters of an wireless link, one can estimate performance in an analytical basis. A link budget would consider amplitude of the transmitted signal, transmit gain, receive gain, and estimated losses in the system in order to predict error rates. The transmit signal and transmit gain can be viewed together as transmit power where power is proportional to the transmit gain times the amplitude squared. Bit error performance can be analytically estimated based on this signal power.

However, there are limitations in a purely analytical approach when considering hardware implementations of SDR. Hardware often introduces irregularities into performance estimations, especially with SDR platforms such as the USRP. For example, non-linearities in receiver front ends can make setting receiver gain difficult. USRP implementations often let the user set receiver gain settings even though this is traditionally handled internally as part of an automatic gain control operation. A setting that is too low may not amplify the signal enough, while a setting that is too high will increase the noise without appreciably increasing the signal level. Similarly, lack of isolation between a transmitter and receiver on the same unit can lead to leakage of transmit power into the receiver. Furthermore, considering transmit gain and the amplitude of the transmitted signal together can be limiting. This approach fails to consider how changing the two values together may affect system performance.

For these reasons, there is justification to consider approaching setting of these parameters from an experimental design perspective. As discussed previously, experimental design provides insight into how input parameters affect the output in terms of main input factors and second order interactions between parameters. Furthermore, the process quantifies the significance of each parameter. This might provide greater insight into the internal operations or faults within SDR and cognitive radio.

Hence, this section uses the proposed experimental design framework to investigate the relationships

and identify recommended settings of the USRP universal hardware driver (UHD) transmit (Tx) gain, UHD receiver (Rx) gain, and the software gain. The system model uses the *liquid* digital signal processing system, so software gain is referred to as the *liquid* gain which is also the parameter set by the cognitive engine. These parameters are analyzed in a systematic manner in contrast to other approaches. The specific problem focuses on how to identify transmitter and receiver hardware gain settings on an SDR that will lead to low packet and bit errors in operations. These settings also have a tangible effect on the usable dynamic range of transmit power which is set as an amplitude in software by the cognitive engine. Typically one considers dynamic range in terms of receivers. However, strong cognitive radio performance will also depend on the power control range that an SDR is capable of producing. Proper operation of the CE will require proper calibration of this power control range prior to operations. The goal of the statistical approach is to identify a general range for Rx and Tx gain that enables adequate dynamic range of output power without negatively affecting performance metrics such as the BER and PER.

The contributions of this section include extending the framework beyond traditional cognitive radio inputs and using it to identify representative knowledge of SDR performance. Furthermore, the results of the experimental design approach provide greater insight into SDR operations by quantifying input parameter significance and identifying interactions between input parameters. This chapter also introduces several non-traditional visual representations for performance including interaction plots and surface profiles. These visualizations have not seen use in cognitive radio and are valuable new tools for researchers.

A.2 Implementation of Experimental Design for SDR Parameter Initialization

To illustrate the implementation of experimental design, the initialization settings of a USRP SDR gain settings are investigated. The goals included determining interactions between the gains set at the UHD and the more traditional software gain, and the affect of the UHD gains on the available range of output power. These gains are implemented at the SDR hardware by setting a programmable gain amplifier that is placed near the digital-to-analog and analog-to-digital converters internal to the SDR. The concerns are on how much the UHD gains may limit the total range of

transmit output power that the SDR can produce and whether the UHD gains will have a negative affect on packet errors during a transmission. This may be caused by saturation at the receiver if the UHD Tx gain is too high.

A.2.1 System Model

A pair of USRP SDR radios were set up as a transmitter and receiver. The software signal processing, *liquid* DSP [44], provided the typical control over transmit power of the radio in the form of software gain. The three inputs, UHD Tx Gain, UHD Rx Gain, and software gain were manipulated according to the design as listed in the first three columns of Table A.1. The minimum setting for software gain, that corresponds to a -1 in the design was set to -65dB and the maximum was set to -15dB based on the linear operating ranges of the USRP. The UHD Tx gain had a range of -74dB to 0dB and the UHD Rx gain had a -20 to 40dB [74].

After each input was set, one hundred packets, consisting of a 1,500 bytes each, were sent across the link during each test creating a total $1.2e^6$ bit transmission. Performance metrics tracked include packet error rate (PER), goodput (bps), and output power (dBm) as listed in Table A.1. Goodput consisted of the payload from received packets that passed cyclic redundancy checks (CRC). Therefore, dropped packets, packets in error, or overhead in packet headers were not counted towards goodput. If a connection could not be established, then the PER was listed as 1 and the goodput as 0. In order to measure output power, a spectrum analyzer was coupled between the antenna output and the antenna of the transmitter, as shown in Figure A.1. The coupler incorporated an internal 20dB attenuator on the coupled output. An additional external 10dB attenuater was placed in line between the coupler and the spectrum analyzer. This allowed output power to be measured while isolating erroneous readings. Past experience showed difficulty in measuring output power reliably with the analyzer measuring over the air in an uncoupled state.

A.2.2 Results and Discussion

The *SAS JMP* 9.0 statistical software package was used to analyze the data and identify estimation models for PER, goodput, and output power. Table A.3 lists the model coefficients that include the main parameters, combinations of the parameters, and second-order terms. The *p-value* associated

with each coefficient represents a measure of statistical significance, where a value of less than 0.05 is considered significant. Of the main parameters, software gain was strongly significant to all outputs as indicated by *p-value* < 0.0001.

The UHD Rx gain main factor did not have a significant affect on any of the output metrics as indicated with *p-values* of 0.5116, 0.5239, and 0.6323 for PER, goodput and output power respectively. However, the second-order term for UHD Rx gain showed an impact on output power. This was attributed to higher than expected measured output power for the configuration of [0,1,0] which is defined by software gain set to a nominal value, UHD Tx gain set to maximum, and UHD Rx gain set to a nominal value. The measured output power was -45dBm and the predicted output power was -51dBm. All of the design configurations except [0,1,0] indicated strong statistical fit between observed and predicted settings, as indicated in Figure A.2. This figure graphs the observed values against the predicted values. The diagonal line represents a perfect fit where the observed value is equal to the predicted value and the dotted lines are a 95% confidence region. The discrepancy maybe due to hardware irregularities as these devices are are consumer grade. If this data point is not considered in the analysis, then the significance of the second-order UHD Rx gain term goes to 0.7578 indicating that the irregular reading affected the estimation model by attributing some of the change to UHD Rx gain.

In contrast, UHD Tx gain did have a significant affect on output power with a p-value of < 0.0001. It did not affect PER as a main factor. However, the analysis shows an interaction between software gain and UHD Tx gain is indicated by a *p-value*of < 0.0001 for PER and goodput. This highlights that when the two are changed together, a statistically significant effect on the output is generated. Identifying these types of interactions is one of the benefits of an experimental design approach as the input parameters are deliberately changed together. Traditional testing that varies one factor at a time, while holding all other constant, would have difficulty uncovering this.

A non-traditional graphic shown in Figure A.3, illustrates how the two parameters interact. This plot shows two estimates of goodput, one with UHD Tx gain fixed to +1 (blue line) and the other with UHD Tx gain fixed to -1 (red line). The y-axis indicates software gain. Strong interactions between input pairs are indicated when the blue and red lines are non-parallel. Limited interactions between two parameters would elicit less change across the ranges of the input parameters which

manifests itself with parallel lines. However when the red and blue lines cross, it indicates that a change in the variable on the x-axis causes two different reactions when the other variable is at a minimum and at a maximum. This corresponds to the high level of statistical significance assigned to the software gain*UHD Tx gain term for the goodput estimation model.

This relationship is more clearly indicated in the surface profile of goodput, as shown in Figure A.4. When UHD Tx gain and software gain are both low there are high packet error and consequently low goodput. Similarly, when both are high there is also an increase in PER and drop in goodput. Finally, a high UHD Tx gain combined with a low software gain also showed poor goodput performance. This operational profile represents real-world performance which often differs from theoretical models. In this case, the high transmit power in combination with high UHD Tx gain is causing saturation at the receiver.

While goodput is an important metric, the available output power that the radio can produce has bearing on operations. This experiment was performed in an ambient condition with no known outside interference sources. If the radio faced an environment with higher noise or a jamming signal, a potential mitigation would be to increase transmit power. A cognitive engine must be aware of how much output power change is manifested from a given increase in the software gain. This is because the cognitive engine typically only has control over the software gain. Therefore, an important specification of a cognitive radio is the dynamic range of its transmit output power. The problem is that the highest output power range for this system model is afforded when UHD Tx gain is set to a maximum, as listed in Table A.1. If one were to design the system operations for this cognitive radio based on this metric alone without an understanding of the interactions between UHD Tx gain and software gain, performance would be compromised. The previous results indicate that high UHD Tx gain leads to higher packet errors and lower goodput performance. While these results may not be indicative of all radios, the framework presented here provides a set of tools that generalizes to other devices and applications.

Applying this framework from the perspective of both traditional wireless performance, such as PER and goodput, as well as output power range indicates that there is a trade off when selecting UHD Tx gain. One must balance the effect that it will have on packet error as well as on available output power. Figure A.4 shows that as UHD Tx gain increases from 0 to 1, the goodput performance

drops drastically. However, there is relatively little benefit from decreasing UHD Tx gain from 0 to -1. In other words, more performance is lost by increasing UHD Tx gain than is gained by decreasing it. Table A.1 lists the predicted output power range for a given UHD Tx gain. Results show that approximately 8dBm of range is gained in output power by increasing UHD Tx gain from a nominal value of 0 to a maximum of 1. The same decrease in range is seen by decreasing from a nominal value to the minimum of -1. Therefore, for this specific system, the results suggest that a nominal value for UHD Tx gain will provide a balance between goodput performance and output power range.

A.3 Summary

This paper identified a need for more systematic approaches for identifying cognitive radio engine and SDR initialization parameters. This need also applies to general testing methodologies for cognitive systems. An experimental design framework was presented to address these gaps that applies well researched methods for systematically and efficiently probing a system's input variables in order to develop empirical performance models. Analysis of the experimental designs also identifies input parameter significance and interactions between parameters that are difficult to identify by traditional methods. An illustrative example of applying the framework to identifying SDR initialization parameters was capable of quantifying parameter significance and identified an interaction between input parameters that affects both goodput performance and output power range. The framework and accompanying analysis provides researchers with new tools for performing testing, identifying initialization parameters, and learning representative information about the system.

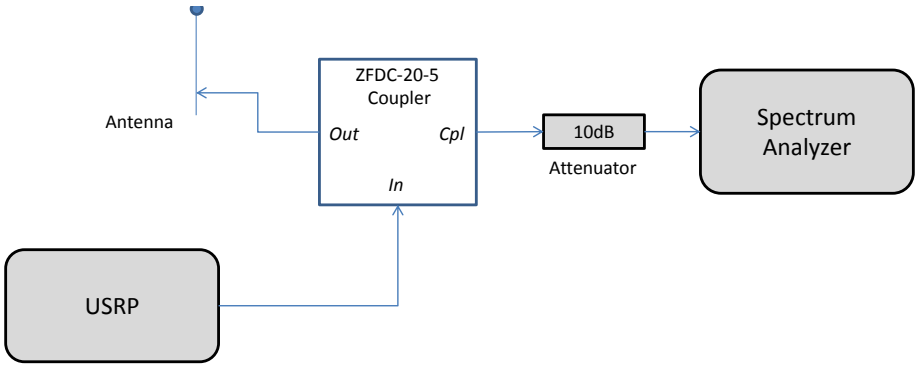


Figure A.1: Experimental setup for measuring output power

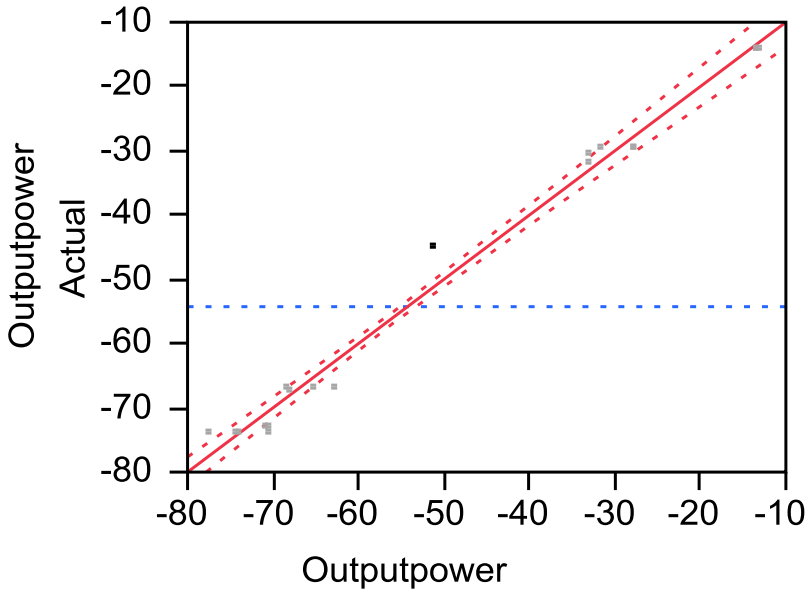


Figure A.2: Observed compared to predicted values of output power

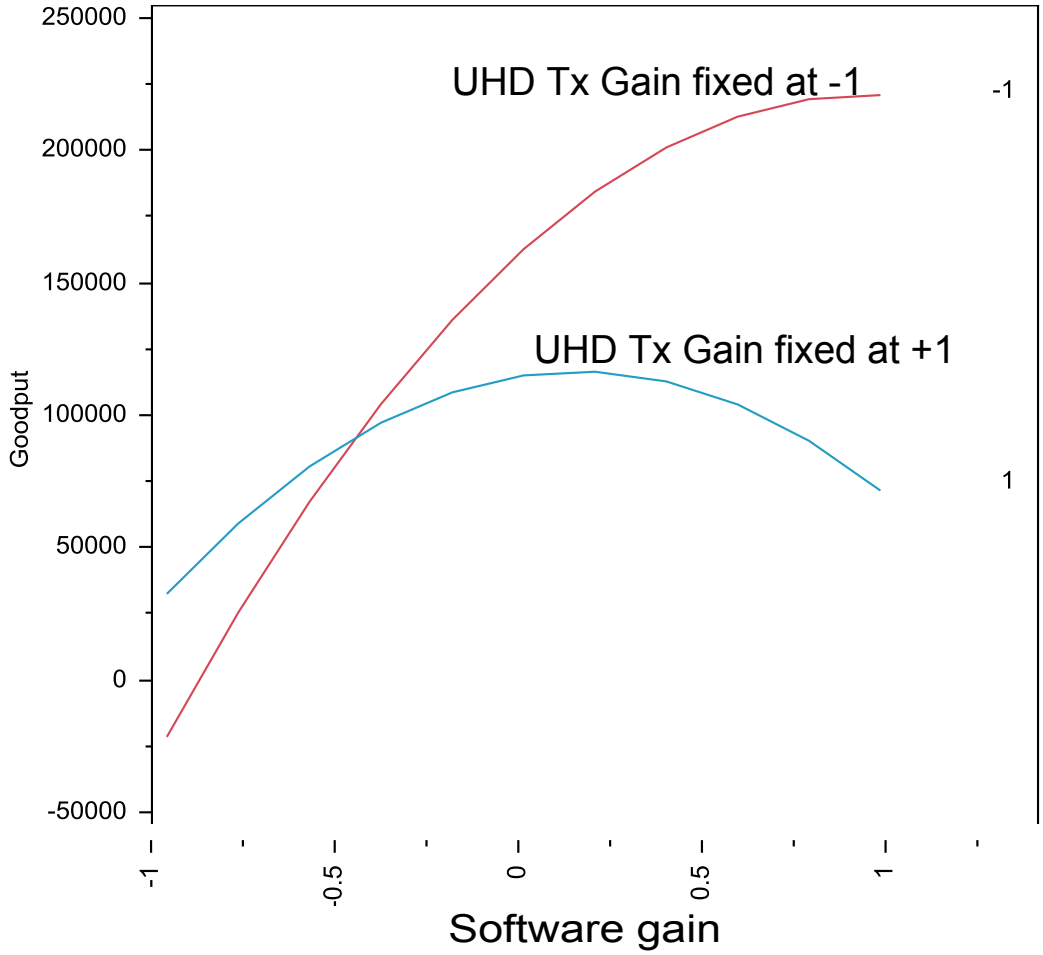


Figure A.3: Interaction between UHD Tx gain(at -1 and +1) and software gain (X-axis) in goodput estimation

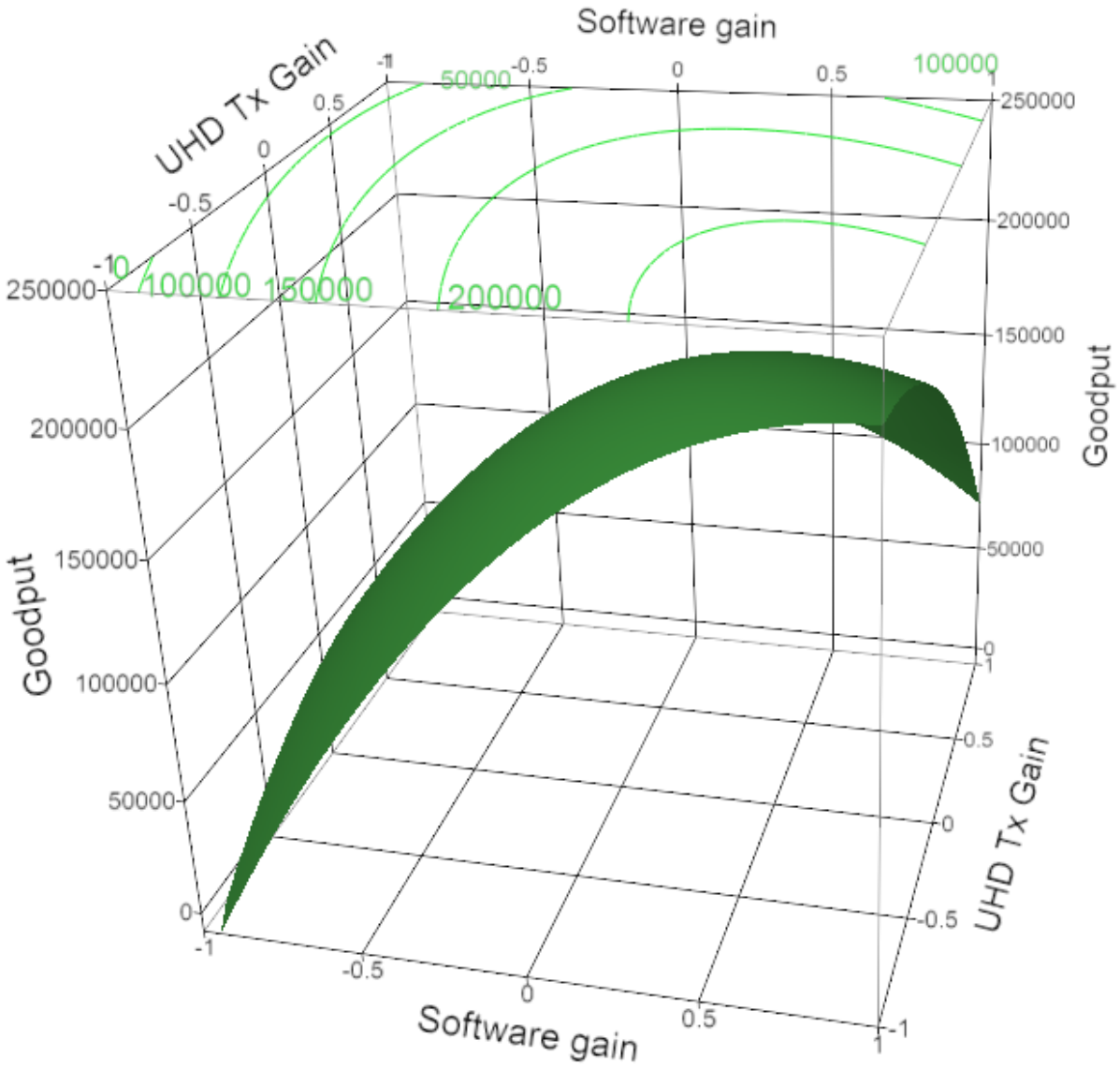


Figure A.4: Response surface profile for goodput

Table A.1: Raw data

Software Gain	Tx Gain	Rx Gain	PER	Goodput	Output Power	Output Power Range
-1	-1	-1	1.000	0	-74.00	41.43
-1	-1	-1	1.000	0	-74.00	41.43
-1	-1	-1	1.000	0	-73.90	41.43
-1	-1	1	1.000	0	-73.85	42.41
-1	-1	1	1.000	0	-73.80	42.41
-1	-1	1	1.000	0	-73.90	42.41
-1	0	0	1.000	0	-73.92	49.74
-1	0	0	1.000	0	-73.86	49.74
-1	0	0	1.000	0	-73.90	49.74
-1	1	-1	1.000	0	-73.80	57.07
-1	1	-1	1.000	0	-72.99	57.07
-1	1	-1	1.000	0	-72.90	57.07
-1	1	1	1.000	0	-72.95	58.04
-1	1	1	1.000	0	-73.00	58.04
-1	1	1	1.000	0	-73.02	58.04
0	-1	0	0.755	50168	-67.20	41.92
0	-1	0	0.714	61475	-67.14	41.92
0	-1	0	0.714	56998	-67.22	41.92
0	0	-1	0.408	130271	-67.12	49.25
0	0	-1	0.367	136541	-67.15	49.25
0	0	-1	0.531	100535	-67.19	49.25
0	0	0	0.265	167081	-67.15	49.74
0	0	0	0.200	183259	-67.20	49.74
0	0	0	0.286	150037	-67.19	49.74
0	0	0	0.204	179976	-67.20	49.74
0	0	0	0.184	184743	-67.22	49.74
0	0	0	0.367	139321	-67.23	49.74
0	0	1	0.122	200677	-67.17	50.22
0	0	1	0.180	177448	-67.22	50.22
0	0	1	0.120	198818	-67.31	50.22
0	1	0	0.000	228693	-45.03	57.55
0	1	0	0.000	218871	-45.12	57.55
0	1	0	0.000	226282	-44.93	57.55
1	-1	-1	0.020	228559	-32.10	41.43
1	-1	-1	0.000	228368	-32.12	41.43
1	-1	-1	0.000	227929	-30.75	41.43
1	-1	1	0.040	215558	-29.60	42.41
1	-1	1	0.000	223247	-29.50	42.41

Table A.2: Raw data (continued)

Software Gain	Tx Gain	Rx Gain	PER	Goodput	Output Power	Output Power Range
1	-1	1	0.060	215577	-29.55	42.41
1	0	0	0.102	205486	-29.80	49.74
1	0	0	0.020	222081	-29.75	49.74
1	0	0	0.000	226557	-29.54	49.74
1	1	-1	0.900	22958	-14.40	57.07
1	1	-1	0.959	13510	-14.10	57.07
1	1	-1	0.939	18175	-14.30	57.07
1	1	1	0.918	22740	-14.08	58.04
1	1	1	0.920	18348	-14.05	58.04
1	1	1	0.918	22653	-14.09	58.04

Table A.3: Coefficients and significance of output models

	PER		Goodput		Output Power	
Term	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
Intercept	0.233	0.0008	170514	< .0001	-65.1985	< .0001
Software gain	-0.307	< .0001	70392	< .0001	24.86867	< .0001
UHD Tx Gain	0.108	0.0149	-23855	0.0192	5.662333	< .0001
UHD Rx Gain	-0.028	0.5116	6274	0.5239	0.257667	0.6323
Software gain*UHD Tx Gain	0.226	< .0001	-50869	< .0001	3.90875	< .0001
Software gain*UHD Rx Gain	0.002	0.9731	-891	0.9353	0.242917	0.6865
UHD Tx Gain*UHD Rx Gain	-0.005	0.9159	1649	0.8806	-0.22375	0.71
Software gain*Software gain	0.297	0.0009	-63048	0.002	12.40351	< .0001
UHD Tx Gain*UHD Tx Gain	0.140	0.0984	-31655	0.1038	8.091839	< .0001
UHD Rx Gain*UHD Rx Gain	0.064	0.4414	-14687	0.4441	-2.99483	0.0065