Performance Analysis of Detection System Design Algorithms

Karl-Johan S. Nyberg

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

Master of Science
in
Industrial and Systems Engineering

Dr. C. Patrick Koelling, Chair
Dr. John E. Kobza, Co-chair
Dr. Ebru K. Bish

March 3, 2003
Blacksburg, VA

Performance Analysis of Detection System Design Algorithms

Karl-Johan Nyberg

(ABSTRACT)

Detection systems are widely used in industry. Designers, operators and users of these systems need to choose an appropriate design, based on the intended usage and the operating environment. The purpose of this research is to analyze the effect of various system design variables (controllable) and system parameters (uncontrollable) on the performance of detection systems. To optimize system performance one must manage the tradeoff between two errors that can occur. A *False Alarm* occurs if the detection system falsely indicates a target is present and a *False Clear* occurs if the detection system falsely fails to indicate a target is present. Given a particular detection system and a pre-specified false clear (or false alarm) rate, there is a minimal false alarm (or false clear) rate that can be achieved. Earlier research has developed methods that address this false alarm, false clear tradeoff problem (FAFCT) by formulating a Neyman-Pearson hypothesis problem, which can be solved as a Knapsack problem.

The objective of this research is to develop guidelines that can be of help in designing detection systems. For example, what system design variables must be implemented to achieve a certain false clear standard for a parallel 2-sensor detection system for Salmonella detection? To meet this objective, an experimental design is constructed and an analysis of variance is performed. Computational results are obtained using the FAFCT-methodology and the results are presented and analyzed using ROC (Receiver Operating Characteristic) curves and an analysis of variance.

The research shows that sample size (i.e., size of test data set used to estimate the distribution of sensor responses) has very little effect on the FAFCT compared to other factors. The analysis clearly shows that correlation has the most influence on the FAFCT. Negatively correlated sensor responses outperform uncorrelated and positively correlated sensor responses with large margins, especially for strict FC-standards (FC-standard is defined as the maximum allowed False Clear rate). Suggestions for future research are also included. FC-standard is the second most influential design variable followed by grid size.

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. John E. Kobza and Dr. C. Patrick Koelling, for guiding me through my first research endeavor. I also want to thank Dr. Sheldon H. Jacobson who helped me through the initial stages of my research before moving to the University of Illinois. They all helped to keep me motivated and focused on the task ahead and provided helpful suggestions and insights when most needed. Additionally, I wish to thank Dr. Ebru K. Bish for taking the time to serve on my committee.

I thank my parents, Majken and Sven Nyberg and their respective life partners Torsten and Carina, for supporting my adventures outside my homeland Sweden and for believing that I would graduate someday. I thank my brothers, Adam Nyberg, Morgan and Markus Palmgren, and my sisters, Mirijam and Jill Nyberg, for believing in their older brother and for forgiving me for not being around to enjoy their company.

Finally, I would like to thank all my friends, co-workers, and fellow students in Sweden, in the DC area, and in Blacksburg that have shown patience and support throughout my research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

.

CHAPTER 1

1 Introduction

Detection systems are concerned with the detection, recognition or classification of objects, using detection device technologies (i.e., *sensors*) arranged in some system network topology (e.g., serial, parallel or mixed). Each sensor can observe an object (or situation) and send a response to a central *fusion center* that makes a global decision based on the individual sensor responses (see Figure 1-1). A *fusion rule* decides how individual sensor responses are combined. In general, an object (or situation) is characterized by any of M possible hypotheses, and the detection system must decide which hypothesis is correct. For example, in a binary case, there are only two possible hypotheses presented, the *null hypothesis* and the *alternate hypothesis*. The alternate hypothesis typically represents the presence of a *target* (i.e., an object that has the characteristics searched for), and the null hypothesis represents the absence of a target.

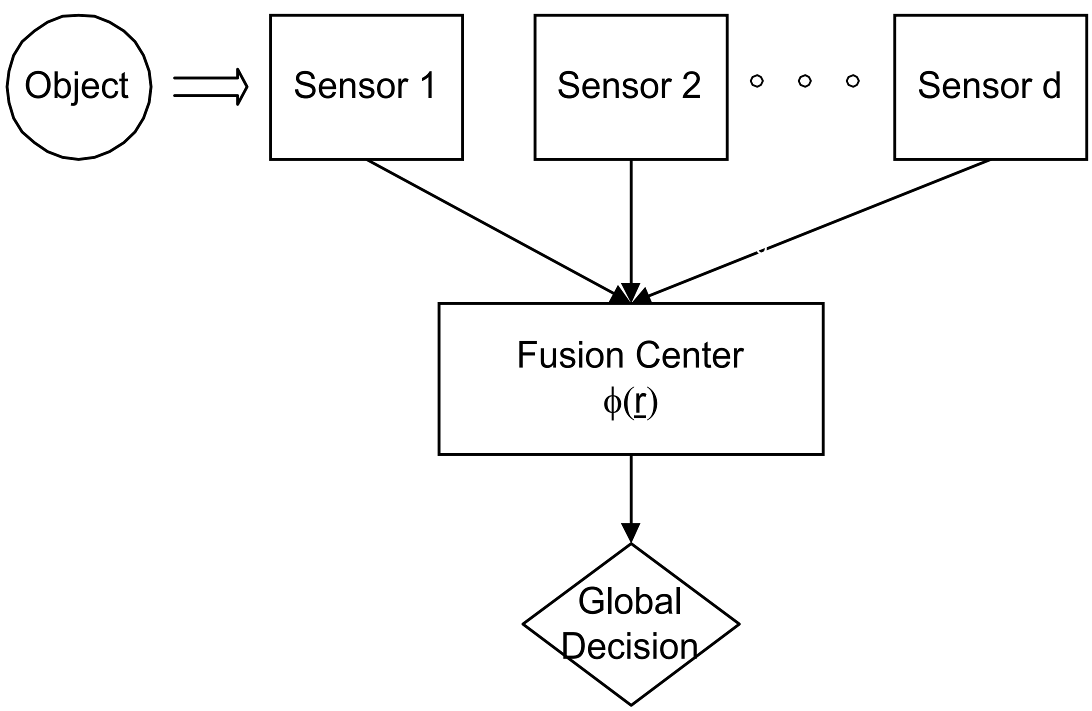

Figure 1-1: ***Example of a Detection System***

Examples of sensors are *X-ray screening devices* in airport access control systems, *radar sensors* in aircraft detection systems, and *biosensors* (i.e., bacterial sensing devices) in Salmonella detection systems. Different sensors or multiple sensors of the same kind are combined, in detection systems, to improve different aspects of system performance (e.g., reliability, speed, survivability, coverage, and likelihood of detection). In addition to multiple sensors, a detection system consists of operational policies and procedures for utilizing the sensors.

Detection systems are widely used in different applications in industry. They are used to detect aircraft, make medical diagnoses (Moore et. al., 1986), control product quality (Christer, 1994), make financial decisions, predict weather, and control airport access. For example, at an airport, baggage and passengers are screened to prevent the admittance of firearms and explosives onto aircraft (Jacobson et al., 2001). In the food industry, producers must detect Salmonella, E.coli, and other organisms before food reaches consumers. Early detection (i.e., before shipping) reduces the cost and damage associated with a batch of contaminated food (e.g., Salmonella contaminated chicken). In the manufacturing industry, non-conforming items must be detected before they are distributed. Failure to detect non-conforming items before shipping could cause delays at other stages in the supply chain, customer dissatisfaction, and expensive corrective actions.

Consider a detection system designed to detect Salmonella (or other potentially fatal bacteria) in food products at a processing plant. The sensors in this system are *biosensors* (i.e., bacterial sensing devices) that are able to detect a particular kind, or several kinds of bacteria to some degree of certainty. Assume that each of d biosensors returns a continuous sensor response value R_i, $0 \le R_i \le 1$, $i = 1, \ldots, d$, to a central fusion center. A global decision is then made, based on the continuous responses from the sensors. The accuracy of this global decision depends on design variables and model parameters such as the following.

- The performance of individual sensors (i.e., device technologies),

- how individual sensor responses are combined to yield a global decision (i.e., format of the fusion rule),
- the number of sensors in the system and how they are arranged (i.e., system topology),
- the level of dependency among sensors in the system (i.e., correlation),
- the sample size (i.e., size of test data set or historical data set) used to estimate the distribution of sensor responses, and
- the estimation procedure (e.g., static grid estimation, Jacobson et al., 2001).

In order to measure the *system performance* (i.e., accuracy of the global decision), consider the possible errors that can occur when making a global decision. A *False Alarm* (or a false positive) occurs if the global decision indicates a *target is present* (e.g., Salmonella in a chicken carcass) when there is *no target present*. A *False Clear* (or a false negative) occurs if the global decision indicates a target is not present when a target is present (Echard, 1991). Both of these errors are undesirable and subject to constraints in the form of safety standards and economic considerations. Unfortunately, because of the relationship between the false alarm and the false clear probabilities, they cannot be simultaneously minimized (Friedl, 1989, Van Trees, 1968). Designers, users, and operators of detection systems are challenged with this false alarm, false clear tradeoff (FAFCT) problem. False alarms decrease throughput and false clears can have serious implications, both economically and politically.

Previous studies have dealt with the problem of finding the optimal tradeoff between the false alarm and the false clear probabilities. Jacobson et al., (2001) developed a method to address the FAFCT problem in an airport access control system environment. They used a mixed probabilistic-deterministic approach for modeling detection systems. More specifically, the problem is addressed by formulating a *Neyman-Pearson hypothesis-testing problem* (Van Trees, 1968), which is transformed into a Knapsack problem and solved using heuristic solution techniques.

This thesis studies the FAFCT problem to understand how different system design variables and parameters affect the performance of a detection system. The method used in this thesis incorporates the method developed by Jacobson et al., (2001), which will be referred to as the FAFCT-methodology. Additions have been made for modeling dependent sensor responses and for presenting the results generated by the method.

Independent and dependent sensor responses are modeled as random variates from given probability distribution functions. The FAFCT is studied for both the dependent and the independent case.

To address the FAFCT using the Neyman-Pearson hypothesis-testing problem, the probability density functions describing the sensor responses must be known or estimated. A static grid estimation procedure developed by Jacobson et al., (2001), is used to estimate these density functions. The static grid estimation procedure uses frequency count on an evenly spaced grid structure to create a discretized representation of the density functions.

The performance analysis involves an experimental design where different design variables (i.e., controllable factors) and model parameters (i.e., uncontrollable factors) are varied to simulate different detection system settings.

Design variables include:

- *Sample size*: size of test data set or historical data set available to use when estimating the joint density functions that describe the distribution of sensor responses, and
- *Grid size*: a measure of how fine (or coarse) the grid structure used in the static grid estimation procedure is. If the grid is too fine, very few data points will be observed in each sub-region. On the other hand, if the grid is too coarse, it cannot capture the form of the density functions.

Model parameters include:

- *Distribution of sensor responses*: defined as two joint density functions (see Figure 1-2) conditional on a target present or no target present, respectively,
- *Dependency among sensor responses*: measured by the correlation coefficient, and
- *FC- standard* (False Clear standard): Maximum false clear rate allowed by the system. For example, the FC-standard set by FAA (Federal Aviation Administration) or its international counterparts in an airport access control system.

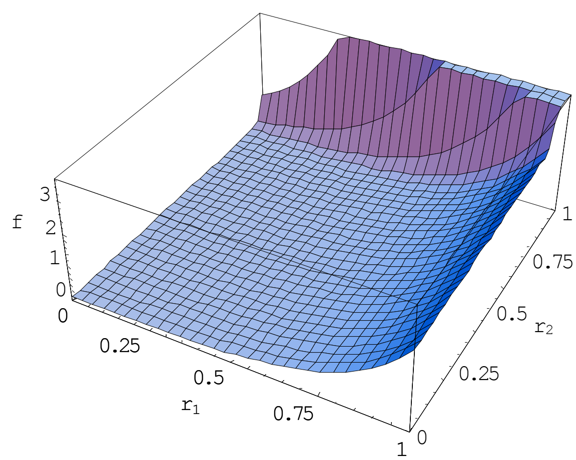

Figure 1-2: ***Example of a joint probability density function, $f(r_1, r_2)$, for a system response conditional on a target is present.***

To present and visualize the performance of different detection system settings, ROC-curves (Receiver Operating Characteristics) are generated and analyzed. ROC-curves are plots of the probability of detection against the probability of false alarm. The ROC-curves provide a convenient tool to present the performance of detection systems.

The thesis is organized as follows. Chapter 2 reviews the literature on detection theory, data fusion and hypothesis testing, as well as the building blocks of the FAFCT-methodology. Chapter 3 presents the methodology used to develop this research. Chapter 4 presents computational results obtained from the experimental design, including an analysis of variance and the resulting ROC-curves. Finally, Chapter 5 summarizes and concludes the work and presents suggestions for future research.

CHAPTER 2

2 Literature Review

Literature related to this research includes probability theory, discrete optimization techniques (i.e., the Knapsack problem), classical and modern detection theory (Van Trees 1968, Varshney 1996), statistical hypothesis testing, and decision fusion (Dasarathy 1994). Applications include airport security systems (Jacobson et al., 2001), quality control systems (Christer 1994), radar detection (Echard 1991), warning systems in risk management (Pate-Cornell 1986), and hypothesis testing in medicine (Moore, Hutchins, and Miller 1986).

The literature review is organized in three sections. The first is intended to give the reader a better understanding of sensors, sensor fusion, detection systems, and their characteristics. These concepts are essential to the research presented in this thesis. The second section describes a detection system model and its components in a mathematical framework. Finally, the third section discusses recent advances in distributed and centralized detection, and identifies the needs for research in this field.

2.1 Sensors, Detection and Fusion

A *sensor* is defined as a device technology, or methodology, that can observe an object or a phenomenon, and based on its observation, either make a decision (binary or multilevel discrete) or give a complete response as output. A complete response is defined as a noise-free continuous response.

Sensor fusion is the study of optimal information processing in distributed multi-sensor environments through intelligent integration of the multi-sensor data (Dasarathy, 1994). The human brain is an example of a very powerful multi-sensor environment. It fuses several different types of signals (including sight, hearing, smell, taste, and touch) received from the five basic human sensors (eyes, ears, nose, tongue, and skin) and makes nearly optimal decisions in real time. One way to divide the very broad area of sensor fusion into smaller groups is by the objective or the purpose of the fusion process (i.e., combining sensor responses). Examples of objectives and purposes include, detecting the presence of an object (e.g., mine detection), recognize a particular object or event (e.g., radar detection of aircraft), classify an object or event (e.g., quality control), or integrate information (e.g., business, financial or security information) to make an intelligent decision. Many real-world applications have multiple purposes for the fusion process. For example, consider the detection of firearms and explosives in an airport access control system. The airport access control system should detect metal content and certain chemical substances, recognize shapes of potential firearms, and classify passengers and baggage in order to make an intelligent decision.

The literature distinguishes between *centralized* and *decentralized* (or distributed) detection. In distributed detection systems (i.e., systems with decentralized hypothesis testing) sensors observe an object, or a phenomenon, and each of the sensors makes a decision based on the individual observations. Local decisions are then distributed to a *fusion center* where combining these decisions, according to some fusion rule, results in a global decision. In centralized detection, sensors observe an object or a phenomenon and distribute unprocessed responses (i.e., complete or near complete responses) to a centralized fusion center. Combining these responses according to some fusion rule results in the global decision.

The purpose of having binary sensor responses (i.e., distributed detection) is often to minimize the amount of data communicated within the system. Communication bandwidth is often restricted in real-world systems. Other reasons why distributed detection systems are preferable in some applications include the following:

- time constraints (i.e., speed in distributing information),
- cost considerations (i.e., cost of distributing large amounts of data),
- reliability (i.e., trustworthiness),
- survivability (i.e., ability of a system to provide adequate service when faced with a fault), and
- coverage (i.e., the scope of the system objective).

The likelihood of detection is higher in a centralized system (assuming the use of sensors that are suited for the application) because the centralized fusion center can process complete responses (no information is lost due to bandwidth considerations) (Tenney and Sandell, 1981). However, a centralized system may not always be possible depending on the intended application. One can argue that a centralized system requires complete sensor responses. However, in reality information can never be classified as complete in this sense. There will always be some form of noise or other constraints present to make total centralization of information impossible.

Sensor responses or local sensor decisions can be either binary, multilevel discrete or continuous. If there are no constraints on communication and processor bandwidths, near complete observations can be brought to the fusion center for data processing. Consider a *distributed detection system* where observations are processed at the local sensors before responses are brought to the fusion center. In this case, some information will be lost. This might be acceptable if speed is more important than accuracy or if there are limitations on communication bandwidth. A distributed detection system where some form of processing is done at the local sensors is also known as a parallel fusion network (Varshney, 1996).

Sensors can be arranged in series, parallel, or a mixed version of the two. Some network topologies also include feedback loops (i.e., information is passed back in the system network for the purpose of improving system performance). For more information on system network topologies with feedback, refer to neural networks or parallel processing networks (Alhakeem and Varshney, 1995, and Haykin, 1994).

Many papers found in the literature assume independent sensor responses. Although independent sensors are easier to model, some level of dependency is often present in reality. Dependency among sensors in a detection system can exist for several reasons. Overlapping regions of coverage of the local sensors is one. In a serial system, observations of an object are determined by the responses from a-priori observations and are therefore dependent (i.e., prior observations decide if the next sensor will observe the object or not). The form of the fusion rule can also induce dependency.

If we have access to a large amount of historical data, the distribution of sensor responses can be estimated. Otherwise, assumptions must be made about the distribution of sensor responses. The system response can be binary as in Figure 2-1 (i.e., Alarm or Clear) or multilevel discrete (e.g., Alarm, Clear or "Further testing necessary"). A continuous system response is a possibility, however not applicable in a fully automated detection system since then; the final decision would still be undecided. A continuous response could be desirable if the final decision is made by a decision-maker or if the system response is used as input, in a detection system of larger scale.

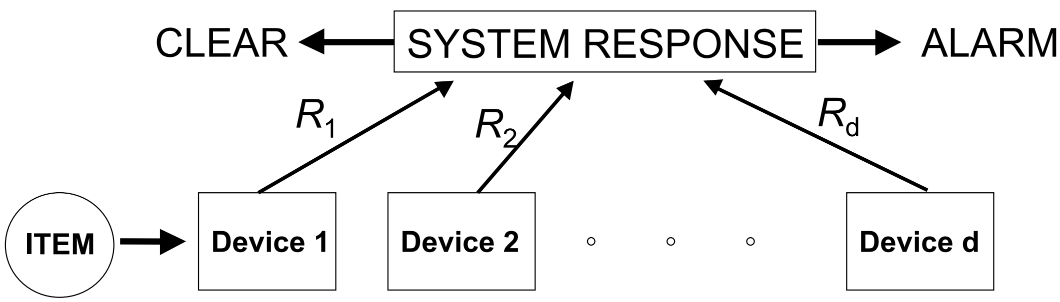

Figure 2-1: ***Detection System model***

2.2 Detection theory and hypothesis testing

Consider a *d*-sensor distributed detection system where each of two sensor responses corresponds to a hypothesis (i.e., the null hypothesis, H_0, or the alternate hypothesis, H_1). In particular, the null and the alternate hypotheses are

H_0 : No target is present and

H_1 : Target is present.

The d different sensors inspect the item and generate d responses: R_i, $0 \leq R_i \leq 1$, $i = 1,\ldots,d$, where a response close to one indicates H_1 is correct (i.e., target is present), and a response close to zero indicates H_0 is correct (i.e., no target is present). Denote this set of d random variables with the vector $\underline{R} = (R_1,R_2,\ldots R_d)$ and denote a particular combination of sensor response values with the vector $\underline{r} = (r_1, r_2,\ldots,r_d)$. Thus, the observation space, Z, is the closed hypercube $[0,1]^d$, and represents the set of all possible values of $\underline{R}$. Each vector of response values $\underline{r}$ represents a point in the observation space, Z, where the values are generated by two joint probability density functions, the joint probability density function conditional on a target, T, being present,

$$p_{\underline{R}|H_1}\left(\underline{r}|H_1\right) = f_{\underline{R}|T}(\underline{r}) = \frac{dF_{\underline{R}|T}(\underline{r})}{d\underline{r}}, \tag{2.1}$$

and the joint probability density function conditional on no target, NT, being present,

$$p_{\underline{R}|H_0}\left(\underline{r}|H_0\right) = f_{\underline{R}|NT}(\underline{r}) = \frac{dF_{\underline{R}|NT}(\underline{r})}{d\underline{r}}. \tag{2.2}$$

$F_{\underline{R}|T}(\underline{r})$ and $F_{\underline{R}|NT}(\underline{r})$ represent the conditional distributions of sensor responses. The two joint probability density functions can be estimated by sampling over various target and non-target items using estimation procedures described in the following subsections.

2.2.1 Static Grid Estimation

One method to estimate the conditional joint probability density functions, (2.1) and (2.2), is the static grid estimation procedure (Jacobson et al., 2001). Classify the response from sensor j into one of n_j evenly spaced cells, each of length $1/n_j$, $j = 0, 1, \ldots, d$. Therefore, $[0, 1]^d$ is decomposed into $\prod_{j=1}^{d} n_j$ d-dimensional hypercubes. Define the notation $(i_1, i_2, \ldots, i_d)$ to denote the hypercube

$$\left(\frac{i_1}{n_1}, \frac{i_1+1}{n_1}\right] \times \left(\frac{i_2}{n_2}, \frac{i_2+1}{n_2}\right] \times \ldots \times \left(\frac{i_d}{n_d}, \frac{i_d+1}{n_d}\right], i_j = 0, 1, \ldots, n_j - 1, j = 1, \ldots, d.$$

Also define $h(i_1, i_2, \ldots, i_d)$ to be the total number of sample points in hypercube $(i_1, i_2, \ldots, i_d)$ and M to be the total number of sample points collected. The algorithm for the static grid estimation procedure to estimate $f_{\underline{R}|T}(\underline{r})$ is as follows (Simms, 1997):

Initialization: Set the values n_j, $j = 1, 2, \ldots, d$.
Set $h(i_1, i_2, \ldots, i_d) = 0$, $i_j = 0, 1, \ldots, n_j - 1$, $j = 1, 2, \ldots, d$.
Set $m = 0$ to be the total number of sample points collected so far.

Sampling: While $m \leq M$ perform the following steps:

Independently sample a vector $\underline{r}$, given a target item. Determine the hypercube in which the point, defined by the $\underline{r}$ vector, falls and increment (by one) the $h(.)$ value associated with this hypercube.

Set $m = m + 1$.

Computation: $h(i_1, i_2, \ldots, i_d) / M$ is an estimator for

$$\int_{i_1/n_1}^{(i_1+1)/n_1} \cdots \int_{i_d/n_d}^{(i_d+1)/n_d} f_{\underline{R}|T}(\underline{r})\, dr_1 \ldots dr_d\,, \quad i_j = 0, 1, \ldots, n_j - 1,\; j = 1, 2, \ldots, d\,.$$

This procedure is repeated for non-target objects to estimate $f_{\underline{R}|NT}(\underline{r})$.

The performance of the static grid estimation procedure is sensitive to how the grid points are chosen and to how many sample points M are collected. In particular, the values for the n_j affect the quality of the estimators for $f_{\underline{R}|NT}(\underline{r})$ and $f_{\underline{R}|T}(\underline{r})$. If the n_j is set too large, the grid will be so fine that very few sample points will be observed in each hypercube. Conversely, if the n_j are set too small, the grid will be too coarse to capture the form for $f_{\underline{R}|NT}(\underline{r})$ and $f_{\underline{R}|T}(\underline{r})$. These tradeoffs exist anytime a static grid is used to estimate a probability density function.

2.2.2 Fusion rule

A *fusion rule*, $\Phi: [0, 1]^d \to [0, 1]$, combines all the local sensor responses and returns a global decision. Using the fusion rule, H_0 (i.e., no target is present) is chosen if $\Phi(\underline{r}) \leq \nu$, and H_1 (i.e., target is present) is chosen if $\Phi(\underline{r}) > \nu$, where $0 \leq \nu \leq 1$ is a threshold value. A system clear occurs if H_0 is chosen and a system alarm occurs if H_1 is chosen. The objective is to find a fusion rule that partitions the observation space, Z, into two sub regions: a clear region, $Z_0 = \{r: \Phi(\underline{r}) \leq \nu\}$, and an alarm region, $Z_1 = \{r: \Phi(\underline{r}) > \nu\}$ (i.e., to find a suitable rule for deciding which hypothesis to select). The decision variables in this design problem are the form of the fusion rule and the value of the threshold.

Consider the binary case where objects conform to one of two hypotheses, the null hypothesis, H_0, or the alternate hypothesis, H_1, then the possible outcomes are:

- A true clear: decide H_0 when H_0 is true,
- A false clear: decide H_0 when H_1 is true,
- A false alarm: decide H_1 when H_0 is true,
- A true alarm: decide H_1 when H_1 is true.

There are two types of errors that can occur. A Type I error occurs when the null-hypothesis is falsely rejected (i.e., a false alarm: decide H_1 when H_0 is true). A Type II error occurs when the null-hypothesis is falsely not rejected (i.e., a false clear: decide H_0 when H_1 is true). These error probabilities provide performance measures for the detection system. More specifically, the performance of a detection system improves as these probabilities decrease. To express these probabilities, define the events:

A = System gives alarm,
NA = System does not give alarm,
NT = No target present,
T = Target present.

Using these events, the Type I and Type II error probabilities are the false alarm, P_F, and false clear, P_{FC}, probabilities respectively, defined as the conditional probabilities

$$P_F = \text{P(Reject } H_0 \mid H_0 \text{ is true)} = \int_{Z_1} p_{\underline{R}|H_0}\left(\underline{r}\middle|H_0\right)dr = \int_{\phi(r)>\nu} f_{\underline{R}|NT}(\underline{r})\,dr = \text{P(A}\mid\text{NT)},$$

and

$$P_{FC} = \text{P(Reject } H_1 \mid H_1 \text{ is true)} = \int_{Z_0} p_{\underline{R}|H_1}\left(\underline{r}\middle|H_1\right)dr = \int_{\phi(r)\le\nu} f_{\underline{R}|T}(\underline{r})\,dr = \text{P(NA}\mid\text{T)}.$$

To illustrate these probabilities, consider an airport access control system where baggage and passengers are screened. It is difficult to estimate the true probability that a bomb or a firearm is about to enter an aircraft (i.e., P(T)). Defining the false alarm and false clear probabilities as conditional probabilities circumvents this problem. The real question is: Suppose that a bomb is about to enter, what is the probability that the system will miss it (i.e., P(False Clear)), or suppose that an innocent passenger is about to enter, what is the probability that the system will indicate an alarm (i.e., P(False Alarm))? The conditional probabilities provide this information.

2.2.3 Neyman-Pearson formulation

Designing a detection system involves selecting an appropriate fusion rule, Φ, and a value for ν so that points, $\underline{r}$, in the observation space, Z, can be mapped to a hypothesis. It would be optimal if the false alarm, and false clear probabilities could both be eliminated. However, in real-world applications this is typically not possible, hence the need for an optimization criterion (used when multiple objectives must be considered in order to make a good decision). To avoid a sub-optimal decision one has to assign rules that decide how much each of the objectives should influence the decision.

In the literature one can find several examples of optimization criteria (Varshney, 1996): Bayes criterion, Minimax detection, Neyman-Pearson criterion, Sequential detection and Constant False Alarm Rate (CFAR) detection. Among the different methods, the most frequently used are the Bayes criterion and the Neyman-Pearson criterion. The Bayes criterion leads to a likelihood ratio test (LRT) and requires known a-priori probabilities P(H_0), P(H_1), and estimated or known costs assigned to the possible outcomes. In cases where it is difficult to assign costs or determine a-priori probabilities, the Neyman-Pearson criterion is a more appropriate alternative, and it will be used in this thesis.

The objective of the Neyman-Pearson criterion is to maximize the probability of detection while minimizing the probability of false alarms. Note that the probability of detection is complementary to the probability of a false clear (i.e., P(True Alarm) = 1-P(False Clear)), and since false clears and false alarms are interdependent they cannot be simultaneously minimized. This leads to the following formulations: Minimize false alarms (or, equivalently, maximize true clears) while keeping false clears to an upper bound, ε_{FC} or Minimize false clears (or, equivalently, maximize true alarms) while keeping false alarms to an upper bound, ε_{FA}. The appropriate formulation depends on the type of application considered. For example, in an airport access control system, false alarms are minimized subject to a common security standard, ε_{FC}, defined by FAA. To see why this is a logical formulation for an airport access control system, consider the effects that false alarm and false clears can cause in a real-world situation. False alarms can cause delays and customer dissatisfaction while false clears can cause, in the case of a bomb explosion on an aircraft, not only large monetary costs but also loss of human life.

The Neyman-Pearson criterion leads to the following mathematical programming formulations:

$$\text{Minimize P(False Alarm)}$$
$$\text{Subject to P(False Clear)} \le \varepsilon_{FC} \tag{2.3}$$

or

$$\text{Minimize P(False Clears)}$$
$$\text{Subject to P(False Alarm)} \le \varepsilon_{FA}. \tag{2.4}$$

The decision variables in these mathematical programs are the fusion rule, $\Phi(\underline{r})$, the threshold value, ν, and the system design (i.e., number of sensors, network topology, etc.). Given a system design, the choice of a fusion rule $\Phi(\underline{r})$ and a value for ν that does not cause the false clear (or false alarm) probability to exceed ε_{FC} (or ε_{FA}) defines the feasible region over which the minimum false alarm (or false clear) probability can be achieved.

2.2.4 Knapsack problem

To address this constrained optimization problem, Jacobson et al., (2001), formulated a decision problem named the False Alarm, False Clear Tradeoff (FAFCT) problem. The observation space, Z (i.e., the closed hypercube $[0,1]^d$), is partitioned into a finite set, I, of non-overlapping subregions (note that the static estimation procedure can be used to create this set). Define n to be the number of sub-regions in I, where each sub-region, $i \in I$, is indexed so that $I = \{i_0, i_1, \ldots, i_{n-1}\}$. Define $q_1(i_j), i_j \in I, q_1 : I \rightarrow [0, 1]$ as the probability that a system response falls into sub-region $i_j, j=0, 1, \ldots, n-1,$ given a target, and $q_2(i_j)$, $i_j \in I$, $q_2 : I \rightarrow [0, 1]$ as the probability that a system response falls into sub-region i_j, $j=0, 1, \ldots, n-1$, given a non-target, where

$$q_1(i_j) = \int_{\{\underline{r}:\underline{r} \in \text{ subregion } i_j\}} f_{\underline{R}|T}(\underline{r})d\underline{r}\,, \tag{2.5}$$

and

$$q_2(i_j) = \int_{\{\underline{r}:\underline{r} \in \text{ subregion } i_j\}} f_{\underline{R}|NT}(\underline{r})d\underline{r}\,. \tag{2.6}$$

Suppose that each sub-region has a *size*, measured by $q_1(i_j)$ and a *utility*, measured by $q_2(i_j)$. These values can be estimated using the proportion of samples that fall into the appropriate sub-region given a target or a non-target item, respectively. Given the sub-region discretization defined by I for $[0, 1]^d$, the goal in addressing the mathematical programming problem in (2.3) is to identify a set of sub-regions I_C, such that $\sum_{i_j \in I_C} q_2(i_j)$ is maximized subject to $\sum_{i_j \in I_C} q_1(i_j) \leq \varepsilon$, where the set of sub-regions I_C represents the values for $\underline{r}$ such that a system clear (i.e., true clear or false clear) occurs following the inspection of an item. The complement of I_C, defined as $I_A = I - I_C$, denotes the set of sub-regions for which a system alarm (i.e., false alarm or true alarm) occurs following the inspection of an item.

The set of sub-regions I_C defines a fusion rule that appropriately bounds the false alarm and false clear probabilities. This fusion rule is dependent on the responses of the local sensors in the detection system, which in turn is described by the conditional probability density functions. Therefore, the solution to the FAFCT problem explicitly defines a fusion rule that meets the false alarm and false clear probability requirements. Jacobson et al. (2001), prove that the FAFCT problem is NP-complete and introduce Knapsack heuristics to solve it, more specifically, a greedy algorithm and a dynamic programming algorithm. A greedy algorithm will be used throughout this thesis.

2.2.5 Knapsack heuristics

To understand how Knapsack heuristics can be applied to the FAFCT problem, consider a bag that needs to be packed with nutritious food for a long hike. The objective is to maximize the nutritional value of the contents (i.e., utility) without exceeding the size (i.e., weight) of the bag, ε. Formally stated, consider the clear region, I_C, to be a knapsack into which some or all of the *n* subregions that form the region *I*, must be placed. The discretized mathematical program in (2.3) can then be written as

$$\text{Maximize} \qquad \sum_{j=0}^{n-1} q_2(i_j)x(i_j)$$

$$\text{subject to} \qquad \sum_{j=0}^{n-1} q_1(i_j)x(i_j) \leq \varepsilon \ ,$$

$$\text{where} \qquad x_j = \begin{cases} 1, & \text{if subregion } i_j \text{ is selected for the clear region;} \\ 0, & \text{otherwise,} \end{cases}$$

$$j = 0,\ldots,n-1.$$

Solving this problem using Knapsack heuristics (e.g., greedy algorithm, dynamic programming) will yield the clear region, I_C, for the given values of $q_1(i_j)$, $q_2(i_j)$, $i_j \in I$, and ε. These heuristics will also return the probability of a non-target item generating a global decision in the clear region (i.e., true clear) and the probability of a target item generating a global decision in the clear region (i.e., false clear). The resulting optimal values for the Type I, and Type II error probabilities are

P_F = 1-P(NA| NT) = 1- $\sum_{i_j \in I_C} q_2(i_j)$ and

P_{FC} = P(NA| T) = $\sum_{i_j \in I_C} q_1(i_j)$, respectively.

2.3 Receiver Operating Characteristic (ROC)

A common way to describe the performance of a detection system is in terms of its Receiver Operating Characteristic (ROC) (Van Trees, 1968). The Bayes criterion or the Neyman-Pearson criterion can be expressed as a Likelihood Ratio Test (LRT). Accept H_0 if $\Lambda(\underline{R}) < \nu$ and reject H_0 if $\Lambda(\underline{R}) > \nu$ where

$$\Lambda(\underline{R}) = \frac{p_{\underline{R}|H_1}(\underline{r}|H_1)}{p_{\underline{R}|H_0}(\underline{r}|H_0)}.$$

The performance of LRT's can be conveniently described in terms of a graph known as the Receiver Operating Characteristic (ROC). This is a plot of the probability of detection, P_D (i.e., 1-P_{FC}), against the probability of false alarm, P_F., for a sensor or a given detection system.

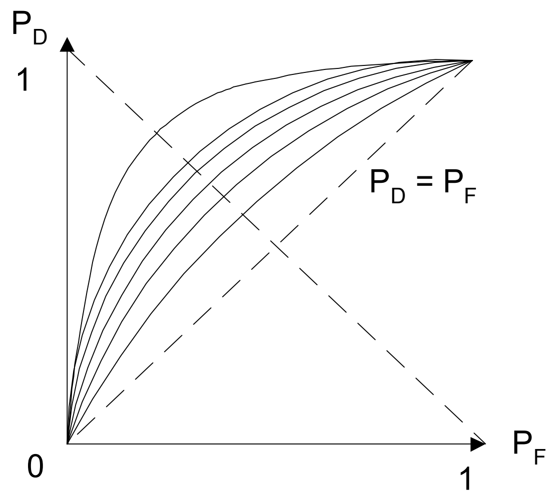

Figure 2-2: ***Receiver Operating Characteristics***

As shown in Van Trees (1968), the ROC's of all continuous LRT's are concave downward and lie above the $P_D = P_F$ line (Figure 2-3). A steeper ROC-curve indicates a better performing detection system (or sensor) because it implies that a higher probability of detection can be obtained with only a small increase in false alarm probability.

2.4 Detection theory and data fusion

The research in this area is extensive, although scattered in many disciplines. The techniques and methodologies used to optimize and analyze distributed detection systems are inherited from signal detection and estimation theory. The first attempt to create a detection theory for distributed sensors, by considering *decentralized hypothesis testing* (i.e., a parallel fusion network or distributed detection systems), was made by Tenney and Sandell (1981). They derived optimum fusion rules at the local sensors by assuming *optimum fusion* at the *fusion center* (i.e., the local sensor responses are optimally combined into a global decision).

A study by Chair and Varshney (1986) complements the original work of Tenney and Sandell by considering the optimization at the fusion center (i.e., optimally combining the local sensor responses into a global decision). They used a Bayesian Likelihood Ratio Test (LRT) to derive an optimal fusion rule for the binary local decisions (i.e., locally processed binary sensor responses) that are assumed optimal.

Reibman and Nolte (1987) generalize the previous work by relaxing the optimality assumptions, both at the local sensors and at the fusion center, and consider a global optimization of a distributed detection system. Reibman and Nolte show that a global optimization can be obtained by solving computationally difficult coupled equations. The performance of four different systems is compared using Receiver Operating Characteristics. Their results show that the globally optimized distributed detection system performs better than the systems developed by Tenney and Sandell, those developed by Chair and Varshney, and almost as well as a *centralized data fusion system* (i.e., complete noise-free observations brought to a central fusion center).

Thomopoulos et al., (1987) consider the problem of optimal fusion in multiple sensor systems using the Neyman-Pearson hypothesis test both at the local sensor level and at the decision fusion level. Analytical results for different system designs are derived, and comparisons between *decentralized systems* (i.e., distributed systems), centralized systems, and systems with quality bits (i.e., small quantities of data describing the quality of the responses) are presented. The decentralized system results in a higher probability of detection than the individual sensors, for cases with three or more sensors and a constant false alarm rate for all the sensors.

Drakopoulos and Lee (1991) consider a distributed detection system where the local sensor responses are dependent, as opposed to earlier studies where independence among sensor responses was a key assumption. To model dependency, they develop a particular correlation model that can be indexed by a single parameter. Note that this correlation model is not appropriate to model continuous responses from local sensors and therefore not suited for purposes of this thesis. Based on this model, they derive and analyze the optimum fusion rule for the dependent case, using the Neyman-Pearson test, and show

how the number of local sensors and the degree of correlation among them affect system performance. The optimum fusion rule is also studied in the two extreme cases, when the degree of correlation is zero and when the degree of correlation is one, and results agree with previous research. Their results show that the fusion rule can be expressed as a function of the given correlation coefficients and that system performance, and the advantages of having multiple local sensors, tend to decrease as correlation increases.

Another group, Kam et al., (1992), also discusses the problem of correlated local sensor responses. Their work is effectively a generalization of the 1986 study by Chair and Varshney, which dealt with fixed binary decisions and statistically independent sensor responses. An optimal fusion rule is derived for the correlated local binary sensor responses in terms of the conditional correlation coefficients. The original fusion rule of Chair and Varshney is shown to be a special case of this analysis, when all the correlation coefficients are zero.

Like Drakopoulos and Lee, Blum (1996) considers distributed detection systems that are optimal under the Neyman-Pearson criterion. Detailed proofs are given for the case with dependent sensor responses, and necessary conditions for optimum fusion at the local sensors and optimum fusion at the fusion center are presented.

Rao and Lyengar (1997) study the case where the probability distributions of the sensors are unknown. They present three methods that can estimate the optimal fusion rule, given a sufficiently large sample. Two of these, Empirical Estimation and the Approximate Decision Rule, are general, but would not yield computationally efficient approximations. The third method, the Nearest-Neighbor Rule, is not general but can provide viable solutions. The latter method is useful when the underlying probabilities are unknown or when the optimal fusion rule is too difficult to implement due to computational complexity.

Jacobson et al., (2001) studied an airport access control system and addressed the research question: *Given a particular access control security system and a pre-specified false clear standard, what is the minimum false alarm rate that can be achieved?* They analyzed two particular access control security systems with independent sensor responses; a 2-sensor system and a 5-sensor system. They introduced two methods for estimating the joint probability density functions; the dynamic, and the static grid estimation procedure.

This thesis complements the work done by Jacobson et al., (2001), by modeling and analyzing the effect of dependency among sensor responses, and the effect that sample size and grid size have on the FAFCT. The next chapter presents the methodology used.

CHAPTER 3

3 Methodology

3.1 Problem Statement

Designers, users, and operators of detection systems are challenged with the false alarm, false clear tradeoff (FAFCT) problem. This thesis studies the FAFCT problem to gain insight into how different system design variables (i.e., sample size and grid size) and model parameters (i.e., correlation coefficient and false clear standard) affect the performance of a 2- sensor detection system.

The research questions addressed in this thesis are: Given a particular detection system and a pre-specified false clear standard,

- What sample sizes (i.e., size of test data set or historical data set used to estimate the distribution of sensor responses) are needed to obtain an appropriate FAFCT?
- How should the grid size (i.e., a measure of how fine (or coarse) the grid structure is) in the static grid estimation procedure be set to assess the distribution of sensor responses?
- How does dependency (measured by a correlation coefficient) among sensors affect the FAFCT?

To answer these questions the following methodology is used. A methodology (i.e., the FAFCT-methodology) is developed to simulate a detection system with two dependent sensors. An experimental design is developed to test the effect sample size, grid size, correlation and FC-standard have on the FAFCT. Computational results are obtained using a computer program originally developed by Simms (1997), which has been complemented with modules for simulating dependency among sensors (i.e., the correlation model developed in this thesis using the conditional distribution method by

Devroye (1986)). To analyze the results, two approaches are taken: creation and analysis of ROC-curves and an analysis of variance.

3.2 FAFCT-methodology

Figure 3-1 depicts the four major steps of the FAFCT-methodology.

1. Collect or generate data points.
2. Estimate probability distributions.
3. Apply knapsack heuristics.
4. Generate ROC-curves.

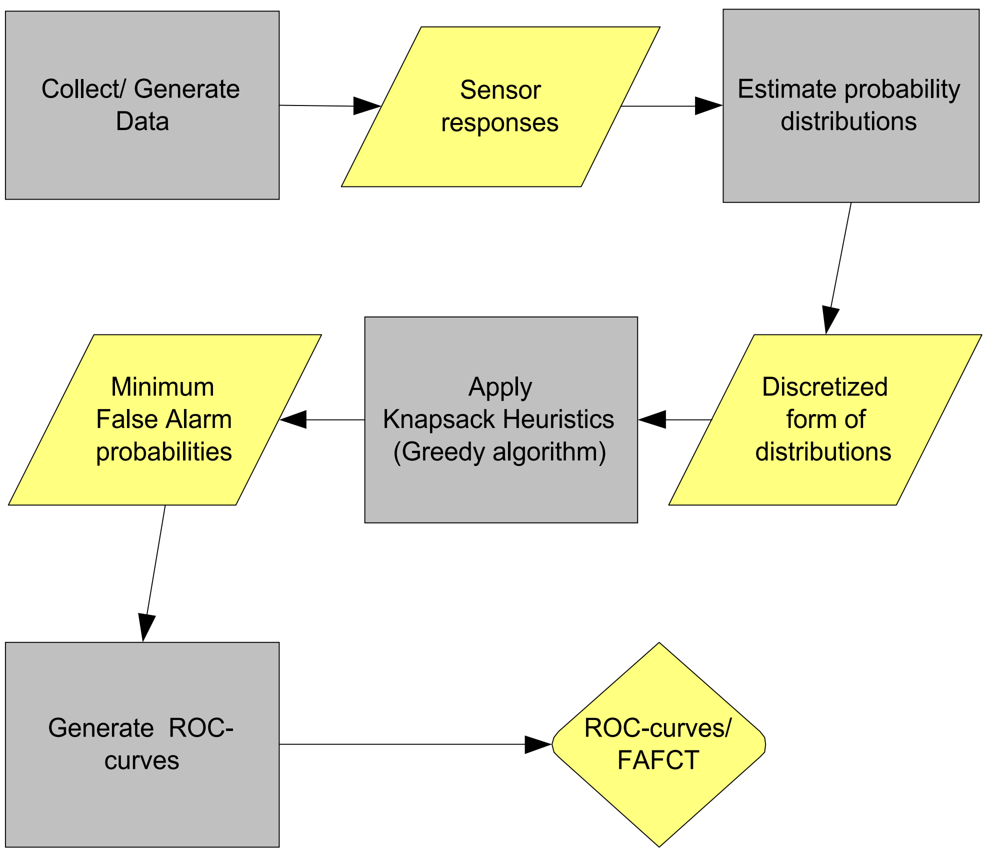

Figure 3-1: ***FAFCT-methodology***

The first step in the FAFCT-methodology is to collect or generate data representing the responses from the local sensors. The actual joint distribution of responses is typically unknown. Historical data or training data (obtained by passing known objects through the system) can be used to estimate the distribution, given that the data points can be scaled to values in the [0, 1] interval. In this thesis, sensor responses are obtained as [0, 1] random variates drawn from known joint probability density functions. These density functions must have certain attributes. First, they should represent the responses from a typical 2-sensor detection system. Second, for modeling purposes, the dependency between the two sensors must be controllable. In Section 3.3 a correlation model is developed to address the problem of designing these joint probability density functions. The conditional distribution method is also introduced (Section 3.4). This method is used to generate random vectors, $\underline{r}$, from the density functions designed with the correlation model. These values will represent the historical or training data used in the second step.

In the second step, the random vectors obtained with the conditional distribution method are used with the static grid estimation procedure to estimate the distribution and to generate a discretized form (i.e., a hypergrid representation) of the sensor responses (see Section 2.2.1). Note that this step is necessary although the probability distributions involved are already known in order to test the effect of grid size and sample size. If historical data or training data were used, the joint probability distribution would not be known. Moreover, the discretized form of the sensor responses is needed as input in the Knapsack heuristics.

Define *sample size* as the number of items observed by the sensors (i.e., the number of random vectors, $\underline{r}$, generated). Also, define *grid size* as the square root of the number of cells in the static grid (i.e., assume that $n_1 = n_2$). For example, a grid size of 10 will yield 100 static grid cells. In a d-sensor detection system the grid size would be defined as the d^{th} root of the number of hypercubes in the static hypergrid assuming that $n_i = n_j$ for $1 \leq i, j \leq d$.

The quality of the estimators obtained in the static grid estimation procedure depends on the sample size and the grid size. Intuitively, a larger sample size will improve the quality of the estimators (i.e., smaller error probabilities), allowing the detection system design algorithm to make a better prediction of the minimum false alarm rate that can be achieved. In many real-world situations it is expensive or impossible to obtain large sample sizes because of time constraints (e.g., if the training data must be obtained when the system is not operating). The problem becomes a search for a minimum sample size that can generate acceptable estimates of the probability density functions. What is acceptable, in this sense, depends on the operating environment and the application of the detection system.

How grid size affects the quality of the estimators is not as intuitive. A large grid size combined with a small sample size might result in very few sample points in each grid cell (or hypercube). On the other hand, if a small grid size is combined with a large sample size, the grid will be too coarse to capture the form of the distribution of sensor responses.

The third step in the FAFCT-methodology involves using Knapsack heuristics to find the optimal FAFCT (i.e., minimum false alarm probability given a particular false clear standard). In particular, the estimators obtained from the static grid estimation procedure are used as the $q_1(i_j)$, and the $q_2(i_j)$ probabilities (Equations (2.5) and (2.6)) in the Knapsack formulation. Solving this mathematical program using Knapsack heuristics (i.e., the greedy algorithm) will return the optimal error probabilities P_F, and P_{FC}.

Finally, ROC curves are obtained by plotting the probability of detection, P_D (i.e., 1-P_{FC}), against the corresponding minimum false alarm probability, P_F. Multiple points are obtained by changing the value of the false clear standard, ε_{FC}. The ROC-curves help us to see tradeoffs. What magnitude of penalty in false alarm probability is associated with a slight decrease in false clear probability (i.e., a stricter false clear standard)?

3.3 Correlation model

The correlation model is written as a linear combination of four marginal density functions. That is, given some marginal density functions, f_1, f_2, g_1, and g_2, find $h(\underline{r}, \lambda)$ where λ represents a parameter that characterizes one of the marginal densities (e.g., f_1). The parameter λ is chosen so that the resulting joint probability density function, $h(\underline{r}, \lambda)$, describes random vectors, $\underline{r}$, with elements that are independent, positively correlated or negatively correlated to some degree. The marginal densities must be chosen so that the compounded joint probability density function, $h(\underline{r}, \lambda)$, properly represents a typical 2-sensor detection system (i.e., sensor responses close to zero should represent a target is not present and sensor responses close to one should represent a target is present). This is a design process that involves some trial and error. Graphing the resulting joint probability density function helps.

Consider a two-device system with marginal density functions f_1, f_2, g_1, and g_2 (e.g., truncated exponential densities), and a joint probability density function expressed as:

$$h(r_1, r_2, \lambda) = \alpha\ f_1(r_1) f_2(r_2) + (1-\alpha)\ g_1(r_1) g_2(r_2). \tag{3.1}$$

Note, $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(r_1, \mathrm{r}_2, \lambda)\, dr_1 dr_2 = 1$, for $0 \le \alpha \le 1$.

To see that the covariance and the correlation coefficient can be written as functions of λ for any set of marginal density functions, $\{f_1, f_2, g_1, g_2\}$, the covariance must first be considered. By definition,

$$Cov(R_1, R_2) = E[R_1 R_2] - E[R_1]E[R_2].$$

The expectations can be obtained as follows. Define,

$$E_f[R_1] = \int_{-\infty}^{\infty} r_1 f_1 dr_1$$

$$E_g[R_1] = \int_{-\infty}^{\infty} r_1 g_1 dr_1$$

$$E_f[R_2] = \int_{-\infty}^{\infty} r_2 f_2 dr_2$$

$$E_g[R_2] = \int_{-\infty}^{\infty} r_2 g_2 dr_2.$$

Then,

$$E[R_1 R_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} r_1 r_2 h_{R_1 R_2}(r_1, r_2)\, dr_1 dr_2$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \alpha\, r_1 r_2 f_1(r_1) f_2(r_2)\, dr_1 dr_2 + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (1-\alpha) r_1 r_2 g_1(r_1) g_2(r_2)\, dr_1 dr_2$$

$$= \alpha \int_{-\infty}^{\infty} r_1 f_1(r_1)\, dr_1 \int_{-\infty}^{\infty} r_2 f_2(r_2)\, dr_2 + (1-\alpha) \int_{-\infty}^{\infty} r_1 g_1(r_1)\, dr_1 \int_{-\infty}^{\infty} r_2 g_2(r_2)\, dr_2$$

$$= \alpha\, E_f[R_1] E_f[R_2] + (1-\alpha) E_g[R_1] E_g[R_2]. \tag{3.2}$$

Also,

$$E[R_1] = \int_{-\infty}^{\infty} r_1 h_{R_1}(r_1)\, dr_1 \text{ and}$$

$$E[R_2] = \int_{-\infty}^{\infty} r_2 h_{R_2}(r_2)\, dr_2 \text{ , where}$$

$$h_{R_1}(r_1) = \int_{-\infty}^{\infty} h_{R_1 R_2}(r_1, r_2)\, dr_2$$

$$= \alpha \int_{-\infty}^{\infty} f_1(r_1) f_2(r_2)\, dr_2 + (1-\alpha) \int_{-\infty}^{\infty} g_1(r_1) g_2(r_2)\, dr_2$$

$$= \alpha\, f_1(r_1) + (1-\alpha) g_1(r_1) \text{, and similarly}$$

$$h_{R_2}(r_2) = \alpha\, f_2(r_2) + (1-\alpha) g_2(r_2).$$

Hence,

$$E[R_1] = \int_{-\infty}^{\infty} r_1 h_{R_1}(r_1)\, dr_1$$

$$= \int_{-\infty}^{\infty} r_1(\alpha\, f_1(r_1) + (1-\alpha) g_1(r_1))\, dr_1$$

$$= \alpha \int_{-\infty}^{\infty} r_1 f_1(r_1)\, dr_1 + (1-\alpha) \int_{-\infty}^{\infty} r_1 g_1(r_1)\, dr_1$$

$$= \alpha\, E_f[R_1] + (1-\alpha) E_g[R_1] \text{, and} \tag{3.3}$$

$$E[R_2] = \alpha\, E_f[R_2] + (1-\alpha) E_g[R_2]. \tag{3.4}$$

The covariance can be expressed using (3.2), (3.3) and (3.4).

$$Cov(R_1, R_2) = E[R_1 R_2] - E[R_1] E[R_2]$$

$$
\begin{aligned}
&= \left(\alpha E_f[R_1]E_f[R_2] + (1-\alpha)E_g[R_1]E_g[R_2]\right) \\
&\quad - \left(\alpha E_f[R_1] + (1-\alpha)E_g[R_1]\right)\left(\alpha E_f[R_2] + (1-\alpha)E_g[R_2]\right) \\
&= \alpha E_f[R_1]E_f[R_2] + (1-\alpha)E_g[R_1]E_g[R_2] \\
&\quad - \alpha^2 E_f[R_1]E_f[R_2] - \alpha(1-\alpha)E_f[R_1]E_g[R_2] \\
&\quad - \alpha(1-\alpha)E_g[R_1]E_f[R_2] - (1-\alpha)^2 E_g[R_1]E_g[R_2] \\
&= \alpha(1-\alpha)E_f[R_1]E_f[R_2] + \alpha(1-\alpha)E_g[R_1]E_g[R_2] \\
&\quad - \alpha(1-\alpha)E_f[R_1]E_g[R_2] - \alpha(1-\alpha)E_g[R_1]E_f[R_2] \\
&= \alpha(1-\alpha)\left(E_f[R_1] - E_g[R_1]\right)\left(E_f[R_2] - E_g[R_2]\right).
\end{aligned} \tag{3.5}
$$

Furthermore, the correlation coefficient can be written as

$$
\rho = \frac{Cov(R_1, R_2)}{\sqrt{Var(R_1)Var(R_2)}}. \tag{3.6}
$$

By definition,

$$
\begin{aligned}
&Var(R_1) = E[R_1^{\,2}] - E[R_1]^2, \text{ and} \\
&E[R_1^{\,2}] = \int_{-\infty}^{\infty} r_1^{\,2} h_1(r_1)\, dr_1 = \int_{-\infty}^{\infty} r_1^{\,2} (\alpha f_1(r_1) + (1-\alpha) g_1(r_1))\, dr_1 \\
&\qquad = \alpha \int_{-\infty}^{\infty} r_1^{\,2} f_1(r_1)\, dr_1 + (1-\alpha) \int_{-\infty}^{\infty} r_1^{\,2} g_1(r_1)\, dr_1 \\
&\qquad = \alpha E_f[R_1^{\,2}] + (1-\alpha)E_g[R_1^{\,2}].
\end{aligned} \tag{3.7}
$$

The variance can be expressed using (3.3) and (3.7).

$$
\begin{aligned}
Var(R_1) &= \alpha E_f[R_1^{\,2}] + (1-\alpha)E_g[R_1^{\,2}] - (\alpha E_f[R_1] + (1-\alpha)E_g[R_1])^2 \\
&= \alpha E_f[R_1^{\,2}] + (1-\alpha)E_g[R_1^{\,2}] - \alpha^2 E_f[R_1]^2 - 2\alpha(1-\alpha)E_f[R_1]E_g[R_1] \\
&\quad - (1-\alpha)^2 E_g[R_1]^2 \\
&= \alpha E_f[R_1^{\,2}] - \alpha^2 E_f[R_1^{\,2}] + \alpha^2 E_f[R_1^{\,2}] - \alpha^2 E_f[R_1]^2 \\
&\quad + (1-\alpha)E_g[R_1^{\,2}] - (1-\alpha)^2 E_g[R_1^{\,2}] + (1-\alpha)^2 E_g[R_1^{\,2}] \\
&\quad - (1-\alpha)^2 E_g[R_1]^2 - 2\alpha(1-\alpha)E_f[R_1]E_g[R_1] \\
&= \alpha^2 Var_f(R_1) + (1-\alpha)^2 Var_g(R_1) + \alpha(1-\alpha)E_f[R_1^{\,2}] \\
&\quad + \alpha(1-\alpha)E_g[R_1^{\,2}] - 2\alpha(1-\alpha)E_f[R_1]E_g[R_1]
\end{aligned}
$$

$$
\begin{aligned}
&= \alpha^2 Var_f(R_1) + (1-\alpha)^2 Var_g(R_1) \\
&\quad + \alpha(1-\alpha)(E_f[R_1^2] + E_g[R_1^2] - 2E_f[R_1]E_g[R_1]) \\
&= \alpha^2 Var_f(R_1) + (1-\alpha)^2 Var_g(R_1) \\
&\quad + \alpha(1-\alpha)(E_f[R_1^2] - E_f[R_1]^2 + E_f[R_1]^2 \\
&\quad + E_g[R_1^2] - E_g[R_1]^2 + E_g[R_1]^2 - 2E_f[R_1]E_g[R_1]) \\
&= \alpha^2 Var_f(R_1) + (1-\alpha)^2 Var_g(R_1) \\
&\quad + \alpha(1-\alpha)(Var_f(R_1) + Var_g(R_1)) \\
&\quad + \alpha(1-\alpha)(E_f[R_1]^2 - 2E_f[R_1]E_g[R_1] + E_g[R_1]^2) \\
&= \alpha\, Var_f(R_1) + (1-\alpha)Var_g(R_1) + \alpha(1-\alpha)(E_f[R_1] - E_g[R_1])^2 . \qquad (3.8)
\end{aligned}
$$

Similarly,

$$
Var(R_2) = \alpha\, Var_f(R_2) + (1-\alpha)Var_g(R_2) + \alpha(1-\alpha)(E_f[R_2] - E_g[R_2])^2 . \qquad (3.9)
$$

The correlation coefficient defined as (3.6) can now be expressed using (3.5), (3.8) and (3.9) as,

$$
\rho = \frac{\alpha(1-\alpha)\big(E_f[R_1] - E_g[R_1]\big)\big(E_f[R_2] - E_g[R_2]\big)}{\sqrt{\begin{pmatrix} \alpha\, Var_f(R_1) + (1-\alpha)Var_g(R_1) \\ + \alpha(1-\alpha)(E_f[R_1] - E_g[R_1])^2 \end{pmatrix} \begin{pmatrix} \alpha\, Var_f(R_2) + (1-\alpha)Var_g(R_2) \\ + \alpha(1-\alpha)(E_f[R_2] - E_g[R_2])^2 \end{pmatrix}}} \qquad (3.10)
$$

Joint probability density functions are designed by choosing an appropriate set of marginal density functions. Two sets of marginal density functions must be selected, one set to design $f_{\underline{R}|NT}(\underline{r})$ and another set to design $f_{\underline{R}|T}(\underline{r})$.

3.3.1 Example of how to apply the correlation model

This example illustrates how the joint probability density function $f_{\underline{R}|T}(\underline{r})$ can be designed using the correlation model described in the previous section. A similar approach can be used to design $f_{\underline{R}|NT}(\underline{r})$ with an appropriately chosen set of marginal density functions.

For this example truncated exponential functions will be used for the marginal density functions f_1, f_2, g_1, and g_2 with $\alpha = 0.5$. The parameters of three of the marginal density functions (e.g., f_2, g_1, and g_2) have to be fixed and the remaining density function (i.e., f_1) can be characterized with the parameter λ. Then

$$f_1 = \frac{\lambda e^{-\lambda r_1}}{1-e^{-\lambda}}\ , \qquad g_1 = \frac{\lambda_3 e^{-\lambda_3 r_1}}{1-e^{-\lambda_3}}, \qquad f_2 = \frac{\lambda_2 e^{-\lambda_2 r_2}}{1-e^{-\lambda_2}},$$

and $g_2 = \frac{\lambda_4 e^{-\lambda_4 r_2}}{1-e^{-\lambda_4}}$, where λ_2, λ_3, and λ_4 are known constants (to be defined later), and $0 \le r_1 \le 1, 0 \le r_2 \le 1$.

In order to express (3.10) in terms of λ, by definition,

$$E_f[R_1] = \int_0^1 r_1 f_1 \, dr_1 = \frac{1-e^{\lambda}+\lambda}{\lambda - \lambda e^{\lambda}}, \tag{3.11}$$

$$E_f[R_1{}^2] = \int_0^1 r_1{}^2 f_1 \, dr_1 = -\frac{2-2e^{\lambda}+2\lambda+\lambda^2}{(e^{\lambda}-1)\lambda^2}, \text{ and} \tag{3.12}$$

$$Var_f(R_1) = E_f[R_1{}^2] - E_f[R_1]^2 = \frac{1+e^{2\lambda}-e^{\lambda}(2+\lambda^2)}{\lambda^2(e^{\lambda}-1)^2}. \tag{3.13}$$

The corresponding expectations and variances for the remaining marginal density functions are known constants defined by their respective λ-values. In particular,

$$E_f[R_2] = \frac{1-e^{\lambda_2}+\lambda_2}{\lambda_2 - \lambda_2 e^{\lambda_2}} = C_2,$$

$$Var_f(R_2) = \frac{1+e^{2\lambda_2}-e^{\lambda_2}(2+\lambda_2{}^2)}{\lambda_2{}^2(e^{\lambda_2}-1)^2} = K_2,$$

$$E_g[R_1] = \frac{1-e^{\lambda_3}+\lambda_3}{\lambda_3 - \lambda_3 e^{\lambda_3}} = C_3,$$

$$Var_g(R_1) = \frac{1+e^{2\lambda_3}-e^{\lambda_3}(2+\lambda_3{}^2)}{\lambda_3{}^2(e^{\lambda_3}-1)^2} = K_3,$$

$$E_g[R_2] = \frac{1-e^{\lambda_4}+\lambda_4}{\lambda_4 - \lambda_4 e^{\lambda_4}} = C_4, \text{ and}$$

$$Var_g(R_2) = \frac{1+e^{2\lambda_4}-e^{\lambda_4}(2+\lambda_4{}^2)}{\lambda_4{}^2(e^{\lambda_4}-1)^2} = K_4.$$

Through substitution of (3.11), (3.12), (3.13), and the known expectations and variances into (3.10), an expression of the correlation coefficient as a function of λ can be obtained.

$$\rho = \frac{\frac{1}{4}\left(\frac{1-e^{\lambda}+\lambda}{\lambda-\lambda e^{\lambda}} - C_3\right)\left(C_2 - C_4\right)}{\sqrt{\begin{pmatrix} \frac{1}{2}\left(\frac{1+e^{2\lambda}-e^{\lambda}(2+\lambda^2)}{\lambda^2(e^{\lambda}-1)^2}\right)+\frac{1}{2}K_3 \\ +\frac{1}{4}\left(\frac{1-e^{\lambda}+\lambda}{\lambda-\lambda e^{\lambda}} - C_3\right)^2 \end{pmatrix}\begin{pmatrix} \frac{1}{2}K_2+\frac{1}{2}K_4 \\ +\frac{1}{4}(C_2-C_4)^2 \end{pmatrix}}}. \tag{3.14}$$

To design a density functions with known correlation, choose a correlation coefficient and numerically solve for λ.

3.3.2 The joint density functions designed for use in this thesis

A total of six joint density functions were designed (i.e., $f_{\underline{R}|NT}(\underline{r})$ and $f_{\underline{R}|T}(\underline{r})$ for three different levels of correlation, ρ = -0.4, ρ = 0, and ρ = 0.4). To design $f_{\underline{R}|T}(\underline{r})$, λ_2, λ_3, and λ_4 must be selected so that the random vector, $\underline{r}$, appropriately simulates the sensor responses of a 2-device detection system when a target is present (i.e., high density of responses close to one). To design $f_{\underline{R}|NT}(\underline{r})$, λ_2, λ_3, and λ_4 must be selected so that a high density of responses close to zero is obtained.

Recall Equation 3.1. The linear combination coefficient α is chosen as 0.5. This simply means that the product of the marginal density functions f_1 and f_2 is weighted equal to the product of marginal density functions g_1 and g_2. Furthermore, recall that two sets of marginal density functions, $\Omega = \{f_1, f_2, g_1, g_2\}$, must be selected. Let Ω_T denote the set used to design joint density functions conditional on a target being present, and let Ω_{NT} denote the set used to design joint density functions conditional on a target not being present. Choose $\Omega_T = \{f_{1_T}, f_{2_T}, g_{1_T}, g_{2_T}\}$, where

$$f_{1_T} = \frac{\lambda e^{-\lambda(1-r_1)}}{1-e^{-\lambda}},\ f_{2_T} = \frac{0.1e^{-0.1(1-r_2)}}{1-e^{-0.1}},\ g_{1_T} = \frac{5e^{-5(1-r_1)}}{1-e^{-5}},\text{ and } g_{2_T} = \frac{50e^{-50(1-r_2)}}{1-e^{-50}}.$$

Furthermore, choose $\Omega_{NT} = \{f_{1_{NT}}, f_{2_{NT}}, g_{1_{NT}}, g_{2_{NT}}\}$, where

$$f_{1_{NT}} = \frac{\lambda e^{-\lambda r_1}}{1-e^{-\lambda}}, \; f_{2_{NT}} = \frac{0.1e^{-0.1r_2}}{1-e^{-0.1}}, \; g_{1_{NT}} = \frac{5e^{-5r_1}}{1-e^{-5}}, \text{ and } g_{2_{NT}} = \frac{50e^{-50r_2}}{1-e^{-50}}.$$

Now, solve equation 3.14 numerically to obtain λ values that define the joint probability density functions (PDF) with known correlation coefficients ρ = -0.4, ρ = 0, and ρ = 0.4.

ρ = -0.4 gives λ = 33.0667, and

$$f_{\underline{R}|NT}(\underline{r}) = 0.5\left(\frac{33.0667e^{-33.0667r_1}}{1-e^{-33.0667}}\right)\left(\frac{0.1e^{-0.1r_2}}{1-e^{-0.1}}\right) + (1-0.5)\left(\frac{5e^{-5r_1}}{1-e^{-5}}\right)\left(\frac{50e^{-50r_2}}{1-e^{-50}}\right)$$
$$= 17.3738e^{-33.0667r_1-0.1r_2} + 125.848e^{-5r_1-50r_2}$$

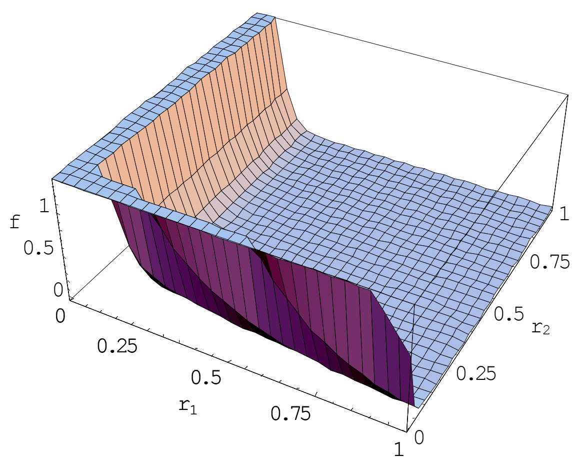

Figure 3-2: ***Joint Probability Density Function (PDF) for a system response conditional on a target not being present and*** ρ ***= -0.4.***

$$f_{\underline{R}|T}(\underline{r}) = 0.5\left(\frac{33.0667e^{-33.0667(1-r_1)}}{1-e^{-33.0667}}\right)\left(\frac{0.1e^{-0.1(1-r_2)}}{1-e^{-0.1}}\right) + (1-0.5)\left(\frac{5e^{-5(1-r_1)}}{1-e^{-5}}\right)\left(\frac{50e^{-50(1-r_2)}}{1-e^{-50}}\right)$$
$$= 6.85141\times10^{-14}e^{33.0667r_1+0.1r_2} + 1.6355\times10^{-22}e^{5r_1+50r_2}$$

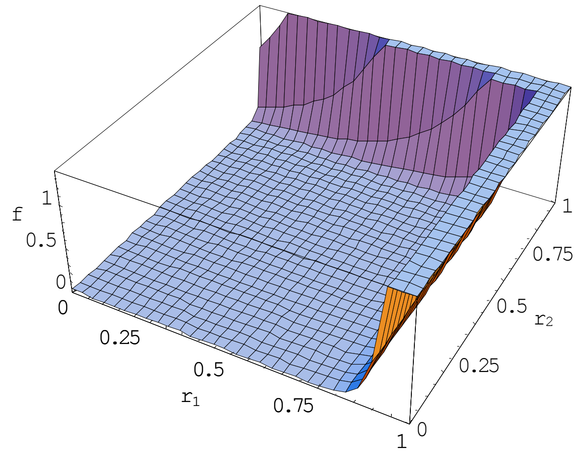

Figure 3-3: ***Joint PDF for a system response conditional on a target being present and*** **ρ = *-0.4.***

ρ = 0 gives λ = 5.0, and

$$f_{\underline{R}|NT}(\underline{r}) = 0.5\left(\frac{5e^{-5r_1}}{1-e^{-5}}\right)\left(\frac{0.1e^{-0.1r_2}}{1-e^{-0.1}}\right) + (1-0.5)\left(\frac{5e^{-5r_1}}{1-e^{-5}}\right)\left(\frac{50e^{-50r_2}}{1-e^{-50}}\right)$$
$$= 2.6449e^{-5r_1-0.1r_2} + 125.848e^{-5r_1-50r_2}$$

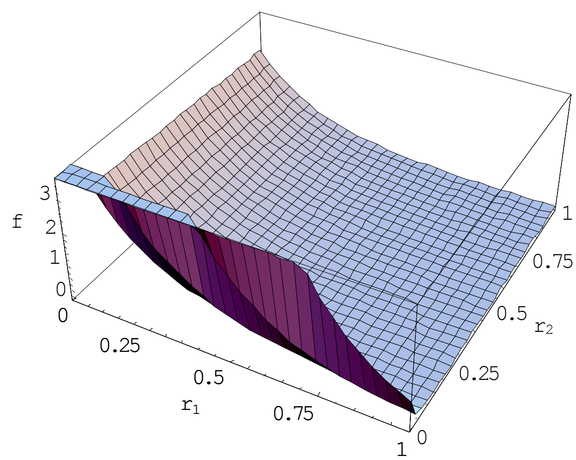

Figure 3-4: ***Joint PDF for a system response conditional on a target not being present and*** **ρ = *0.***

$$f_{\underline{R}|T}(\underline{r}) = 0.5\left(\frac{5e^{-5(1-r_1)}}{1-e^{-5}}\right)\left(\frac{0.1e^{-0.1(1-r_2)}}{1-e^{-0.1}}\right) + (1-0.5)\left(\frac{5e^{-5(1-r_1)}}{1-e^{-5}}\right)\left(\frac{50e^{-50(1-r_2)}}{1-e^{-50}}\right)$$
$$= 0.0161253e^{5r_1+0.1r_2} + 1.6355\times 10^{-22}\, e^{5r_1+50r_2}$$

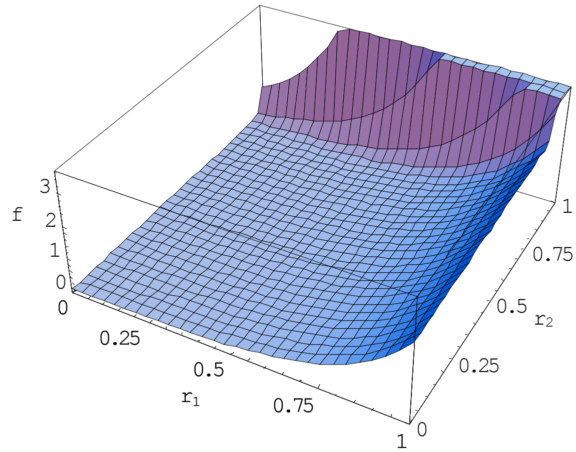

Figure 3-5: ***Joint PDF for a system response conditional on a target being present and*** ρ ***= 0.***

$\rho = 0.4$ gives $\lambda = 0.066015$, and

$$f_{\underline{R}|NT}(\underline{r}) = 0.5\left(\frac{0.066015e^{-0.066015r_1}}{1-e^{-0.066015}}\right)\left(\frac{0.1e^{-0.1r_2}}{1-e^{-0.1}}\right) + (1-0.5)\left(\frac{5e^{-5r_1}}{1-e^{-5}}\right)\left(\frac{50e^{-50r_2}}{1-e^{-50}}\right)$$
$$= 0.54295e^{-0.066015r_1-0.1r_2} + 125.848e^{-5r_1-50r_2}$$

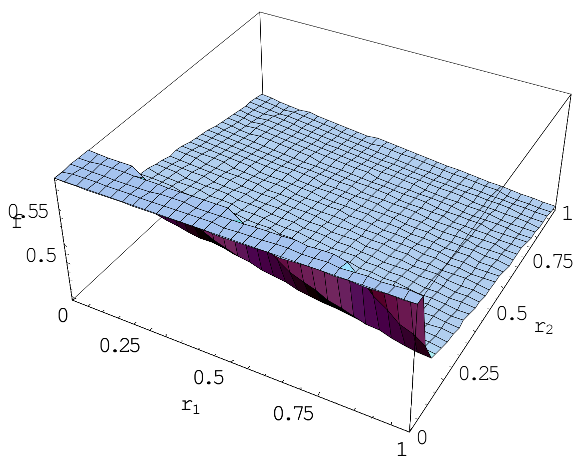

Figure 3-6: ***Joint PDF for a system response conditional on a target not being present and*** ρ ***= 0.4.***

$$f_{\underline{R}|T}(\underline{r}) = 0.5\left(\frac{0.066015e^{-0.066015(1-r_1)}}{1-e^{-0.066015}}\right)\left(\frac{0.1e^{-0.1(1-r_2)}}{1-e^{-0.1}}\right) + (1-0.5)\left(\frac{5e^{-5(1-r_1)}}{1-e^{-5}}\right)\left(\frac{50e^{-50(1-r_2)}}{1-e^{-50}}\right)$$
$$= 0.459897e^{0.066015r_1+0.1r_2} + 1.6355\times10^{-22}e^{5r_1+50r_2}$$

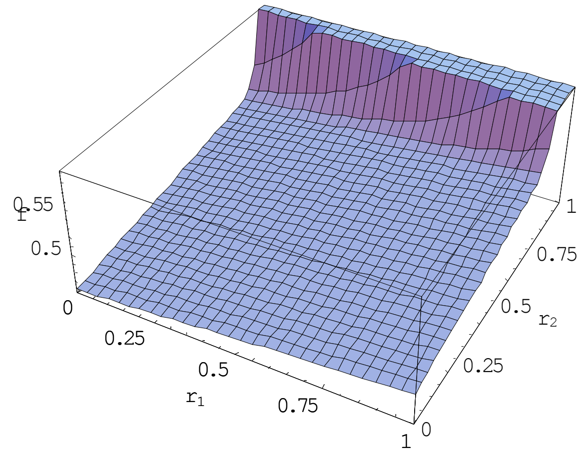

Figure 3-7: ***Joint PDF for a system response conditional on a target being present and*** ρ ***= 0.4.***

3.4 The Conditional Distribution Method

This method is used to generate random vectors, $\underline{r}$, from the density functions designed with the correlation model. The method is called the Conditional Distribution Method and can be found in (Devroye, 1986, Ch. 11, pg. 555).

Suppose a joint distribution function $F_{R_1,R_2,\ldots,R_n}(r_1,r_2,\ldots,r_n)$ (obtained by integrating $f_{\underline{R}|NT}(\underline{r})$ or $f_{\underline{R}|T}(\underline{r})$) is used to generate a random vector $\underline{r}=(r_1,r_2,\ldots,r_n)$. Also assume that for $k=2,3,\ldots,n$, the conditional distribution of R_k given $R_i=r_i$ for $i=1,2,\ldots,k-1$, denoted by $F_{R_k}(r_k|r_1,r_2,\ldots,r_{k-1})$, can be obtained. In addition, let $F_{R_i}(r_i)$ be the marginal distribution function of R_i for $i=1,2,\ldots,n$. Then the following general algorithm can be used to generate a random vector, $\underline{r}$, with joint distribution function $F_{R_1,R_2,\ldots,R_n}(r_1,r_2,\ldots,r_n)$.

Generate r_1 with distribution function $F_{R_1}(r_1)$.

Set i = 2.

Do While i ≤ n:

    Generate r_i with distribution function $F_{R_i}(\cdot|r_1,r_2,\ldots,r_{i-1})$.

    Set i = i + 1.

Return $\underline{r}=(r_1,r_2,\ldots,r_n)$.

To generate $\underline{r}$, the method of inversion is used. Given a distribution function $F_{R_i}\left(\cdot \middle| r_1, r_2, \ldots, r_{i-1}\right)$ and the values of $r_1, r_2, \ldots, r_{i-1}$, the value of r_i is obtained as $r_i = F_{R_i}^{-1}\left(U \middle| r_1, r_2, \ldots, r_{i-1}\right)$, where U is a uniform pseudo-random number. Since it is very unlikely to find a closed-form expression for the inverse function $F_{R_i}^{-1}$ a numerical search method is applied to find r_i.

3.4.1 A simple 2-sensor example

Using the example where ρ = 0.4 and λ = 0.066015 we can obtain the marginal distribution function $F_{R_1|T}(r_1)$ and the conditional distribution function $F_{R_2|T}\left(r_2 \middle| r_1\right)$.

$$F_{R_1|T}\left(r_1\right) = \int_0^{r_1}\int_0^1 f_{\underline{R}|T}(\underline{r})\partial r_2 \partial u = \int_0^{r_1}\int_0^1 0.54295e^{-0.066015u-0.1r_2} + 125.848e^{-5u-50r_2}\partial r_2 \partial u$$
$$= \int_0^{r_1} 2.51696e^{-5u} + 0.516685e^{-0.066015u}\partial u = 8.33018 - 0.503392e^{-5r_1} - 7.82679e^{-0.066015r_1}$$

, and

$$F_{R_2|T}\left(r_2 \middle| r_1\right) = \int_0^{r_2} \frac{f_{\underline{R}|T}(\underline{r})}{f_{R_1}(r_1)}\partial v = \int_0^{r_2} \frac{0.54295e^{-0.066015r_1-0.1v} + 125.848e^{-5r_1-50v}}{2.51696e^{-5r_1} + 0.516685e^{-0.066015r_1}}\partial v$$
$$= \frac{e^{0.066015r_1-50r_2} + 2.15717e^{5r_1+0.1r_2} - e^{0.066015r_1} - 2.15717e^{5r_1}}{-e^{0.066015r_1} - 0.205282e^{5r_1}}.$$

Random vectors, $\underline{r}$, can now be obtained by numerically solving the equations $F_{R_1|T}{}^{-1}\left(r_1\right) = U$ and $F_{R_2|T}{}^{-1}\left(r_2 \middle| r_1\right) = U$, where the values of U are uniform random numbers in the interval [0, 1].

3.5 Experimental Design

The experiments are based on a 3^4 - factorial design. Computational results from the experimental design are used to perform an analysis of variance with the help of interaction plots and ANOVA tables (Montgomery, 1991, Ch.12, pp.387). Four factors (i.e., Sample size, Grid size, Correlation, and FC-standard), each at three levels (i.e., low, intermediate, and high) results in eighty-one treatment combinations (see Table 3-1).

Fifty replications of each treatment combination are run. Although only three values of the false clear standard are required in the 3^4 - factorial design, twelve values in the range ($0 \leq \varepsilon_{FC} \leq 1$) are used to obtain more points to plot the ROC - curves. The following setup was used.

	Level		
Factor	Low (0)	Intermediate (1)	High (2)
Sample size (A)	1,000	10,000	100,000
Grid size (B)	5 (25 subregions)	10 (100 subregions)	15 (225 subregions)
Correlation (C)	-0.4	0	+0.4
FC-standard (D)	0.001	0.01	0.025

Table 3-1: ***Experimental design (Factors and their respective levels)***

Each combination of sample size, grid size and correlation yields one ROC-curve that represents the optimal FAFCT for that particular system setting. The set of ROC-curves are superimposed and the resulting graphs can be used to make conclusions and suggest guidelines about design issues that often face designers, operators, and users of detection systems.

A 3-level design was chosen to account for positive and negative correlated sensors as well as independent sensors. The correlation coefficients were chosen by trial and error with the intention to obtain realistic joint density functions. More positive or negative correlation coefficient would yield joint density functions with very extreme (i.e., unrealistic) shapes and would also decrease solvability (remember that random variates must be obtained through numerical inversion, as described in Section 3.4). A negative correlation coefficient means that the two different sensors tend to disagree (i.e., an indication of a "target is present" by one of the sensors increases the probability that the second sensor will indicate a "target is not present"). Positive correlation, on the other hand, means that the sensors tend to agree (i.e., an indication of a "target is present" by one of the sensors increases the probability that the second sensor will also indicate a "target is present").

The levels for sample size are chosen to obtain a significant dispersion and at the same time keep the values within a realistic range. A very large sample size is both time-consuming and costly to obtain in reality, and a very small sample size would be of little use since it would not give sufficient knowledge of the system. Grid sizes are chosen with the sample sizes in mind. For example, a large grid size (e.g., 225 subregions) in conjunction with a small sample size will imply that few sample points are observed in each sub-region, resulting in a poor estimation of the joint probability density functions. The FC-standard is chosen to represent different detection system objectives. Depending on the severity (i.e., economic and political implications) of a potential false clear, the FC-standard is set to more or less strict levels.

CHAPTER 4

4 Results

The computational results obtained from the experimental design are analyzed in two ways. First, an analysis of variance is performed to check the null hypothesis, H_0: There is no significant difference between the sample means of the different treatment combinations. The formal statement of the null, and the alternate hypothesis is:

$H_0 : \mu_1 = \mu_2 = \ldots = \mu_{81}$, and

$H_1 : \mu_i \neq \mu_j$, for at least one pair (i, j).

Secondly, the ROC-curves are presented and analyzed.

4.1 Analysis of Variance

The purpose of the analysis of variance is to find out if changes in the levels of the factors (i.e., sample size, grid size, correlation, and FC-standard) affect the false alarm probabilities (i.e., to test the null hypothesis).

Table 4-1 is a representation of the eighty-one treatment combinations. Each value represents the average false alarm probability out of fifty independent replications. By inspection, it is clear that some differences between the treatment means are present. For example, by comparing the upper three rows (i.e., correlation on the low level) with the lower three rows (i.e., correlation on the high level), a distinct difference is noticeable. The false alarm probabilities in the upper three rows range from 0.001 to 0.104 while the corresponding values in the lower three rows range from 0.489 to 1.000. Similar differences can be found by closer inspection. False alarm probabilities for each of the fifty independent replications and eighty-one treatment combinations appear in Appendix B.

		Sample size (A)								
		0			1			2		
		Grid size (B)								
Correlation (C)	FC-standard (D)	0	1	2	0	1	2	0	1	2
0	0	0.099	0.036	0.025	0.104	0.055	0.036	0.102	0.054	0.035
0	1	0.012	0.009	0.013	0.006	0.003	0.020	0.006	0.002	0.003
0	2	0.006	0.006	0.007	0.006	0.003	0.002	0.006	0.002	0.001
1	0	0.654	0.335	0.217	0.835	0.567	0.471	0.950	0.608	0.559
1	1	0.269	0.162	0.131	0.278	0.211	0.185	0.283	0.216	0.198
1	2	0.149	0.091	0.080	0.158	0.110	0.097	0.153	0.111	0.102
2	0	1.000	0.973	0.839	1.000	1.000	0.997	1.000	1.000	1.000
2	1	0.993	0.691	0.601	1.000	0.691	0.619	1.000	0.675	0.626
2	2	0.659	0.508	0.489	0.661	0.518	0.496	0.660	0.514	0.498

Table 4-1: ***Average false alarm probabilities for the 81 treatment combinations.***

A residual analysis detected a number of possible outliers and one can argue that these data points should have been removed. However, they were kept for several reasons. They did not have much of an impact on the mean false alarm rates; they could not be explained without further experiments; and removing them would have made the model unbalanced. Data points were deemed outliers if they diverged more than three standard deviations from the mean. Outliers are indicated in Appendix B.

It is clear by studying the average false alarm probabilities that correlation (i.e., dependency among the sensor responses) strongly influences the performance of the detection system. The Main Effects Plot, Figure 4-1, further demonstrates the magnitude of the influence of correlation in comparison with the other factors.

Figure 4-1 consists of four plots. Each plot depicts the average false alarm probabilities for each level of the factors sample size, grid size, correlation, and FC-standard. For example: the point representing sample size at level 0 (i.e., sample size equal to 1000) is computed as the mean of the first three columns in Table 4-1.

Consistent with Table 4-1, Figure 4-1 shows that correlation is the most influential factor. One can expect that positively correlated sensor responses will not perform desirably (i.e., relatively large false alarm probabilities). Moreover, Figure 4-1 shows that a finer grid (i.e., larger grid size) seem to provide better estimates of the probability densities involved, and thus reduces the false alarm probabilities. Changes in grid size from the intermediate to the high level seem insignificant. The differences, in false alarm probabilities, between sample sizes of 1000, 10000, and 100000 also seem insignificant and indicates that sample size has no effect on the false alarm probabilities. Figure 4-1 also shows that the false clear standard and the false alarm rate are inversely related, as should be expected. In order to secure a high probability of detection (i.e., small false clear standard) a higher probability of false alarm must be accepted.

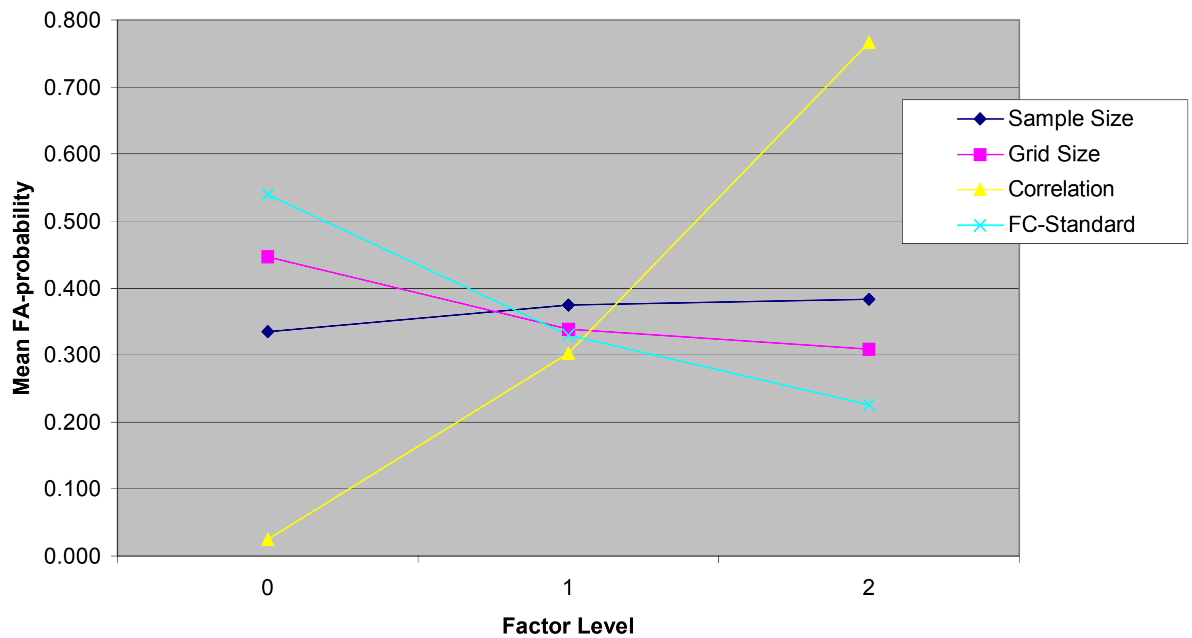

Figure 4-1: ***Main Effects Plot - Data Means for FA-probabilities***

As shown, the influence of correlation is so strong in comparison with the other factors that it is wise to isolate the correlation as a factor and treat the analysis of the remaining factors as three separate scenarios (i.e., for negatively correlated sensor responses, uncorrelated sensor responses, and positively correlated sensor responses). Each scenario is treated as an independent 3^3 - factorial design where the effect of sample size, grid size, and FC-standard on the false alarm probabilities is studied. This approach makes sense from a practical standpoint also since one can argue that correlation is a dependent design parameter and hence controllable to some extent (i.e., if the detection system is not already in place one can make a decision whether to employ correlated or uncorrelated sensors).

Furthermore, it was hard to make any definitive statements from analyzing the ANOVA tables since the mean square error (MSE) was so small. Due to this complication interaction plots will be the main focus for the remaining part of the analysis. The initial analysis, with correlation treated as one of four factors in a 3^4 - factorial design and generated ANOVA tables; appear in Appendix E, in its entirety.

4.1.1 Negatively Correlated Sensor Responses

Since the effect of sample size and grid size showed to be very small compared to correlation the decision was made to isolate the correlation and look at sample size, grid size, and FC-standard for each level of correlation.

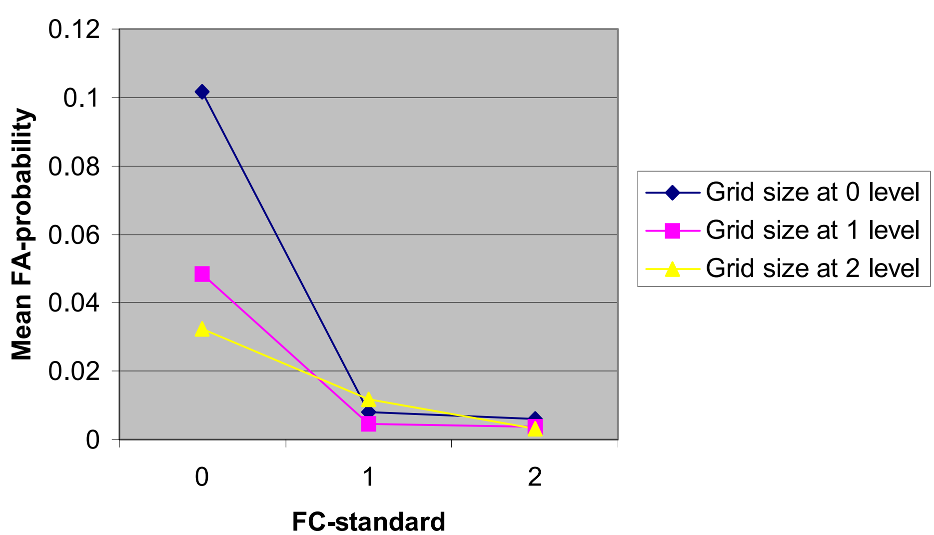

Figure 4-2: ***Grid size – FC-standard interaction for negatively correlated sensors***

Figure 4-2 shows the interaction between grid size and FC-standard for negatively correlated sensor responses. It clearly shows that for strict FC-standards, grid size does matter. Increasing the grid size drastically decreases the false alarm probabilities (i.e., improves the performance of the detection system).

Figure 4-2 also shows that grid size seems insignificant for FC-standards at the intermediate (i.e., 0.01) and high (i.e., 0.025) levels. To verify whether grid size is in fact insignificant for larger FC-standards one could create confidence intervals around each plot in Figure 4-2. If the confidence intervals overlap one can determine that grid size is in fact insignificant. This study is suggested as future research and will not be covered in this thesis.

Figure 4-3 shows the interaction between sample size and FC-standard for negatively correlated sensor responses. Comparing Figure 4-2 and Figure 4-3 it is clear that grid size have a larger effect on the false alarm probabilities than sample size. A smaller sample size seems preferable for strict FC-standards whereas a larger sample size seems preferable for less strict FC-standards.

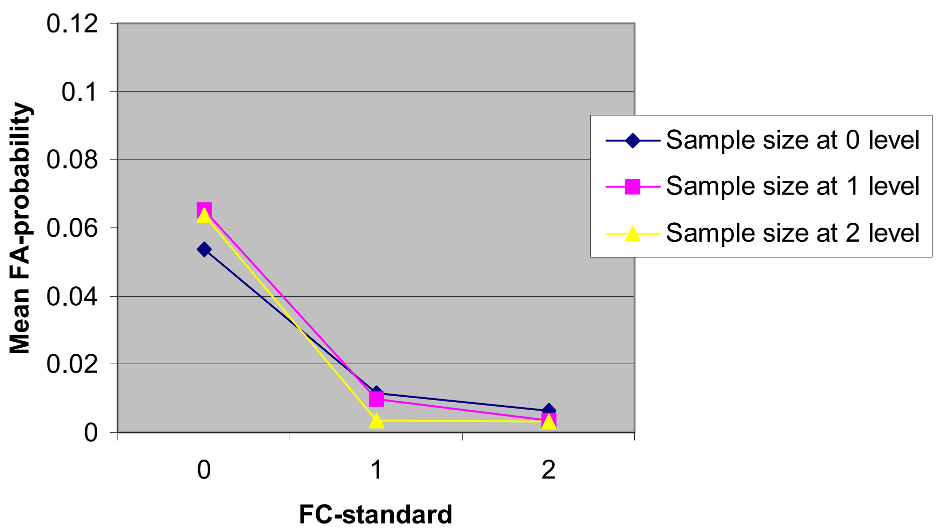

Figure 4-3: ***Sample size – FC-standard interaction for negatively correlated sensors***

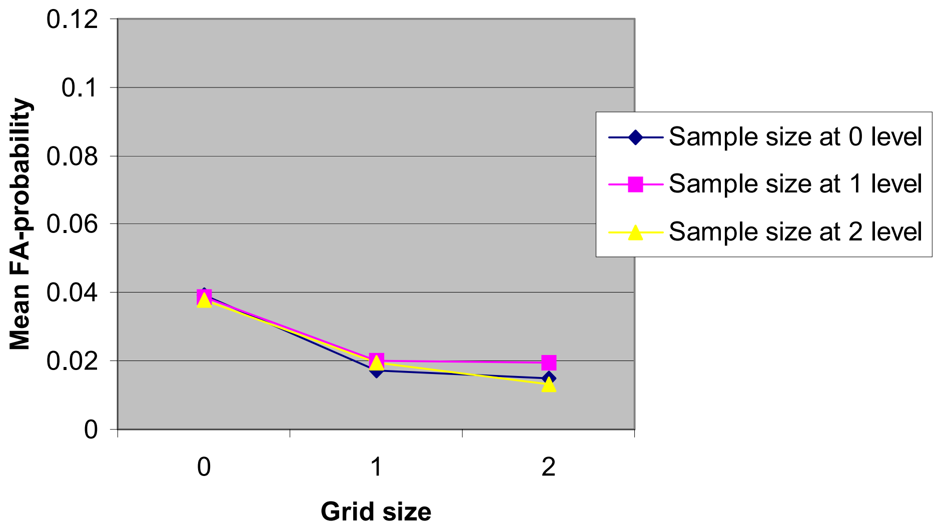

Figure 4-4: ***Sample size – Grid size interaction for negatively correlated sensors***

Figure 4-4 shows the interaction between sample size and grid size for negatively correlated sensor responses. It shows that sample size seems to affect the false alarm probabilities for larger grid sizes (i.e., more sub-regions in the static grid estimation procedure). It is hard to explain why a sample size of 1,000 would be preferable to a sample size of 10,000 when a sample size of 100,000 is preferable to a sample size of 1,000. One would expect that increasing the sample size would successively increase or decrease the false alarm rates. One must suspect that the differences between the levels of sample size in Figure 4-4 are insignificant and that it is not possible to draw any conclusion about the effect of sample size on the false alarm probabilities for negatively correlated sensor responses.

Studying the three-factor interaction between sample size, grid size, and FC-standard for negatively correlated sensor responses might give some more insights. Figure 4-5, 4-6, and 4-7 shows the interaction between sample size and grid size for different levels of FC-standard while sensor responses are negatively correlated. Increasing grid size improves the false alarm probabilities for strict FC-standards (i.e., FC-standard at the low level). However, for less strict FC-standards grid size seem to have no significance.

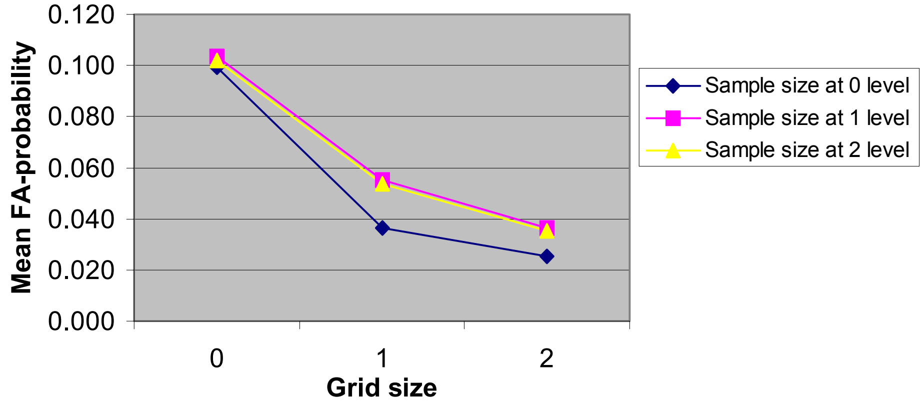

Figure 4-5: ***Sample size – Grid size interaction for negatively correlated sensor responses and FC-standard equal to 0.001 (low level)***

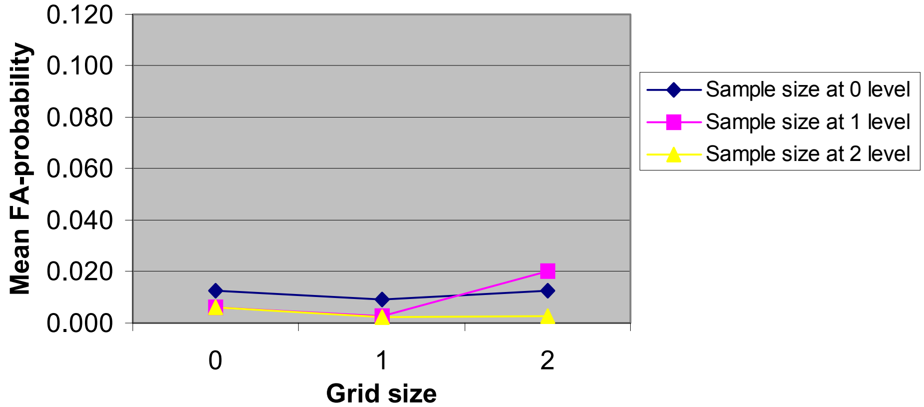

Figure 4-6: ***Sample size – Grid size interaction for negatively correlated sensor responses and FC-standard equal to 0.01 (intermediate level)***

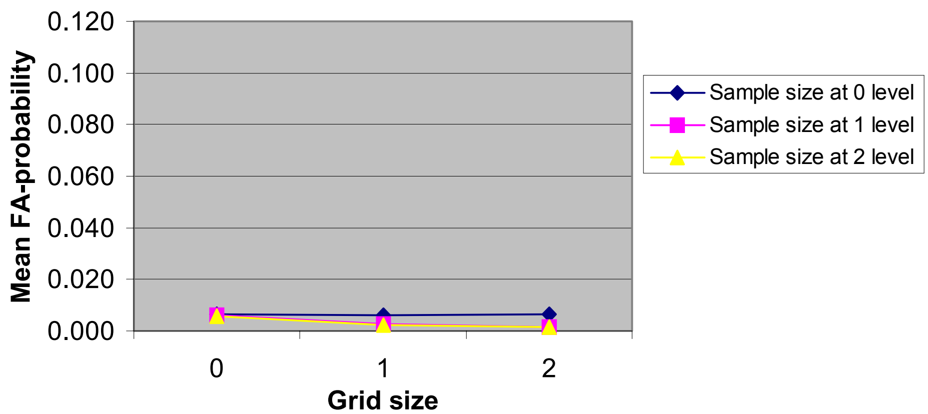

Figure 4-7: ***Sample size – Grid size interaction for negatively correlated sensor responses and FC-standard equal to 0.025 (high level)***

4.1.2 Uncorrelated Sensor Responses

Even though it has been determined that negatively correlated sensor responses outperform uncorrelated and positively correlated sensor responses it is worthwhile studying these latter scenarios in more detail. The reason being the consideration of existing detection system architectures, not easily subject to re-design. Figure 4-8 and 4-9 shows the interaction between grid size and FC-standard, and sample size and FC-standard for uncorrelated sensor responses.

The interaction plots show that sample size and grid size does matter for strict FC-standards. However, for FC-standard at the intermediate and high levels sample size is insignificant and the effect of grid size is very small.

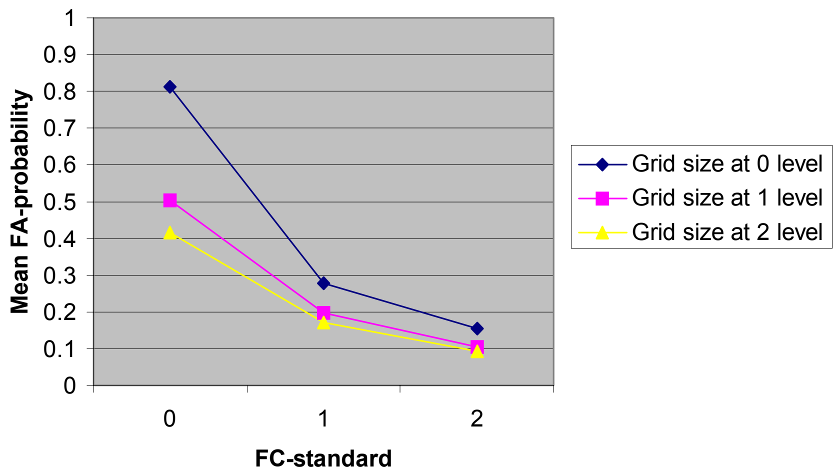

Figure 4-8: ***Grid size – FC-standard interaction for uncorrelated sensors***

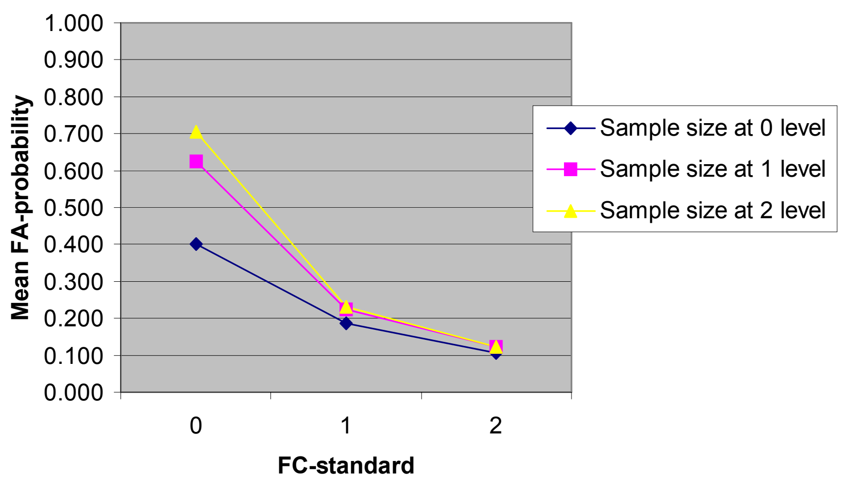

Figure 4-9: ***Sample size – FC-standard interaction for uncorrelated sensors***

Figure 4-10 shows the interaction between sample size and grid size for uncorrelated sensor responses. It is noticeable that sample size affects the false alarm probabilities more for uncorrelated sensor responses than for negatively correlated sensor responses. It is surprising that a small sample size in combination with a large grid size results in the lowest false alarm probabilities.

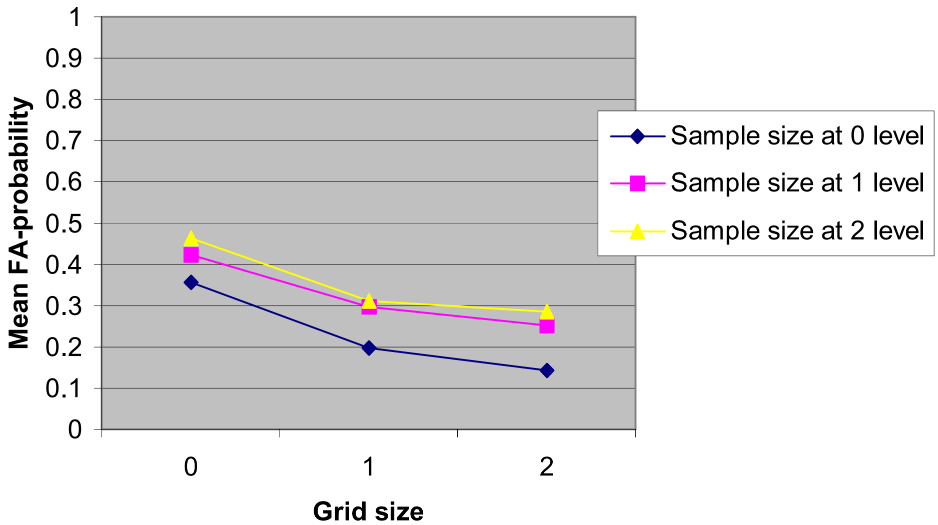

Figure 4-10: ***Sample size – Grid size interaction for uncorrelated sensors***

4.1.3 Positively Correlated Sensor Responses

Figure 4-11 shows the interaction between grid size and FC-standard and the fact that positively correlated sensors do not perform desirably. Even for a detection system where the least strict FC-standard is acceptable (i.e., 0.025), the minimum false alarm rate that can be achieved is close to 0.5. This means that half of the objects passing through the detection system would be falsely indicated as a treat. Figure 4-11 also shows that larger grid sizes would minimize the false alarm probability.

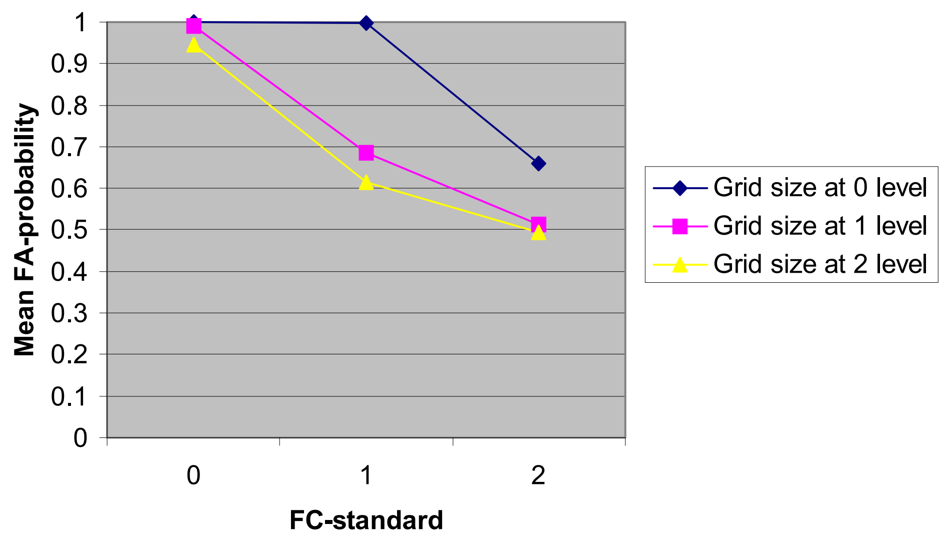

Figure 4-11: ***Grid size – FC-standard interaction for positively correlated sensors***

Figure 4-12 also shows that positively correlated sensors do not perform well. It also shows that sample size does not affect the false alarm probabilities for positively correlated sensor responses.

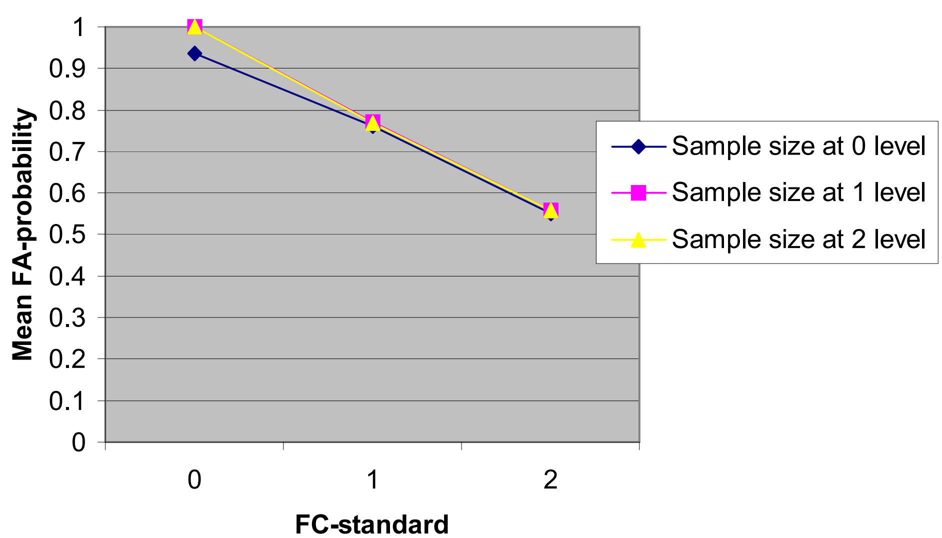

Figure 4-12: ***Sample size – FC-standard interaction for positively correlated sensors***

Figure 4-13 shows the interaction between sample size and grid size. Recalling the interaction between sample size and grid size for negatively correlated sensor responses (Figure 4-4) and for uncorrelated sensor responses (Figure 4-10) one can see that sample size is insignificant for correlated sensor responses, especially for smaller grid sizes. Sample size does however seem to have an effect when sensors are not correlated.

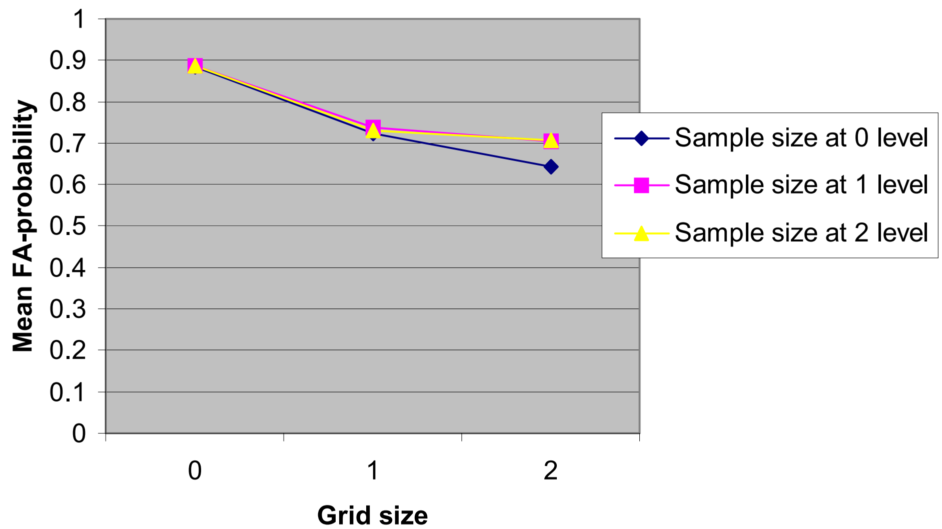

Figure 4-13: ***Sample size – Grid size interaction for positively correlated sensors***

4.2 Receiver Operating Characteristics

Presented below is a series of ROC-curves, generated to give a complementary approach to analyze the results of the experiments. Recall that the ROC is a graph plotting the probability of detection against the probability of false alarm. In general, a steeper ROC-curve indicates a better performing detection system (or sensor) because it implies that a higher probability of detection can be obtained with only a small increase in false alarm probability. The ROC-curves tend to confirm the results seen in the analysis of variance.

The ROC-curves are uniquely labeled with a combination of characters and numbers. The characters used are "s", "g", and "c" representing the factors sample size (s), grid size (g), and correlation (c) respectively. The numbers represent the level of each factor (i.e., 0 for the low level, 1 for the intermediate level, and 2 for the high level.

Figure 4-14 depicts the nine ROC-curves that were obtained for a sample size equal to 1000. They show, consistent with previous analyses, that performance increases as grid sizes increase although this is only noticeable for uncorrelated and positively correlated sensor responses. Similar graphs for sample sizes of 10,000 and 100,000 are attached in Appendix D (Figure D-1 and D-2). They show that grid size has little or no effect on the false alarm probability for large sample sizes.

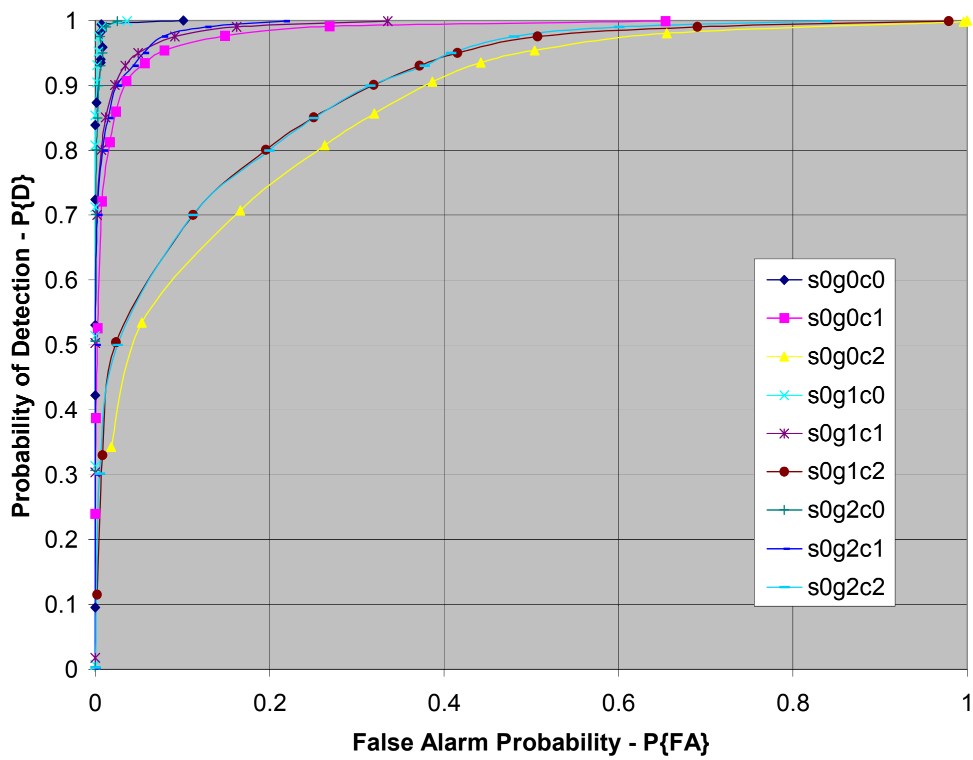

Figure 4-14: ***ROC-curves for sample size equal to 1000***

Figure 4-15 depicts the ROC-curves for a grid size of 15. The cluster of ROC-curves for positively correlated sensor responses indicates that performance increases as sample size decreases. This effect is insignificant for negatively or uncorrelated sensors. Similar graphs, for grid sizes of 5 and 10 are attached in Appendix D (Figure D-3 and D-4). They do not provide any new insights.

Figures 4-16, 4-17, and 4-18 depict the ROC-curves for correlation equal to -0.4, 0, and +0.4, respectively. Note that the scale on the False Alarm Probability axis is different for the figures. The ROC-curves clearly shows that negatively correlated sensors outperform uncorrelated and positively correlated sensors.

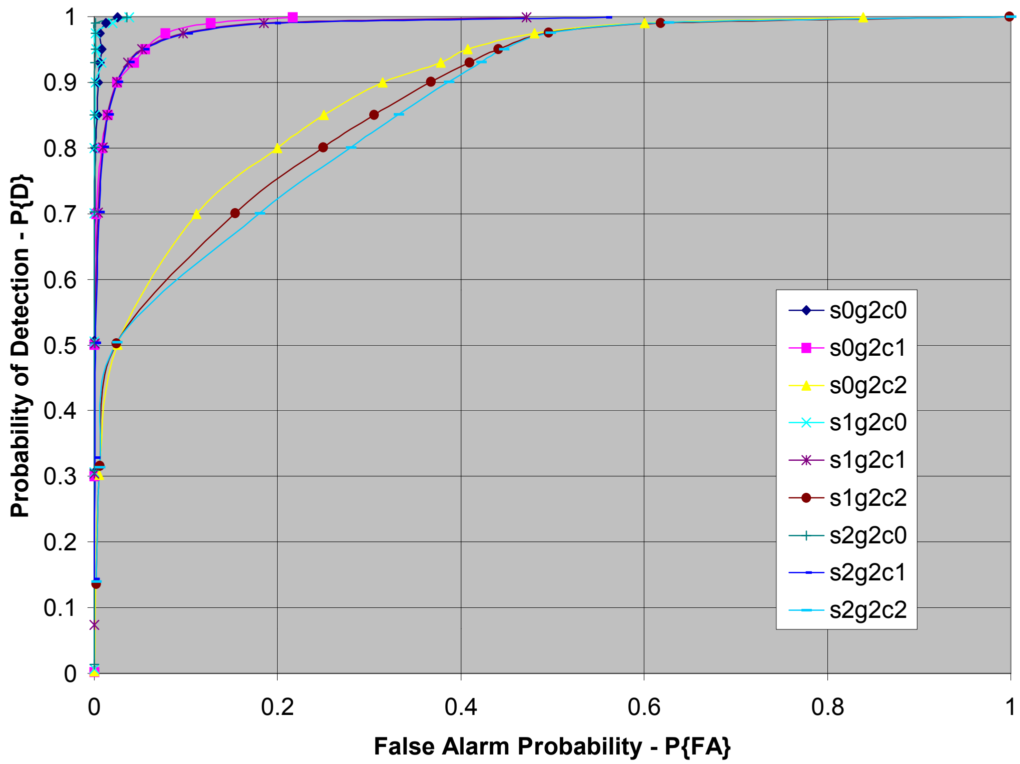

Figure 4-15: ***ROC-curves for grid size equal to 15***

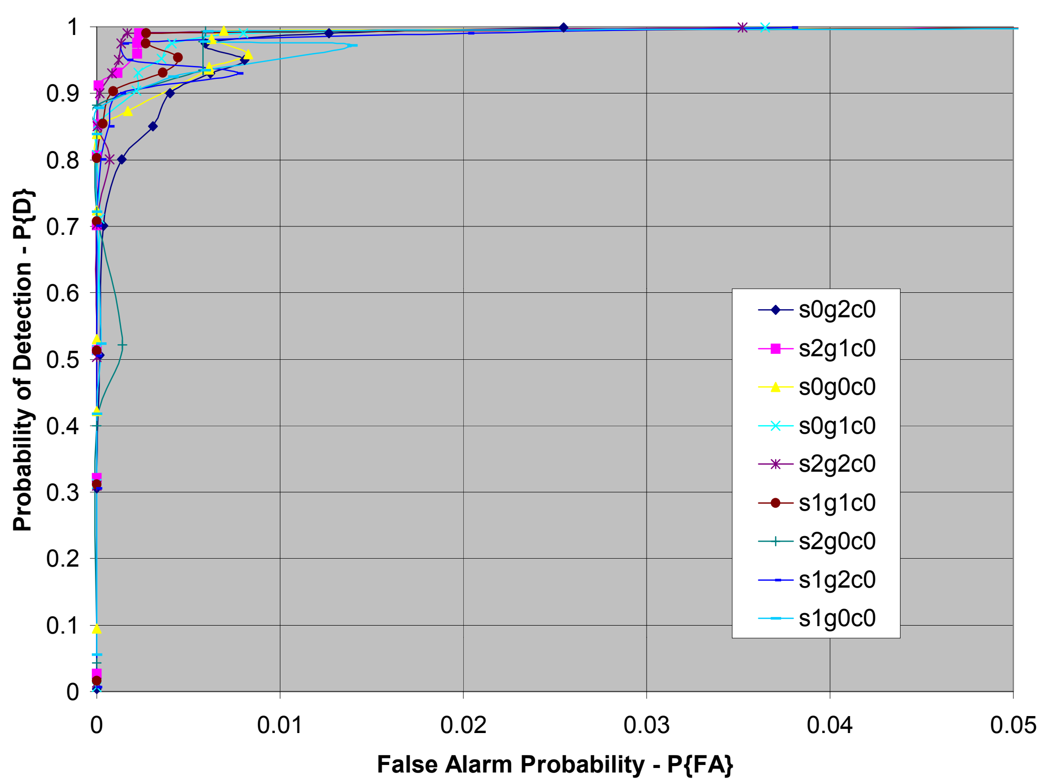

Figure 4-16: ***ROC-curves for correlation equal to -0.4***

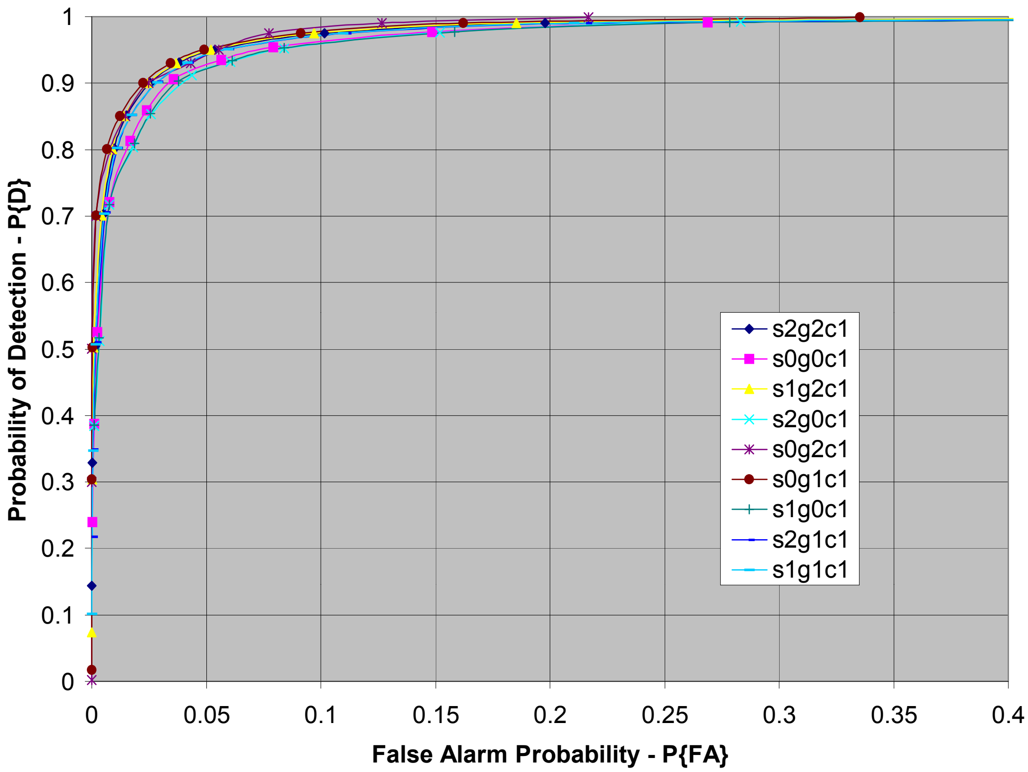

Figure 4-17: ***ROC-curves for correlation equal to 0***

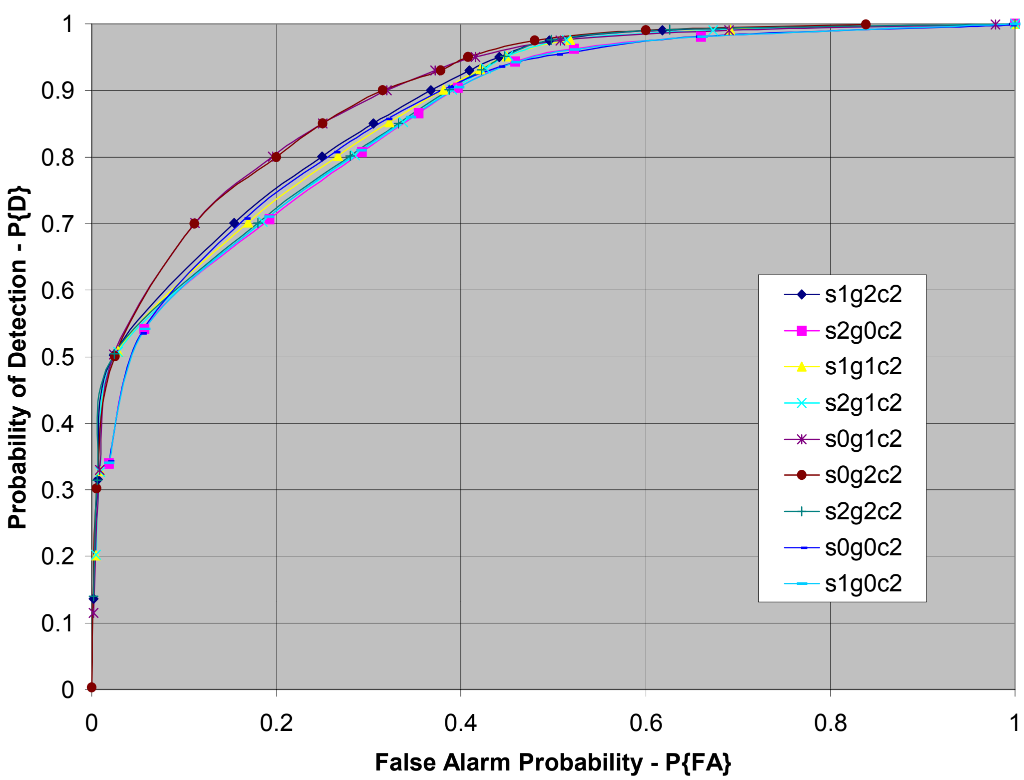

Figure 4-18: ***ROC-curves for correlation equal to +0.4***

Figure 4-19 depicts the four steepest (i.e., best performing) ROC-curves (note the scale change on the False Alarm Probability axis). Even though sample size does not seem significant in previous analysis this graphical representation indicates that a large sample size should be selected for optimum performance. It is not possible to distinguish between the intermediate and the high level of grid size.

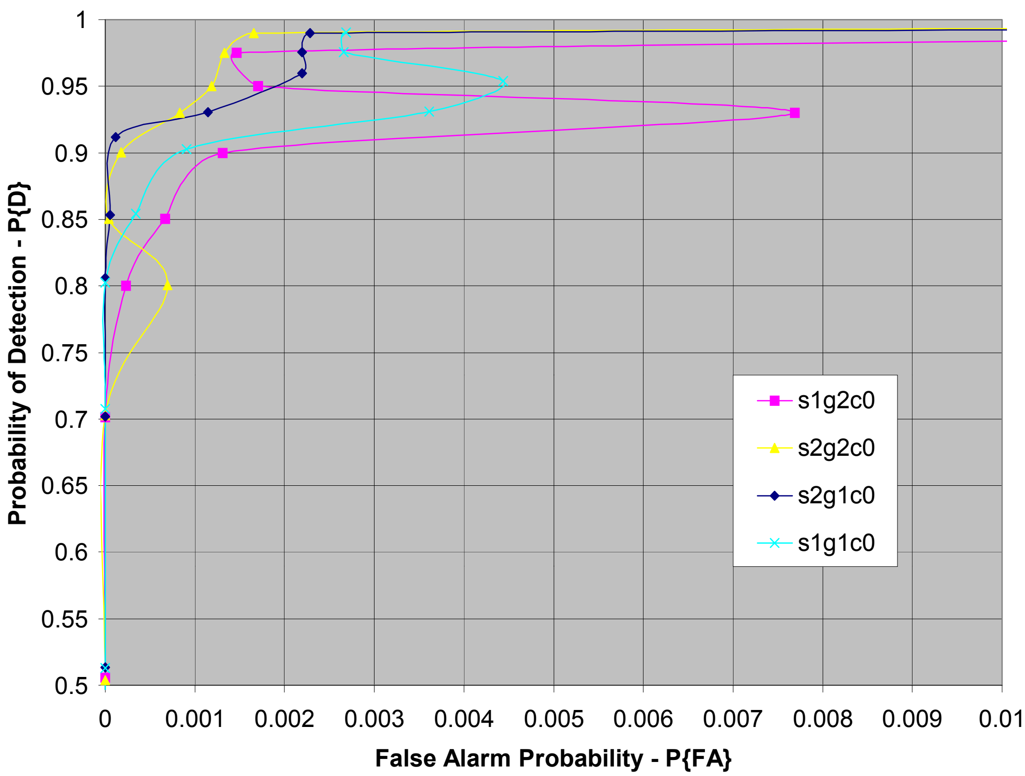

Figure 4-19: ***The four steepest ROC-curves (correlation = -0.4)***

CHAPTER 5

5 Summary

This thesis began with a discussion of the use of detection systems in different real-world applications. In order to optimally design, implement, and utilize detection systems, one needs a deeper understanding of the performance of detection systems in different situations. How can designers, operators, and users of detection systems optimally utilize system technologies and procedures? To answer this question and similar issues involved in the design and operation of detection systems, one is faced with the false alarm, false clear tradeoff. Given a particular detection system and a false clear standard, there is a minimum false alarm rate that can be achieved. Researchers have developed methods to determine this false alarm rate. The objective of the research presented here is to analyze how different system design variables and system parameters affect the overall system false alarm probabilities. The method used to obtain the false alarm probabilities is a modified and generalized version of a method developed by Jacobson et al., (2001) with additions for modeling dependency among sensor responses.

5.1 Conclusions

The objective of this research was to develop guidelines that can be of help in designing detection systems. More specifically, to study how system design variables (controllable) such as sample size and grid size, and model parameters (uncontrollable) such as correlation, FC-standard, and distribution of sensor responses affect the FAFCT (False Alarm False Clear Tradeoff).

The research questions addressed in this thesis were:

Given a particular detection system and a pre-specified false clear standard,

- What sample sizes are needed to obtain an appropriate FAFCT?
- How should the grid size in the static grid estimation procedure be set to assess the distribution of sensor responses?
- How does dependency among sensors affect the FAFCT?

In the absence of real sample data from a particular detection system, such as a Salmonella detection system, it is hard to answer the first two research questions. What is an appropriate FAFCT depends on the nature of the detection system and the implication caused by possible false alarms and false clears.

In general, the research shows that sample size has a very small effect on the FAFCT compared to other factors. The analysis shows, somewhat surprisingly, that a smaller sample size seems to imply lower false alarm probabilities, except for cases when sensor responses are negatively correlated and/or a large FC-standard is acceptable. However, the differences in false alarm probabilities between sample sizes at different levels are very small, especially between the intermediate and high level (i.e., 10,000 and 100,000) of sample size. The analysis of ROC-curves communicates the same results.

The analysis shows that a combination of a small sample size and a large grid size seems favorable in general. However, the optimum detection system setting according to the results is large sample size, combined with a large grid size in a system with negatively correlated sensors.

Studying the effect of sample size for the three different scenarios: negatively correlated sensor responses, uncorrelated sensor responses; and positively correlated sensor responses reveals that samples size is insignificant when sensor responses are correlated. However, for uncorrelated sensor responses, sample size does seem to have a small effect on the false alarm probabilities. Hence, sample size cannot be dismissed as an influential factor in the design of detection systems although there will be cases, especially for strongly correlated sensors, where sample size will be insignificant.

It was shown that grid size could have a significant effect on the performance of detection systems. In a detection system with negatively correlated sensors and strict FC-standard, increasing grid size will drastically increase the performance (i.e., lower the false alarm probability).

A better approach for evaluating the performance of the static grid estimation procedure would include the use of real sample data from a known detection system.

Experiments would need to be run for different grid sizes until the FAFCT methodology returns false alarm probabilities that are comparable with the false alarm rates of the known detection system.

The analysis clearly shows that correlation has the most influence on the FAFCT. Negatively correlated sensor responses outperform uncorrelated and positively correlated sensor responses with large margins, especially for strict FC-standards. Hence, sensors that are designed to detect different characteristics of an object (i.e., to some extent tending to disagree) perform better. Interestingly, the differences between negatively correlated sensor responses and uncorrelated sensor responses are not as noticeable for large FC-standards.

The results of this research have applicability in many fields of study including airport security, bacteria detection, health care, industrial quality control, and military applications.

5.2 Future Research

Future research could include performing more complex experimental designs where more factors are studied and also experiments where the existing factors are studied for other levels of sample size, grid size, correlation, and FC-standard. A particular case for further study is the use of strict FC-standards, negatively correlated sensor responses, large grid sizes, and large sample sizes.

Case studies for a real-world application should be conducted. Real-world data could help assess the performance of the static grid estimation procedure.

Other grid estimation procedures such as the Dynamic Grid Estimation procedure developed by Simms (1997) should be studied for the case with dependent sensor responses.

Finally, modeling of more complex detection systems with three or more sensors and possibly decentralized decisions could give valuable insights to the research of detection systems.

CHAPTER 6

6 References

Alhakeem, S. and Varshney, P. K., (1995). "A Unified Approach to the Design of Decentralized Detection Systems," *IEEE Trans. Aerospace and Electronic Systems*, v. 31, no. 1, pp. 9-20.

Blum, R. S., (1996). "Necessary Conditions for Optimum Distributed Sensor Detectors under the Neyman-Pearson Criterion," *IEEE Trans. on Information Theory,* vol. 42, no. 3, pp. 990-994.

Chair, Z. and Varshney, P. K., (1986). "Optimal Data Fusion in Multiple Sensor Detection Systems," *IEEE Trans. Aerospace and Electronic Systems*, v. 22, no. 1, pp. 98-101.

Christer, A. H., (1994). "Modelling the Quality of Automatic Quality Checks," *Journal of The Operational Research Society*, v. 45, pp. 806-816.

Dasarathy, B.V., (1994). "Decision Fusion," *IEEE Computer Society Press*, Los Alamitos, CA, 1994.

Devroye, Luc., (1986). "Non-Uniform Random Variate Generation," *Springer-Verlag, New York*, New York.

Drakopoulus, E. and Lee, C. C., (1991). "Optimum Multisensor Fusion of Correlated Local Decisions," *IEEE Trans. Aerospace and Electronic Systems*, v. 27, no. 4, pp. 593-605.

Echard, J. D., (1991). "Estimation of Radar Detection and False Alarm Probabilities," *IEEE Trans. Aerospace and Electronic Systems*, v. 27, no. 2, pp. 255-260.

Friedl. (1989). "Is it possible to reduce false alarms?," ICCST, Zurich, Switzerland.

Haykin, S., (1994). "Neural Networks: A Comprehensive Foundation," *Macmillan,* New York.

Jacobson, S. H, Kobza, J.E., and Easterling, A.E., (2001). "A Detection Theoretic Approach to Modeling Aviation Security Problems using The Knapsack Problem," *IIE Transactions,* vol. 33, no. 9, pp. 747-759.

Kam, M., Zhu, Q. and Gray, W.S., (1992). "Optimal Data Fusion of Correlated Local Decisions in Multiple Sensor Detection Systems," *IEEE Trans. Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 916-920.

Kobza and Jacobson. (1996). "Addressing the Dependency Problem in Access Security System Architecture Design," *Society for Risk Analysis*, v. 16, no. 6, pp. 801-812.

Kobza and Jacobson. (1997). "Probability models for access security system architectures," *Journal of the Operational Research Society*, v.48, no. 3, pp. 255-263.

Montgomery, C. D., (1991). "Design and Analysis of Experiments," *John Wiley & Sons*, New York.

Moore, W. G., Hutchins, G. M. and Miller, R. E., (1986). "A New Paradigm for Hypothesis Testing in Medicine, with Examination of The Neyman Pearson Condition," *Theoretical Medicine 7,* pp. 269-282.

Pate-Cornell. (1986). "Warning Systems in Risk Management," *Society for Risk Analysis*, v. 6,no. 2,pp. 223-234.

Rao, N. S. V. and Lyengar, S. S., (1995). "Distributed decision fusion under unknown distributions," *Optical Engineering,* vol. 35, no. 3, pp. 617-624.

Reibman, A. R. and Nolte, L. W., (1987). "Design and Performance Comparison of Distributed Detection Networks," *IEEE Trans. Aerospace and Electronic Systems*, v. 23, no. 6, pp. 789-797.

Reibman, A. R. and Nolte, L. W., (1987). "Optimal Detection and Performance of Distributed Sensor Systems," *IEEE Trans. Aerospace and Electronic Systems*, v. 23, no. 1, pp. 24-30.

Simms, A. E., (1997). "A Stochastic Approach to Modeling Aviation Security Problems Using the Knapsack Problem," *Master Thesis,* Virginia Polytechnic Institute and State University.

Tenney, R. R. and Sandell, N. R., (1981). "Detection With Distributed Sensors," *IEEE Trans. Aerospace and Electronic Systems*, v. 17, no. 4, pp. 98-101.

Thomopoulos, S. C. A., Viswanathan, R. and Bougoulias, D. C., (1987). "Optimal Decision Fusion in Multiple Sensor Systems," *IEEE Trans. Aerospace and Electronic Systems*, v. 23, no. 5, pp. 644-653.

Van Trees, H.L., (1968). "Classical Detection and Estimation Theory," *John Wiley & Sons.*

Varshney, V., (1996). "Distributed Detection and Data Fusion," *Springer-Verlag, New York*, New York.

Appendix A

Appendix A contains the C-Code for computational evaluation of the FAFCT-Methodology.

```
/*****************************************************************/
/*The following is code for the False Alarm, False Clear research    */
/*problem.  This code applies to the two-device example, but is      */
/*easily modified for the five-device example CORRELATED             */
/*DEVICE RESPONSES. This version uses the new definitions            */
/*of P(FA)=P(A|NT).  Programmer: Amy Simms, Modifications by: Johan  */
/*Nyberg                                                             */
/*****************************************************************/

#include <stdio.h>            /*included header files*/
#include <stdlib.h>
#include <math.h>

#define M 1000          /*number of samples*/
#define m 10                  /*grid size*/
#define size 100        /*size of arrays - m^p*/
#define p 2             /*number of devices*/
#define sample 50       /*# of Monte Carlo replications*/

/*function prototypes*/

void generate(float Q1[size],float Q2[size],int locate[size][p]);
double findroot(double randnr, double (*f)(double));
double findroot2(double res, double randnr, double
(*f)(double,double));
double f1t(double x);
double f1nt(double x);
double f2t(double res,double x);
double f2nt(double res,double x);
void heapsort(float Q1[size], float Q2[size], int locate[size][p]);
void reheapdown(float Q1[size], float Q2[size], int locate[size][p],
int root, int bottom);
void swapf(float *element1, float *element2);
void swapi(int *element1, int *element2);
float greedy(float PFC, float Q1[size], float Q2[size], int
X[12][size], int locate[size][p], int fcnt, float *zt);
void output(float z[12][sample],float zt[12][sample],int
locate[size][p],int X[12][size]);

main()
{                  /*define variables*/

float z[12][sample];  /*sum of non-threat responses in CLEAR region*/
float zt[12][sample];   /*sum of threat responses in CLEAR region*/
int X[12][size] = {0};  /*accept indicator*/
int locate[size][p];    /*original location*/
float Q1[size];               /*threat responses - float because of
                              division in sort procedure*/
float Q2[size];         /*non-threat responses*/
```

```
float PFC[12]= {0.999, 0.7, 0.5, 0.3, 0.2, 0.15, 0.1, 0.07, 0.050,
0.025, 0.01, 0.001};
/*total allowable probability of False Clear- standard set by FAA*/
int cnt;
int seed;

for (seed=0; seed< sample; seed++)
{     /*initializes an entry point in a pseudorandom number stream.*/

      srand48(seed+1);        /* srand48 must be invoked before drand48
                              is called*/
      generate(Q1,Q2,locate);
      heapsort(Q1,Q2,locate);

      for (cnt=0; cnt < 12; cnt++)        /* Going through 12 loops
                                          with PFC set to PFC[cnt] */
      {

            zt[cnt][seed]=0;
            z[cnt][seed] =
            greedy(PFC[cnt],Q1,Q2,X,locate,cnt,&zt[cnt][seed]);
            /*PFC=PFC/10;*/

      }
}

output(z,zt,locate,X);
return;
}

/*****************************************************************/
/*Function: generate                                             */
/*This function creates randomly generated data to be used in the    */
/*knapsack algorithm.  Note that the data read in IS NOT the         */
/*probabilistic values q1 and q2. Since many knapsack algorithms     */
/*are integer programming algorithms, the data read is the sum of    */
/*non-threat and threat responses in each subregion. In all          */
/*algorithms, the limit on q1, that is PFC must be multiplied by M.  */
/*Once the knapsack algorithm is complete, all values must be divided*/
/*to be M to get the corresponding probabilities. When z             */
/*(sum of q2 in CLEAR region) is divided by M, it represents P{NA|NT}*/
/*YOU MUST* make sure there are the appropriate number of device     */
/*locations                                                          */
/*****************************************************************/

void generate(float Q1[size],float Q2[size],int locate[size][p])
{
int response1[m][m]={0};      /*original storage grid*/
int response2[m][m]={0};
int cnt,cnt1,cnt2;            /*sample counter*/
int devcnt;       /*device counter*/
int devresp[m];         /*stores responses of each device*/
double u, x;            /*random numbers*/
float temp;       /*random number between 0 and m*/
int i,j;
double root1, root2;
```

```
for (cnt=1; cnt<=M; cnt++)          /*Samples M IID data points from
                                    each of the two*/
{           /* conditional joint probability density functions.*/

      for (cnt2=0; cnt2<2; cnt2++)
      {

            if (cnt2==0)
            {

                  u = drand48();          /*threat item*/
                  root1=findroot(u,f1t);  /*function returning 1st
                                    rand. variate*/
                  u = drand48();
                  root2= findroot2(root1, u, f2t);

            }
            else
            {

                  u = drand48();
                  root1=findroot(u, f1nt);
                  u=drand48();
                  root2=findroot2(root1,u,f2nt);

            }

            temp = m*root1;
            devresp[0] = floor(temp);  /*round down to find grid
                                          square*/
            temp = m*root2;
            devresp[1] = floor(temp);

            if (cnt2==0)
            {

                  response1[devresp[0]][devresp[1]]++;

            }/*Sum of threat responses*/
            else
            {

                  response2[devresp[0]][devresp[1]]++;

            }/*Sum of non threat resp*/

      }

}

cnt=0;                        /*convert to one-dimensional arrays*/
for (cnt1=0; cnt1<m; cnt1++)
{
      for (cnt2=0; cnt2<m; cnt2++)
      {
```

```
            Q1[cnt]=response1[cnt1][cnt2];      /*Sum of threat
                          responses in a one-dimensional array*/
            Q2[cnt]=response2[cnt1][cnt2];      /*Non threat responses
                          in a one-dimensional array*/
            locate[cnt][0]=cnt1;
            locate[cnt][1]=cnt2;          /*What is this doing?*/
            cnt++;

      }

}
return;
}

/**********************************************************************/
/*Functions: f1t,f1nt,f2t,f2nt                                        */
/*These four functions defines the density functions used to simulate*/
/*correlated sensor responses.                                        */
/**********************************************************************/

double f1t(double x)
{/*function evaluating the 1st marginal cond. on a threat*/

      return (double) 1.00339-0.5*exp(-33.0667*x)-0.503392*exp(-5*x);

}

double f1nt(double x)
{/*function eval. the 1st marg. cond. on a non-threat*/

      return (double) -0.00339183+0.00339183*exp(5*x)+2.17914*pow(10,-
      15)*exp(33.0667*x);

}

double f2t(double res, double x)
{

      return (double) (0.152235*exp(33.0667*res-
      50*x)+10.5083*exp(5*res-0.1*x)-10.5083*exp(5*res)-
      0.152235*exp(33.0667*res))/(-exp(5*res)-
      0.152235*exp(33.0667*res));

}

double f2nt(double res, double x)
{

      return (double) (3.27099*pow(10,-24)*exp(5*res)+6.85141*pow(10,-
      13)*exp(33.0667*res)-6.85141*pow(10,-13)*exp(33.0667*res+0.1*x)-
      3.27099*pow(10,-24)*exp(5*res+50*x))/(-0.0169591*exp(5*res)-
      7.20569*pow(10,-14)*exp(33.0667*res));

}
```

```
/*****************************************************************/
/*Functions: findroot, findroot2                                 */
/*These two functions finds the correlated responses from device 1   */
/*and 2 resp. using a numerical search method                   */
/*(i.e., the bisection method)                                   */
/*****************************************************************/

double findroot(double randnr, double (*f) (double))
{     /*Uses the bisection method to find roots*/

double temp, soln, llim, rlim, e;
llim=0; rlim=1; e=0.00001;
      while (rlim-llim>2*e)
      {

      soln = (llim+rlim)/2;
      temp = f(soln)-randnr;

            if (temp*(f(llim)-randnr)>0)
            {

            llim = soln;
            }
            else
            {

            rlim = soln;

            }

      }

return (llim+rlim)/2;

}

double findroot2(double res, double randnr, double (*f) (double,
double)) {  /*Uses the bisection method to find roots*/
   double temp, soln, llim, rlim, e;
   llim=0; rlim=1; e=0.00001;
   while (rlim-llim>2*e) {
       soln = (llim+rlim)/2;
       temp = f(res,soln)-randnr;
       if (temp*(f(res,llim)-randnr)>0)
           llim = soln;
       else
               rlim = soln;
    }
    return (llim+rlim)/2;
}
```

```
/*****************************************************************/
/*Function:  heapsort                                            */
/*Reference: Pascal Plus Data Structures, 3rd Edition, Dale & Lilly, */
/*p.689. This function uses a heap structure to sort the arrays Q1,  */
/*Q2, and locate. A heap is an array that is ordered in a tree-like  */
/*structure, where the first entry in the array is the root and the  */
/*second and third entries are the children of that root. Entries 4  */
/*and 5 are the children of the second entry, entries 6 and 7 are the*/
/*children of the third entry, and so on. Each entry may have at most*/
/*2 children. Futhermore (for descending order), each child must be  */
/*greater than its parent. Because of the order property of a heap,  */
/*the minimum value in the array must be the root node. The general  */
/*approach of the function Heapsort is:                              */
/*1.  Arrange the original array elements in a heap                  */
/*2.  Remove the root (minimum) element from the heap and put it in  */
/*its appropriate position in the array.                             */
/*3.  Arrange the remaining elements in a heap - this puts the next  */
/*smallest element in the root position.                             */
/*4.  Repeat from step 2 until there are no more elements in the heap*/
/*This version of heapsort uses Q2/Q1 as the measure of size         */
/*Efficiency of Heapsort:  Heapsort is O(nlog(2)n)                   */
/*****************************************************************/

void heapsort(float Q1[size], float Q2[size], int locate[size][p])
{
int index;

for (index=size/2-1; index>=0; index--)
{
      reheapdown(Q1,Q2,locate,index,size-1);
}     /*build original heap from unsorted elements*/

for (index=size-1; index>=1; index--)
{
      swapf(Q2+0, Q2+index);/*swap the root (current smallest)       */
      swapf(Q1+0, Q1+index);/*value with the last unsorted value -   */
      swapi(&locate[0][0], &locate[index][0]);/* swap Q1, Q2, and    */
      swapi(&locate[0][1], &locate[index][1]);/*locate               */
      reheapdown(Q1,Q2,locate,0,index-1); /*reheap the remaining part
                                          of the array*/
}
return;
}

/*****************************************************************/
/*Function:  reheapdown                                          */
/*Reference:  Dale & Lilly, p.616                                */
/*This function creates a heap, as described above, from an array   */
/*Heap order is restored, starting at the root, by the size measure  */
/*Q2/Q1.                                                             */
/*****************************************************************/

void reheapdown(float Q1[size], float Q2[size], int locate[size][p],
int root, int bottom)
{
int minchild;           /*index of child with smaller value*/
int rightchild;         /*index of right child node*/
```

```
int leftchild;          /*index of left child node*/

leftchild = root*2+1;
rightchild = root*2 + 2;
if (leftchild <= bottom)      /*root in not a leaf (a leaf has no
                              children)*/
{
      if (leftchild == bottom)      /*there is only one child node*/
      {minchild = leftchild;}
      else              /*pick the smaller of the two children*/
      {
            if ((Q2[leftchild]/Q1[leftchild]) <
            (Q2[rightchild]/Q1[rightchild]))
            {minchild = leftchild;}
            else
            {minchild = rightchild;}
      }
      if ((Q2[root]/Q1[root]) > (Q2[minchild]/Q1[minchild]))
            /*order property not intact*/
      {
            swapf(Q2+root, Q2+minchild);        /*swap and reheap*/
            swapf(Q1+root, Q1+minchild);
            swapi(&locate[root][0], &locate[minchild][0]);
            swapi(&locate[root][1], &locate[minchild][1]);
            reheapdown(Q1,Q2,locate,minchild,bottom);
      }
}
return;
}

/*******************************************************************/
/*Function: swapi                                                  */
/*This function swaps two integer elements by address              */
/*******************************************************************/

void swapi(int *element1, int *element2)
{
int temp;

temp = *element1;
*element1 = *element2;
*element2 = temp;
return;
}

/*******************************************************************/
/*Function: swapf                                                  */
/*This function swaps two float elements by address                */
/*******************************************************************/

void swapf(float *element1, float *element2)
{
float temp;

temp = *element1;
*element1 = *element2;
*element2 = temp;
```

```
return;
}

/******************************************************************/
/*Function: greedy                                                */
/*Reference: Knapsack Problems: Algorithms & Computer             */
/*Implementations, Martello & Toth  p. 27-29                      */
/*This function applies the greedy algorithm to a knapsack problem   */
/*It returns z, the total accumulated true clear probability      */
/*It also modifies the X vector to reflect which subregions are   */
/*rejected/accepted to the Clear region                           */
/******************************************************************/

float greedy(float PFC, float Q1[size], float Q2[size], int
X[12][size], int locate[size][p], int fcnt, float *zt)
{
float sigma;      /*accumulated threat responses in clear region*/
float z;/*accumulated non-threat responses in clear region*/
int jstar;  /*index of subregion with largest true clear probability*/
int cnt;                /*counter*/
int Xthis[size]={0};
int index;

sigma = PFC*M;    /*initialization*/
z = 0;
jstar = 0;

for (cnt=0; cnt<size; cnt++)
{
      index = m*locate[cnt][0]+locate[cnt][1];
      if (Q1[cnt] > sigma)
      {
            Xthis[cnt]=0;
      }
      else
      {
            Xthis[cnt]=1;
            X[fcnt][index]++; /*accept subregion*/
            sigma = sigma - Q1[cnt];
            z = z + Q2[cnt];
            *zt = *zt + Q1[cnt];
      }

      if (Q2[cnt] > Q2[jstar])
      {
            jstar = cnt;
      }
}

if ((Q2[jstar] > z)&(Q1[jstar]<=sigma))
{
      z = Q2[jstar];
      *zt = Q1[jstar];
      for (cnt=0; cnt<size; cnt++)
      {
            if (Xthis[cnt]==1)
            {
```

```
                  index=m*locate[cnt][0]+locate[cnt][1];
                  X[fcnt][index]--;
            }
      }
      X[fcnt][jstar]++;
}
return z;
}


/******************************************************************/
/*Function: output                                                */
/*This function computes and outputs to file the subregions contained*/
/*in the CLEAR and ALARM regions, P{FC}, P{FA}                    */
/*MUST BE edited for the appropriate number of devices            */
/******************************************************************/

void output(float z[12][sample],float zt[12][sample],int
locate[size][p],int X[12][size])
{

int cnt1,cnt2,cnt3;
float accumz, accumzt;
float PFC[12]= {0.999, 0.7, 0.5, 0.3, 0.2, 0.15, 0.1, 0.07, 0.05,
0.025, 0.01, 0.001};
FILE *out_ptr;          /*pointer to output file*/
float xtemp;
int index;

out_ptr = fopen("s0g1c02.out", "w");

fprintf(out_ptr,"The following results apply to a grid of size 10, a
sample of size 1k, and -0.4 correlation coefficient\n\n");

for (cnt1=0; cnt1<12; cnt1++)
{
      fprintf(out_ptr,"\nThe results for allowable P{FC} =
      %e:\n",PFC[cnt1]);
      fprintf(out_ptr,"Greedy Algorithm:\n\n");
      fprintf(out_ptr,"Run    P{FC}                   P{FA}\n");
      accumz=0;
      accumzt=0;
      for (cnt2=0; cnt2<sample; cnt2++)
      {
            zt[cnt1][cnt2]=zt[cnt1][cnt2]/M;
            accumzt=accumzt+zt[cnt1][cnt2];
            z[cnt1][cnt2]=(1-(z[cnt1][cnt2])/M);
            accumz=accumz+z[cnt1][cnt2];
            fprintf(out_ptr,"%d     %.15f
                  %.15f\n",cnt2+1,zt[cnt1][cnt2],z[cnt1][cnt2]);
      }
      fprintf(out_ptr,"AVG    %.15f
            %.15f\n",accumzt/(sample),accumz/(sample));
      /*fprintf(out_ptr,"\nThe probability of each grid square
      appearing in the CLEAR region is:\n");*/
      /*fprintf(out_ptr,"Loc1 Loc2  Prob\n");*/
      /*for (cnt2=0; cnt2<m; cnt2++)*/
```

```
        /*{*/
            /*for (cnt3=0; cnt3<m; cnt3++)*/
            /*{ */
                /*index = m*cnt2+cnt3;*/
                /*xtemp=X[cnt1][index];*/
                /*fprintf(out_ptr,"%d   %d
                %f\n",cnt2,cnt3,xtemp/sample);*/
            /*}*/
        /*}*/
        /*PFC=PFC/10;*/
    }
    return;
    }
```

Appendix B

Below is a listing of the false alarm probabilities for the 81 treatment combinations. The four digit number combination at the top of each column uniquely identifies each of the 81 treatment combinations. Each digit can take on a value of 0 (low level), 1 (intermediate level), or 2 (high level). The first digit indicates the level of factor A (Sample size), the second digit indicates the level of factor B (Grid size), the third digit indicates the level of factor C (correlation), and the fourth digit indicates the level of factor D (FC-standard). The factors and their respective levels are shown in Table 3-1 in Chapter 3.5, Experimental Design.

The bolded values in italics are data points considered outliers. Outliers are defined as data points that diverge more than three standard deviations from the mean of the fifty replications.

0000	0100	0200	1000	1100	1200	2000	2100	2200
0.094	0.045	0.032	0.1017	0.0528	0.0149	0.10009	0.05237	0.0348
0.097	0.005	0.01	0.104	0.0513	0.0364	0.10191	0.05426	0.03572
0.097	0.057	0.036	0.1041	0.0546	0.0354	0.10252	0.05445	0.03553
0.1	0.007	0.01	0.1018	0.0553	0.0372	0.10185	0.05421	0.03518
0.083	0.008	0.012	0.103	0.0536	0.0339	0.10254	0.05384	0.03543
0.106	0.053	0.035	0.105	0.0549	0.0369	0.10074	0.05275	0.03471
0.107	0.054	0.036	0.1059	0.0581	0.0402	0.10024	0.05243	0.03441
0.111	0.06	0.005	0.1039	0.0565	0.0403	0.10268	0.05403	0.03602
0.101	0.005	0.005	0.1026	0.0562	***0.0083***	0.10191	0.05375	0.03562
0.101	0.059	0.039	0.0991	0.0535	***0.0126***	0.10181	0.05412	0.03538
0.109	0.056	0.039	0.1076	0.055	0.0371	0.10114	0.05308	0.03448
0.104	0.058	0.038	0.1051	0.0559	0.036	0.1036	0.05539	0.03656
0.095	0.002	0.003	0.1004	0.0555	0.0384	0.10209	0.05377	0.03588
0.098	0.056	0.037	0.1033	0.0552	0.0355	0.10317	0.05408	0.0352
0.09	0.048	0.037	0.1064	0.0586	0.0411	0.10129	0.0529	0.03433
0.09	0.014	0.013	0.1031	0.0564	0.0422	0.10309	0.05469	0.03592
0.119	0.063	0.044	0.1117	0.0574	0.0415	0.10195	0.05238	0.03531
0.097	0.046	0.036	0.1021	0.0525	0.0346	0.10008	0.05227	0.03415
0.096	0.051	0.037	0.1078	0.0562	0.0405	0.10298	0.05418	0.03605
0.014	0.014	0.03	0.1066	0.0563	0.0402	0.10205	0.05299	0.03498
0.097	0.049	0.038	0.0964	0.0519	0.0341	0.10111	0.05315	0.03529
0.102	0.004	0.01	0.1038	0.0555	0.0394	0.10257	0.05349	0.03549
0.115	0.007	0.005	0.1049	0.0542	0.0388	0.10135	0.05356	0.03533
0.099	0.052	0.039	0.1027	0.0503	0.0402	0.1032	0.05426	0.03553
0.106	0.003	0.003	0.1024	0.057	0.0436	0.10178	0.05338	0.03511
0.117	0.017	0.016	0.1076	0.0591	0.0423	0.10335	0.05463	0.0366

0.104	0.047	0.007	0.1044	0.0564	0.0394	0.10073	0.0537	0.03545
0.095	0.052	0.036	0.1024	0.0544	0.039	0.10168	0.05348	0.03503
0.101	0.011	0.007	0.0986	0.0533	0.0376	0.10269	0.05287	0.03438
0.103	0.048	0.032	0.1071	0.0569	0.0432	0.10255	0.0544	0.03559
0.096	0.006	0.007	0.1037	0.0538	0.0366	0.10125	0.05291	0.03475
0.088	0.043	0.034	0.102	0.0552	0.0375	0.102	0.05385	0.03596
0.109	0.053	0.068	0.1068	0.0572	0.0437	0.10302	0.05422	0.03543
0.096	0.055	0.012	0.1038	0.0548	0.0362	0.10304	0.05433	0.03556
0.09	0.049	0.039	0.1023	0.0538	0.0365	0.10007	0.05281	0.03424
0.083	0.037	0.024	0.1003	0.0524	0.0329	0.10148	0.05356	0.03501
0.093	0.011	0.01	0.1016	0.0528	***0.0103***	0.10035	0.05333	0.03432
0.102	0.061	0.012	0.1032	0.0558	0.0425	0.10184	0.05369	***0.03803***
0.097	0.008	0.004	0.0999	0.0542	0.0387	0.10168	0.05385	0.03488
0.092	0.051	0.031	0.1035	0.0558	0.0378	0.10271	0.05381	0.03539
0.114	0.054	0.008	0.1017	0.0542	0.0378	0.1025	0.054	0.03529
0.108	0.011	0.015	0.1071	0.0571	0.0448	0.10062	0.05379	0.03569
0.108	0.005	0.007	0.1005	0.053	0.0403	0.10269	0.0537	0.03506
0.105	0.048	0.031	0.1038	0.0547	0.036	0.10237	0.05392	0.03533
0.103	0.054	0.036	0.1034	***0.0648***	0.0399	0.10113	0.05432	0.03568
0.102	0.038	0.026	0.1012	0.0516	0.0363	0.10213	0.05291	0.03458
0.112	0.02	0.035	0.1009	0.0543	0.0386	0.101	0.05405	0.03514
0.102	0.053	0.035	0.1062	0.0584	0.0387	0.10109	0.05298	0.03421
0.099	0.049	0.063	0.1058	0.0538	0.0345	0.10328	0.05424	0.03499
0.11	0.067	0.05	0.1068	0.0585	0.0393	0.10168	0.05388	0.0349
0.09914	**0.03648**	**0.02548**	**0.1036**	**0.05522**	**0.036394**	**0.101893**	**0.05366**	**0.035278**

0001	0101	0201	1001	1101	1201	2001	2101	2201
0.004	0.006	0.032	0.0061	0.0043	0.0016	0.00565	0.0025	0.00182
0.006	0.003	0.005	0.0062	0.0025	0.065	0.00603	0.00218	0.00161
0.007	0.009	0.036	0.0065	0.0026	0.0019	0.00624	0.00247	0.0016
0.008	0.007	0.01	0.0061	0.0041	0.0372	0.00589	0.00224	0.00143
0.008	0.005	0.012	0.0056	0.0038	0.0009	0.00634	0.00234	0.00182
0.007	0	0.035	0.0058	0.0018	0.0014	0.0059	0.00239	0.00155
0.005	0.004	0.007	0.0061	0.0021	0.0021	0.00567	0.00216	0.00138
0.006	0.003	0.005	0.0067	0.0025	0.0374	0.00628	0.00225	0.00184
0.011	0.003	0.005	0.0053	0.0021	0.0011	0.00593	0.00241	0.00201
0.01	0.004	0.012	0.0064	0.0026	0.0066	0.00634	0.00235	0.00183
0.109	0.056	0.006	0.0058	0.0024	0.0017	0.00577	0.00223	0.00146
0.008	0.058	0.038	0.0061	0.0036	0.0016	0.00607	0.00235	0.0018
0.003	0.053	0.003	0.0058	0.004	0.002	0.00574	0.00251	0.00164
0.007	0.002	0.037	0.0062	0.0026	0.0355	0.00593	0.00229	0.00144
0.003	0.001	0.032	0.0062	0.0026	0.0412	0.00568	0.00223	0.0013
0.01	0.003	0.009	0.0081	0.0038	0.0098	0.00655	0.00268	0.00217
0.008	0.003	0.007	0.0057	0.0023	0.0022	0.00569	0.00224	0.00173
0.007	0.006	0.036	0.0057	0.0053	0.0346	0.00571	0.00216	0.00156
0.008	0.007	0.037	0.006	0.0023	0.041	0.00595	0.00239	0.00184
0.014	0.002	0.006	0.0075	0.0027	0.0071	0.00623	0.00227	0.00179
0.005	0.001	0.038	0.0063	0.0027	0.0341	0.00608	0.0024	0.0015
0.009	0.001	0.01	0.0068	0.0021	0.0389	0.00555	0.00199	0.00144
0.006	0	0.005	0.0055	0.0024	0.0364	0.00548	0.00219	0.00183

0.009	0.006	0.004	0.0065	0.0027	0.0088	0.00636	0.00243	0.00152
0.004	0.002	0.003	0.0056	0.0024	0.0016	0.0058	0.0022	0.00169
0.005	***0.069***	0.01	0.005	0.0012	0.0013	0.0058	0.00214	0.00165
0.006	0.004	0.007	0.007	0.0035	0.0025	0.00601	0.00239	0.00143
0.001	0	0.009	0.005	0.0018	0.0426	0.00568	0.00221	0.00134
0.005	0.002	0.007	0.0061	0.0027	0.0349	0.00557	0.00219	0.00146
0.004	0.002	0.001	0.0069	0.0026	0.0027	0.00591	0.00213	0.00149
0.005	0	0.003	0.0048	0.0015	0.0366	0.00574	0.00213	0.00136
0.006	0.004	0.002	0.006	0.0031	0.0375	0.00611	0.00234	0.00176
0.005	0.002	0.036	0.006	0.0017	0.0397	0.0057	0.00214	0.00133
0.008	0.003	0.009	0.0058	0.0023	0.0386	0.00597	0.00226	0.00177
0.008	0.001	0.005	0.0072	0.0029	0.039	0.00577	0.00218	0.00147
0.003	0.035	0.003	0.0057	0.002	0.0012	0.006	0.00215	0.00174
0.007	0.002	0.01	0.0066	0.0023	0.0016	0.00619	0.00244	0.00174
0.009	0.006	0.006	0.0059	0.0048	0.0395	0.00599	0.00217	***0.04572***
0.007	0.002	0.004	0.0056	0.0012	0.0013	0.00616	0.00222	0.00179
0.007	0.005	0.006	0.0052	0.0019	0.0378	0.00597	0.00231	0.0018
0.004	0.004	0.002	0.0059	0.002	0.0378	0.006	0.00238	0.00191
0.01	0.006	0.015	0.0068	0.0026	0.0023	0.006	0.0023	0.00185
0.006	0.002	0.004	0.0065	0.0024	0.0018	0.00551	0.00235	0.00141
0.001	0.048	0.001	0.0052	0.0033	0.0382	0.00622	0.00236	0.00153
0.005	0.001	0.008	0.0048	0.0021	0.0016	0.00589	0.00229	0.00131
0.102	0.001	0.023	0.0053	0.0021	0.0399	0.0057	0.00229	0.00176
0.011	0.004	0.007	0.0071	0.0018	0.0016	0.00595	0.00202	0.00166
0.102	0.002	0.008	0.0065	0.0031	0.035	0.00606	0.00249	0.00194
0.007	0.004	0.007	0.0061	0.0047	0.007	0.00603	0.0023	0.00183
0.006	0.006	0.011	0.006	0.0023	0.0417	0.00581	0.00222	0.00145
0.01244	**0.0092**	**0.01268**	**0.006072**	**0.002684**	**0.020308**	**0.005932**	**0.002285**	**0.002522**

0002	0102	0202	1002	1102	1202	2002	2102	2202
0.004	0.006	0.007	0.0059	0.003	0.0016	0.0055	0.00246	0.0015
0.006	0.003	0.005	0.0061	0.0023	0.0012	0.0059	0.00211	0.00127
0.007	0.009	0.004	0.0064	0.0025	0.0014	0.0061	0.00237	0.00144
0.007	0.007	0.01	0.0057	0.0027	0.002	0.0057	0.00217	0.00132
0.008	0.005	0.006	0.0056	0.0038	0.0008	0.0062	0.00226	0.00132
0.007	0	0.002	0.0055	0.0016	0.001	0.0058	0.00226	0.00139
0.005	0.003	0.003	0.006	0.002	0.0016	0.0055	0.00206	0.00119
0.004	0.003	0.002	0.0064	0.0025	0.0011	0.0062	***0.00386***	0.00139
0.011	0.003	0.005	0.0052	0.002	0.001	0.0058	0.0023	0.0015
0.01	0.004	0.002	0.0063	0.0026	0.0013	0.0062	0.00227	***0.00379***
0.007	***0.056***	0.004	0.0058	0.0047	0.0014	0.0056	0.0021	0.00132
0.008	0.005	0.001	0.0061	0.0036	0.0013	0.0059	0.0022	0.00126
0.003	0.001	0.001	0.0057	0.004	0.0016	0.0056	0.00242	0.00152
0.007	0.005	0.009	0.0061	0.0026	0.0014	0.0058	0.00226	0.00135
0.003	0.001	0.002	0.0061	0.0026	0.0017	0.0055	0.00214	0.00117
0.01	0.007	0.003	***0.008***	0.0038	0.0022	0.0064	0.00258	0.00155
0.008	0.003	0.003	0.0055	0.004	0.0015	0.0056	0.00216	0.0013
0.007	0.007	0.01	0.0055	0.0024	0.0012	0.0056	0.0021	0.0011
0.008	***0.051***	0.035	0.006	0.0023	0.0038	0.0058	0.00229	0.00143
0.014	0.002	0.03	0.0073	0.0027	***0.0046***	0.0061	0.00218	0.00136

0.005	0.001	***0.038***	0.0063	0.0027	0.0016	0.006	0.00231	0.00137
0.009	0.001	0.002	0.0067	0.0021	0.0013	0.0055	0.00194	0.00114
0.006	0	0.001	0.0055	0.0023	0.0015	0.0054	0.00213	0.00134
0.008	0.005	0.004	0.0062	0.0025	***0.004***	0.0062	0.00238	0.00138
0.004	0.002	0.001	0.0054	0.0023	0.0015	0.0056	0.00208	0.00121
0.005	0.001	0	0.005	0.0013	0.0005	0.0057	0.00208	0.00118
0.006	0.003	0.003	0.0068	0.0035	0.002	0.0059	0.00229	0.00129
0.001	0	0.009	0.0048	0.0017	0.0012	0.0055	0.0021	0.00124
0.005	0.002	0.002	0.0059	0.0024	0.0014	0.0055	0.00213	0.00134
0.003	0.001	0.001	0.0064	0.0043	0.002	0.0058	0.00205	0.00135
0.004	0	0	0.0046	0.0035	0.0007	0.0055	0.00201	0.00127
0.006	0.008	0.002	0.006	0.0031	0.0011	0.006	0.00223	0.00128
0.005	0.002	0.035	0.006	0.0017	0.0013	0.0056	0.00207	0.00124
0.006	0.005	0.009	0.0055	0.0023	0.0012	0.0058	0.00215	0.0013
0.008	0.001	0.002	0.0069	0.0026	0.0019	0.0056	0.00209	0.00129
0.003	0.035	0.001	0.0055	0.0019	0.0011	0.0059	0.00206	0.00125
0.007	0.002	0.01	0.0064	0.0023	0.0018	0.006	0.00235	0.00137
0.009	0.006	0.004	0.0059	0.0048	0.0022	0.0058	0.00212	0.00135
0.007	0.002	0.003	0.0055	0.0012	0.0009	0.006	0.0021	0.00134
0.007	0.005	0.031	0.0052	0.0019	0.0015	0.0059	0.00227	0.00137
0.003	0.007	0.001	0.0058	0.0019	0.0013	0.0058	0.00226	0.00142
0.009	0.006	0.004	0.0066	0.0026	0.0021	0.0058	0.00221	0.00139
0.006	0.002	0.002	0.0065	0.0025	0.0016	0.0053	0.00224	0.00128
0.001	0	0.005	0.0052	0.0021	0.0013	0.0061	0.00226	0.00143
0.005	0.001	0.003	0.0047	0.0045	0.0008	0.0058	0.0022	0.00125
0.002	0.001	0.001	0.0053	0.0021	0.0012	0.0055	0.00213	0.00128
0.01	0.008	0.003	0.007	0.0018	0.0012	0.0058	0.00195	0.00118
0.007	0.002	0.001	0.006	0.0027	0.002	0.0059	0.00232	0.00143
0.007	0.004	0.004	0.0058	0.0024	0.0015	0.0059	0.00223	0.00132
0.006	0.006	0.002	0.006	0.0023	0.0012	0.0057	0.00215	0.00136
0.00628	**0.006**	**0.00656**	**0.005932**	**0.00266**	**0.0016**	**0.0058**	**0.002229**	**0.001374**

0010	0110	0210	1010	1110	1210	2010	2110	2210
0.897	0.483	0.302	0.9316	0.6188	0.5053	1	0.61157	0.57
0.562	0.377	0.243	0.6206	0.5032	0.4519	1	0.61155	0.56626
0.55	0.385	0.241	0.9383	0.5881	0.5014	1	0.60969	0.56993
0.467	0.277	0.174	0.9332	0.5807	0.5054	1	0.61023	0.53401
0.548	0.29	0.206	0.617	0.5855	0.518	1	0.61002	0.57515
0.873	0.29	0.196	0.623	0.4921	0.4779	1	0.61096	0.55798
0.867	0.406	0.203	0.939	0.5507	0.4315	1	0.60974	0.5552
0.798	0.308	0.178	0.9349	0.5365	0.4476	1	0.58952	0.55227
0.456	0.171	0.12	0.9354	0.541	0.4378	1	0.60914	0.54498
0.822	0.317	0.187	0.9361	0.5892	0.4765	0.93688	0.61232	0.54867
0.583	0.333	0.212	0.6161	0.5822	0.4895	1	0.63087	0.5671
0.531	0.273	0.174	0.614	0.5563	0.4958	1	0.61046	0.56362
0.939	0.31	0.265	0.6188	0.58	0.5014	0.61624	0.61003	0.56404
0.934	0.517	0.215	1	0.5673	0.4949	0.93688	***0.57067***	0.55349
0.782	0.301	0.203	0.6145	0.5379	0.4428	1	0.61129	0.57599
0.479	0.35	0.2	0.6186	0.5505	0.4436	1	0.59032	0.54743
0.859	0.315	0.162	1	0.6027	0.4686	1	0.61057	0.56864

0.503	0.394	0.244	0.6109	0.5751	0.4604	1	0.60747	0.54746
0.554	0.357	0.211	0.9376	0.5852	0.4688	0.61653	0.61058	0.54451
0.484	0.268	0.196	0.6159	0.5492	0.4431	1	0.5926	0.56559
0.376	0.198	0.15	0.612	0.5081	0.4351	0.93427	0.58716	0.54212
0.476	0.316	0.221	0.933	0.5342	0.3898	1	0.61007	0.56413
0.55	0.264	0.15	0.9349	0.596	0.5109	1	0.6091	0.56676
0.89	0.364	0.226	0.6144	0.5569	0.4926	1	0.60896	0.56315
0.794	0.294	0.327	0.943	0.6495	0.5283	1	0.60853	0.56221
0.876	0.449	0.329	0.9357	0.6586	0.5482	1	0.60979	0.54513
0.485	0.223	0.157	0.9356	0.5482	0.4499	0.93559	0.60875	0.57106
0.566	0.404	0.168	0.9329	0.5749	0.4605	0.93548	0.60971	0.55209
0.483	0.315	0.231	0.9381	0.5591	0.4877	1	0.58861	0.54511
0.614	0.46	0.233	0.6159	0.49	0.417	0.61448	0.58862	0.55636
0.874	0.267	0.193	0.9376	0.4641	0.43	0.93537	0.59212	0.55354
0.562	0.34	0.197	0.9323	0.5675	0.4633	0.9349	0.60678	0.5525
0.821	0.412	0.178	0.6156	0.513	0.4063	0.93693	0.60761	0.54604
0.558	0.36	0.242	0.9263	0.5884	0.4579	1	0.6114	0.57036
0.852	0.449	0.258	0.9401	0.5563	0.417	1	0.62629	0.56
0.901	0.445	0.301	0.932	0.6022	0.4878	1	0.60993	0.55251
0.587	0.249	0.176	0.9336	0.5666	0.4902	1	0.60824	0.5525
0.884	0.411	0.29	0.9396	0.5921	0.4989	1	0.63183	0.57922
0.593	0.241	0.173	0.9367	0.5768	0.5143	1	0.60845	0.54586
0.529	0.354	0.213	0.6218	0.5354	0.4522	1	0.60952	0.58166
0.472	0.396	0.235	0.9357	0.5134	0.4584	1	0.61137	0.5659
0.737	0.246	0.323	1	0.6273	0.511	1	0.60927	0.57515
0.863	0.324	0.21	1	0.5678	0.4656	1	0.61376	0.58429
0.326	0.225	0.178	0.611	0.5068	0.4103	1	0.60672	0.5519
0.543	0.35	0.222	1	0.6013	0.5027	1	0.63043	0.58329
0.55	0.274	0.169	0.9344	0.5896	0.4911	0.9366	0.61171	0.54876
0.422	0.278	0.185	0.9398	0.6798	0.4715	0.61629	0.60853	0.54355
0.609	0.349	0.338	0.9379	0.5328	0.4616	0.61572	0.60881	0.55384
0.602	0.315	0.189	0.6229	0.5861	0.521	1	0.60865	0.55757
0.82	0.463	0.24	1	0.6181	0.4789	1	0.61088	0.54866
0.65406	**0.33514**	**0.21668**	**0.834966**	**0.566662**	**0.471444**	**0.950043**	**0.607824**	**0.558951**

0011	0111	0211	1011	1111	1211	2011	2111	2211
0.316	0.181	0.141	0.2716	0.2077	0.1834	0.2797	0.22002	0.19423
0.27	0.18	0.131	0.2798	0.1972	0.1881	0.2845	0.21589	0.2001
0.355	0.203	0.162	0.28	0.221	0.1911	0.28059	0.21038	0.19058
0.246	0.144	0.099	0.2925	0.2268	0.1955	0.28256	0.21382	0.19509
0.254	0.163	0.117	0.2847	0.2169	0.1837	0.28442	0.21373	0.19826
0.261	0.141	0.109	0.2886	0.2125	0.1783	0.28325	0.21371	0.19517
0.313	0.174	0.122	0.262	0.2008	0.1678	0.28254	0.22099	0.19806
0.277	0.161	0.15	0.263	0.2081	0.1771	0.28253	0.213	0.19347
0.22	0.101	0.091	0.2878	0.2028	0.1806	0.28373	0.21531	0.19555
0.266	0.189	0.111	0.2864	0.2078	0.1863	0.28344	0.21328	0.1958
0.325	0.202	0.139	0.2838	0.2094	0.1875	0.2864	0.2156	0.19939
0.233	0.131	0.12	0.2854	0.2194	0.1948	0.28337	0.22247	0.20181
0.277	0.206	0.182	0.2917	***0.2477***	0.2076	0.28529	0.2148	0.1991
0.321	0.196	0.118	0.2834	0.2282	0.1899	0.28377	0.21332	0.1954

0.241	0.157	0.11	0.2823	0.1994	0.181	0.28193	0.21956	0.20431
0.309	0.156	0.1	0.2664	0.1929	0.1784	0.2837	0.21965	0.19886
0.252	0.165	0.103	0.2907	0.226	0.1977	0.2816	0.2198	0.19794
0.293	0.21	0.244	0.2837	0.2247	0.1937	0.28038	0.21884	0.20003
0.252	0.182	0.095	0.285	0.2162	0.19	0.28386	0.21375	0.19838
0.261	0.152	0.106	0.292	0.2277	0.1996	0.28437	0.22174	0.20286
0.163	0.129	0.112	0.2564	0.2071	0.1738	0.28127	0.21271	0.19072
0.221	0.15	0.163	0.2618	0.1799	0.159	0.28249	0.21243	0.20062
0.228	0.121	0.115	0.2811	0.2077	0.1754	0.28329	0.22259	0.20739
0.32	0.146	0.222	0.2847	0.2062	0.1857	0.28275	0.21284	0.20207
0.233	0.141	0.167	0.2869	0.2152	0.1865	0.2817	0.21835	0.19711
0.39	0.185	0.183	0.2843	0.2226	0.1911	0.28527	0.22175	0.20016
0.212	0.131	0.086	0.262	0.2069	0.1847	0.27966	0.21063	0.1962
0.259	0.15	0.097	0.2869	0.224	0.1957	0.28272	0.22156	0.19846
0.208	0.135	0.102	0.2904	0.2164	0.1963	0.28547	0.21374	0.19516
0.393	0.228	0.135	0.2585	0.1932	0.1743	0.28279	0.20391	0.18831
0.389	0.147	0.088	0.2666	0.2023	0.1702	0.28255	0.21313	0.19463
0.279	0.18	0.126	0.2614	0.2098	0.1861	0.28129	0.21215	0.19818
0.285	0.153	0.111	0.2547	0.1935	0.1724	0.28288	0.21326	0.19393
0.246	0.129	0.124	0.2636	0.2007	0.1761	0.28373	0.21449	0.19255
0.272	0.199	0.116	0.2863	0.2005	0.1624	0.28214	0.21177	0.19238
0.312	0.199	0.151	0.2879	0.2262	0.2021	0.28175	0.22141	0.20396
0.208	0.122	0.097	0.277	0.2036	0.1824	0.28325	0.21352	0.19445
0.215	0.167	0.111	0.2874	0.2126	0.1839	0.2842	0.2141	0.19977
0.255	0.152	0.124	0.2892	0.2263	0.2045	0.28509	0.21505	0.1958
0.309	0.177	0.105	0.2857	0.2155	0.1846	0.28493	0.2215	0.20164
0.253	0.171	0.158	0.2849	0.2046	0.1893	0.28096	0.22039	0.20196
0.265	0.148	0.126	0.2899	0.2116	0.1914	0.28281	0.22166	0.20498
0.267	0.176	0.158	0.2814	0.212	0.1831	0.28645	0.22265	0.20062
0.213	0.126	0.098	0.2568	0.205	0.1766	0.28278	0.21186	0.19394
0.263	0.163	0.114	0.2889	0.2146	0.1844	0.28315	0.21614	0.19849
0.239	0.139	0.106	0.256	0.2032	0.1734	0.28303	0.21261	0.19791
0.243	0.126	0.094	0.2623	0.2098	0.1977	0.2858	0.21484	0.19517
0.238	0.177	***0.328***	0.2877	0.2211	0.1997	0.28406	0.21344	0.20122
0.251	0.165	0.122	0.2889	0.2172	0.1948	0.28587	0.21482	0.19966
0.271	0.184	0.147	0.2645	0.2002	0.1784	0.28389	0.21513	0.19915
0.26884	**0.1622**	**0.13072**	**0.278298**	**0.211254**	**0.185362**	**0.283199**	**0.215882**	**0.19782**

0012	0112	0212	1012	1112	1212	2012	2112	2212
0.159	0.102	0.076	0.1463	0.1033	0.0897	0.15115	0.1084	0.10132
0.175	0.108	0.083	0.1475	0.1066	0.094	0.15219	0.11474	0.1024
0.194	0.12	0.099	0.1755	0.1192	0.1063	0.15087	0.10769	0.09881
0.145	0.096	0.064	0.1809	0.1107	0.0995	0.14989	0.10863	0.1003
0.14	0.092	0.069	0.1511	0.1128	0.0984	0.1529	0.11414	0.10368
0.13	0.08	0.091	0.1796	0.1062	0.0935	0.15157	0.10902	0.09806
0.147	0.089	0.055	0.1761	0.1135	0.0959	0.1525	0.11278	0.10439
0.142	0.116	0.089	0.1546	0.1055	0.0988	0.15118	0.1088	0.09945
0.144	0.064	0.058	0.1572	0.1081	0.098	0.15418	0.11434	0.10206
0.152	0.092	0.074	0.1522	0.1125	0.0974	0.151	0.10939	0.09812
0.189	0.102	0.074	0.1701	0.1125	0.0987	0.15356	0.10975	0.09948

0.137	0.076	0.07	0.1513	0.1115	0.0973	0.15209	0.1094	0.10341
0.148	0.107	0.158	0.1768	0.1207	0.1045	0.15365	0.11381	0.10189
0.161	0.114	0.067	0.1512	0.1096	0.0919	0.15159	0.10977	0.1008
0.137	0.078	0.057	0.1492	0.1068	0.0956	0.1511	0.10755	0.09827
0.159	0.08	0.059	0.158	0.108	0.0912	0.15209	0.11367	0.10325
0.157	0.094	0.074	0.1537	0.1116	0.1014	0.15036	0.10779	0.10042
0.199	0.124	**0.244**	0.171	0.115	0.1027	***0.17309***	0.11262	0.10434
0.14	0.089	0.055	0.1494	0.1038	0.0913	0.15287	0.11449	0.1037
0.142	0.096	0.065	0.1785	0.1177	0.1045	0.15309	0.11476	0.10345
0.105	0.075	0.097	0.1517	0.0996	0.0875	0.15193	0.10944	0.09989
0.139	0.098	0.116	0.1493	0.0998	0.0863	0.15039	0.10798	0.09894
0.103	0.074	0.075	0.152	0.104	0.0923	0.1514	0.11374	0.1028
0.138	0.072	0.106	0.1517	0.1134	0.0996	0.15129	0.11425	0.10547
0.1	0.092	0.066	0.1524	0.1041	0.0923	0.15156	0.11005	0.10178
0.165	0.109	0.108	0.1773	0.1126	0.1007	0.15228	0.10978	0.10056
0.091	0.065	0.052	0.1535	0.1102	0.0961	0.15033	0.10818	0.09892
0.157	0.075	0.055	0.1721	0.1132	0.1041	0.15142	0.11214	0.10583
0.127	0.085	0.06	0.1814	0.1174	0.0998	0.15322	0.10943	0.10152
0.17	0.099	0.064	0.1401	0.1071	0.0925	0.15018	0.10773	0.09738
0.169	0.088	0.07	0.1502	0.108	0.0959	0.15161	0.10803	0.09777
0.18	0.105	0.085	0.1524	0.1103	0.0999	0.1515	0.11327	0.10262
0.156	0.08	0.077	0.1457	0.1037	0.0891	0.1513	0.10947	0.10182
0.142	0.084	0.072	0.1508	0.1041	0.089	0.15188	0.10873	0.0987
0.175	0.084	0.064	0.1538	0.1033	0.0902	0.15154	0.10792	0.10022
0.152	0.088	0.066	0.1792	0.1193	0.1034	0.15057	0.11336	0.10255
0.096	0.08	0.074	0.142	0.104	0.0931	0.15226	0.11302	0.10056
0.148	0.093	0.073	0.1521	0.1169	0.1047	0.15301	0.1149	0.10512
0.143	0.087	0.077	0.1803	0.119	0.1076	***0.17664***	0.11427	0.10395
0.152	0.087	0.063	0.1469	0.1074	0.0958	0.15315	0.11422	0.10326
0.173	0.106	0.086	0.1711	0.1128	0.1028	0.14993	0.11326	0.10157
0.156	0.098	0.072	0.1562	0.1124	0.0963	0.15185	0.11338	0.10369
0.153	0.09	0.088	0.1529	0.1098	0.0986	0.15285	0.10952	0.10076
0.118	0.075	0.053	0.1504	0.1033	0.0924	0.15129	0.10806	0.10062
0.18	0.096	0.067	0.1563	0.113	0.1001	0.15233	0.11007	0.10222
0.14	0.085	0.075	0.1507	0.106	0.0956	0.15197	0.11354	0.1035
0.145	0.083	0.062	0.1543	0.1052	0.0993	0.15209	0.11285	0.10312
0.163	0.101	0.13	0.154	0.112	0.1002	0.15293	0.10799	0.10008
0.144	0.091	0.072	0.1531	0.1117	0.1023	0.15324	0.10983	0.10257
0.151	0.098	0.118	0.1563	0.104	0.0947	0.15258	0.10928	0.10067
0.14856	**0.09124**	**0.08048**	**0.158408**	**0.1097**	**0.097056**	**0.152789**	**0.110985**	**0.101521**

0020	0120	0220	1020	1120	1220	2020	2120	2220
1	0.994	0.85	1	1	1	1	1	1
1	0.988	0.845	1	1	1	1	1	1
1	0.983	0.795	1	1	***0.961***	1	1	1
1	0.997	0.852	1	1	1	1	1	1
1	0.994	0.735	1	1	***0.9479***	1	1	1
1	0.923	0.767	1	1	1	1	1	1
1	***0.847***	0.872	1	1	1	1	1	1
1	0.999	0.857	1	1	0.9927	1	1	1

1	0.998	0.77	1	1	1	1	1	1
1	0.99	0.823	1	1	1	1	1	1
1	1	0.801	1	1	1	1	1	1
1	0.992	0.784	1	1	1	1	1	1
1	0.961	0.825	1	1	0.9952	1	1	1
1	0.997	0.928	1	1	0.9982	1	1	1
1	0.959	0.78	1	1	1	1	1	1
1	0.913	0.837	1	1	0.9975	1	1	1
1	0.962	0.817	1	1	1	1	1	1
1	0.989	0.927	1	1	1	1	1	1
1	0.988	0.826	1	1	0.9982	1	1	1
1	0.992	0.877	1	1	0.9975	1	1	1
1	0.993	0.755	1	1	0.9983	1	1	1
1	0.99	0.934	1	1	0.9974	1	1	1
1	0.883	0.74	1	1	1	1	1	1
1	0.986	0.833	1	1	0.9982	1	1	1
1	0.927	0.897	1	1	1	1	1	1
1	0.995	0.955	1	1	1	1	1	1
1	0.995	0.792	1	1	0.9979	1	1	1
1	0.975	0.831	1	1	1	1	1	1
1	0.989	0.866	1	1	0.9978	1	1	1
1	0.995	0.858	1	1	0.9984	1	1	1
1	0.988	0.798	1	1	1	1	1	1
1	0.977	0.837	1	1	0.9978	1	1	1
1	0.957	0.821	1	1	1	1	1	1
1	0.995	0.819	1	1	1	1	1	1
1	0.955	0.912	1	1	0.9971	1	1	1
1	0.895	0.842	1	1	1	1	1	1
1	0.999	0.728	1	1	1	1	1	1
1	0.992	0.864	1	1	1	1	1	1
1	0.989	0.782	1	1	1	1	1	1
1	0.985	0.86	1	1	1	1	1	1
1	0.976	0.867	1	1	1	1	1	1
1	0.986	0.822	1	1	1	1	1	1
1	0.997	0.83	1	1	0.995	1	1	1
1	***0.797***	0.789	1	1	0.9931	1	1	1
1	0.991	0.842	1	1	0.9978	1	1	1
1	0.998	0.871	1	1	0.9987	1	1	1
1	0.987	0.934	1	1	0.9976	1	1	1
1	0.991	0.863	1	1	0.9982	1	1	1
1	0.988	0.955	1	1	0.9975	1	1	1
1	0.993	0.876	1	1	0.9975	1	1	1
1	**0.9726**	**0.83882**	**1**	**1**	**0.99693**	**1**	**1**	**1**

0021	**0121**	**0221**	**1021**	**1121**	**1221**	**2021**	**2121**	**2221**
1	0.855	0.615	1	0.7732	0.6756	1	0.67139	0.63131
1	0.668	0.551	1	0.6725	0.6316	1	0.67445	0.63524
1	0.655	0.52	1	0.7201	0.5914	1	0.67187	0.59326
1	0.657	0.555	1	0.6779	0.6208	1	0.67396	0.61824
1	0.591	0.514	1	0.6765	0.5975	1	0.67351	0.62734

0.977	0.596	0.499	1	0.6726	0.5971	1	0.67354	0.62674
1	0.632	0.567	1	0.6805	0.6348	1	0.67201	0.63203
1	0.724	0.56	1	0.6702	0.6244	1	0.67226	0.62519
1	0.599	0.47	1	0.6766	0.6071	1	0.67417	0.63283
1	0.627	0.504	1	0.6776	0.6152	1	0.67453	0.63383
1	0.67	0.531	1	0.7678	0.6155	1	0.67416	0.63388
1	0.65	0.725	1	0.6676	0.5889	1	0.67049	0.63084
1	0.688	0.532	1	0.6725	0.6165	1	0.67292	0.62369
1	0.782	0.926	1	0.6695	0.6255	1	0.67174	0.63107
0.933	0.69	0.529	1	0.6698	0.598	1	0.67184	0.63164
1	0.691	0.509	1	0.6788	0.5996	1	0.67459	0.63181
1	0.741	0.771	1	0.6789	0.6263	1	0.67249	0.63267
1	0.717	0.62	1	0.6729	0.613	1	0.67178	0.63131
1	0.773	0.577	1	0.7527	0.6375	1	0.67519	0.63457
1	0.697	0.539	1	0.7779	0.6301	1	0.67617	0.63463
1	0.705	0.562	1	0.6683	0.6248	1	0.67155	0.59264
1	0.766	0.667	1	0.6784	0.6123	1	0.67498	0.633
1	0.599	0.666	1	0.6707	0.6195	1	0.67304	0.63122
1	0.784	0.55	1	0.673	0.6296	1	0.67399	0.63385
1	0.673	0.878	1	0.6732	0.6288	1	0.6702	0.63045
1	0.83	0.642	1	0.79	0.6281	1	0.67445	0.59526
0.987	0.587	0.529	1	0.6778	0.6334	1	0.67011	0.60117
1	0.672	0.519	1	0.7202	0.638	1	0.67336	0.63388
1	0.679	0.865	1	0.7561	0.6355	1	0.67106	0.62884
1	0.799	0.576	1	0.6763	0.5929	1	0.67317	0.6108
0.985	0.794	0.797	1	0.6776	0.6245	1	0.67381	0.59575
1	0.744	0.579	1	0.6781	0.6228	1	0.67126	0.63069
0.974	0.752	0.574	1	0.6703	0.6087	1	0.67084	0.6097
1	0.665	0.602	1	0.7221	0.6183	1	0.67348	0.63294
1	0.733	0.634	1	0.6787	0.6308	1	0.67148	0.63056
1	0.684	0.502	1	0.7201	0.6332	1	***0.7923***	0.63263
1	0.622	0.541	1	0.6711	0.5896	1	0.67416	0.63233
0.981	0.735	0.606	1	0.675	0.6365	1	0.67381	0.63426
1	0.698	0.726	1	0.787	0.6646	1	0.67274	0.63275
0.982	0.645	0.555	1	0.6756	0.6324	1	0.67321	0.62767
1	0.763	0.558	1	0.6729	0.6054	1	0.67288	0.60394
1	0.635	0.599	1	0.6666	0.5886	1	0.67321	0.63153
1	0.65	0.546	1	0.6627	0.6211	1	0.67316	0.63497
1	0.585	0.521	1	0.674	0.6017	1	0.67163	0.63036
0.862	0.584	0.487	1	0.6781	0.5961	1	0.67231	0.63237
1	0.667	0.547	1	0.6705	0.6227	1	0.67452	0.63423
0.981	0.708	0.576	1	0.6741	0.6233	1	0.67375	0.63308
1	0.66	0.863	1	0.6777	0.5986	1	0.67313	0.63016
1	0.647	0.566	1	0.6751	0.6375	1	0.67323	0.63457
1	0.762	0.581	1	0.6767	0.6027	1	0.67466	0.63145
0.99324	**0.6906**	**0.60056**	**1**	**0.691482**	**0.618968**	**1**	**0.675371**	**0.626383**

0022	0122	0222	1022	1122	1222	2022	2122	2222
0.668	0.585	0.475	0.6598	0.5299	0.5076	0.6586	0.51445	0.5006
0.657	0.501	0.427	0.6599	0.5097	0.4927	0.6614	0.51745	0.5043

0.66	0.508	0.445	0.6579	0.5102	0.4834	0.6594	0.51113	0.4906
0.648	0.529	0.428	0.6631	0.5365	0.5105	0.6607	0.51503	0.5003
0.646	0.471	0.427	0.6632	0.5166	0.4944	0.6612	0.51722	0.5045
0.664	0.468	0.413	0.6586	0.5094	0.4914	0.6606	0.51489	0.4943
0.642	0.522	0.435	0.6671	0.5199	0.4995	0.6589	0.5143	0.4985
0.642	0.532	0.448	0.6562	0.5117	0.4964	0.6589	0.51376	0.493
0.649	0.462	0.383	0.6656	0.5171	0.5007	0.6616	0.51566	0.5017
0.679	0.482	0.424	0.6653	0.5151	0.5034	0.6614	0.51525	0.502
0.667	0.513	0.439	0.6571	0.5413	0.502	0.6618	0.51822	0.5003
0.64	0.507	0.724	0.6561	0.5085	0.4895	0.6575	0.51366	0.4996
0.654	0.483	0.405	0.6599	0.5148	0.4938	0.6606	0.51565	0.4961
0.626	0.52	***0.924***	0.6545	0.5114	0.4875	0.6586	0.51256	0.4999
0.655	0.475	0.419	0.6568	0.5142	0.487	0.6596	0.51366	0.4935
0.652	0.49	0.415	0.6665	0.5187	0.4971	0.6619	0.51573	0.4954
0.642	0.498	0.809	0.6665	0.5283	0.5081	0.6589	0.5146	0.501
0.654	0.522	0.429	0.6614	0.5348	0.4942	0.6593	0.51152	0.4982
0.675	0.512	0.422	0.6648	0.52	0.504	0.6625	0.51661	0.4969
0.655	0.51	0.422	0.6624	0.5418	0.5046	0.6627	0.51815	0.5032
0.633	0.488	0.435	0.6582	0.5077	0.4883	0.6593	0.51236	0.4924
0.66	0.564	0.479	0.6652	0.5202	0.4947	0.6615	0.51578	0.4983
0.668	0.473	0.637	0.6591	0.5079	0.4944	0.6608	0.51366	0.4946
0.653	0.506	0.422	0.6595	0.5091	0.4865	0.6607	0.51417	0.5
0.663	0.508	0.49	0.6618	0.5136	0.4942	0.6579	0.51177	0.4987
0.841	0.545	0.489	0.6561	0.5389	0.5033	0.6619	0.51516	0.4996
0.63	0.481	0.421	0.6644	0.515	0.501	0.6574	0.5116	0.4909
0.672	0.475	0.416	0.6644	0.5266	0.4981	0.6606	0.51478	0.5015
0.666	0.531	0.649	0.6645	0.5185	0.4998	0.6585	0.51439	0.4982
0.674	0.546	0.459	0.6649	0.5087	0.4945	0.6609	0.51235	0.4924
0.645	0.496	0.798	0.6643	0.5187	0.4979	0.6615	0.51515	0.4954
0.654	0.538	0.444	0.6658	0.5129	0.4949	0.6582	0.51138	0.4969
0.656	0.49	0.436	0.6563	0.5101	0.4874	0.6581	0.51291	0.4923
0.649	0.54	0.448	0.6575	0.5392	0.4895	0.6611	0.51485	0.4936
0.672	0.527	0.433	0.6667	0.5162	0.4952	0.6588	0.51151	0.4978
0.654	0.448	0.39	0.6584	0.5126	0.4923	0.6602	0.51342	0.5005
0.641	0.488	0.433	0.6598	0.5044	0.4889	0.661	0.51461	0.4941
0.65	0.505	0.429	0.6615	0.5233	0.4949	0.6607	0.51685	0.502
0.63	0.512	0.721	0.6575	0.54	0.4989	0.6597	0.51391	0.5004
0.65	0.504	0.431	0.6635	0.5192	0.5007	0.6601	0.5147	0.4954
0.658	0.511	0.411	0.6591	0.5167	0.4908	0.6597	0.51516	0.4925
0.639	0.493	0.484	0.6535	0.5121	0.4927	0.6606	0.51245	0.4991
0.666	0.524	0.449	***0.6482***	0.5046	0.4853	0.6594	0.51605	0.4967
0.661	0.488	0.424	0.6598	0.5129	0.4895	0.6588	0.51261	0.4998
0.646	0.478	0.415	0.6664	0.5209	0.5013	0.6597	0.51453	0.4961
0.662	0.521	0.456	0.6581	0.5131	0.4928	0.6618	0.5157	0.5011
0.675	0.529	0.439	0.6607	0.5092	0.4939	0.6605	0.51502	0.4937
0.66	0.52	0.862	0.6658	0.5072	0.4914	0.6599	0.514	0.4954
0.657	0.536	0.46	0.6632	0.5209	0.5043	0.6607	0.51772	0.5039
0.679	0.53	0.463	0.6632	0.5244	0.4986	0.6611	0.51667	0.497
0.6588	**0.5077**	**0.48872**	**0.661002**	**0.518294**	**0.495676**	**0.6602**	**0.514494**	**0.4977**

Appendix C

The initial analysis was performed on the results of the 3^4-factorial design. Since the results were inconclusive the decision was made to isolate correlation as a factor and treat the analysis as three independent 3^3-factorial designs (see Chapter 4). The results of the initial 3^4-factorial design are listed below.

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square (MS)	F_0 (MS/MSE)	P(F<X)
Main effects					
Sample size (A)	1.8102	2	0.9051	546.85	0.000
Grid size (B)	14.0503	2	7.0251	4244.50	0.000
Correlation (C)	379.8799	2	189.9399	1.1×10^5	0.000
FC-standard (D)	68.7440	2	34.3720	2.1×10^4	0.000
2-factor interactions					
AB	0.1975	4	0.0494	29.83	0.000
AC	1.9795	4	0.4949	299.00	0.000
AD	2.1915	4	0.5479	331.02	0.000
BC	4.8563	4	1.2141	733.53	0.000
BD	1.3485	4	0.3371	203.68	0.000
CD	25.3555	4	6.3389	3829.87	0.000
3-factor interactions					
ABC	0.1313	8	0.0164	9.91	0.000
ABD	0.1670	8	0.0209	12.61	0.000
BCD	9.5914	8	1.1989	724.37	0.000
CDA	2.0456	8	0.2557	154.49	0.000
4-factor interaction					
ABCD	0.2090	16	0.0131	7.89	0.000
Error	6.5691	3969	0.0017 (MSE)		
Total	519.1266	4049			

Table C-1: ***Summary – ANOVA table***

The four-factor interaction, between all the factors, denoted "ABCD" in Table C-1 is added to the error term since it would be hard if not impossible to explain this interaction even if it was significant. Assuming that our assumption is correct that the four-factor interaction is indeed not significant, this approach will increase reliability for the rest of the analysis by increasing the mean square error (MSE). The results of the reduced model (i.e., with the four-factor interaction added to the error term) are shown in table C-2.

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square (MS)	F_0 (MS/MSE)	P(F<X)
Main effects					
Sample size (A)	1.8102	2	0.9051	532.12	0.000
Grid size (B)	14.0503	2	7.0251	4130.19	0.000
Correlation (C)	379.8799	2	189.9399	1.1×10^5	0.000
FC-standard (D)	68.7440	2	34.3720	2.0×10^4	0.000
2-factor interactions					
AB	0.1975	4	0.0494	29.03	0.000
AC	1.9795	4	0.4949	290.95	0.000
AD	2.1915	4	0.5479	322.11	0.000
BC	4.8563	4	1.2141	713.78	0.000
BD	1.3485	4	0.3371	198.20	0.000
CD	25.3555	4	6.3389	3726.72	0.000
3-factor interactions					
ABC	0.1313	8	0.0164	9.65	0.000
ABD	0.1670	8	0.0209	12.27	0.000
BCD	9.5914	8	1.1989	704.87	0.000
CDA	2.0456	8	0.2557	150.33	0.000
4-factor interaction					
ABCD	N/A	N/A	N/A	N/A	
Error	6.7782	3985	0.0017 (MSE)		
Total	519.1266	4049			

Table C-2: ***ANOVA table***

Despite increasing the MSE it is still very small and will result in large F-values (see Table C-2). Note: the increase in MSE is in fact so small that it is not reflected in four decimal places. However, it does affect the F-values somewhat, which is reflected in Table C-2. In order to reduce the model further, one has to first study the interactions and explain them as insignificant. Plotting the interactions will allow for better analysis of significance.

Studying the three-factor interaction between grid size, correlation, and FC-standard reveals that the lowest false alarm probabilities will be achieved with correlation at the low level (i.e., negative correlation), FC-standard at the high level (i.e., maximum allowed conditional false clear probability equal to 0.025), and grid size at the high level (i.e., 225 sub-regions in the estimation grid). Figures C-1, C-2, and C-3 provide a graphical representation of the interaction between grid size, correlation, and FC-standard.

Comparing Figure C-1 with Figure C-2 and C-3 can explain the interaction effect. Figure C-2 and C-3 have essentially the same form. The only significant difference between the two is the mean false alarm probability, which is slightly smaller for larger grid sizes. Otherwise, the interaction between correlation and FC-standard does not change as grid size change from level 1 to level 2. However, in Figure C-1 it is noticeable how much more performance decreases as correlation is changed from level 0 to level 1, and from level 1 to level 2. This is especially noticeable for strict FC-standards. Note that Figure C-1 falsely indicates that the performance, for strict FC-standard, decreases less rapidly when correlation changes from level 1 to level 2. This is merely due to the fact that the maximum false alarm rate (i.e., 1) is reached.

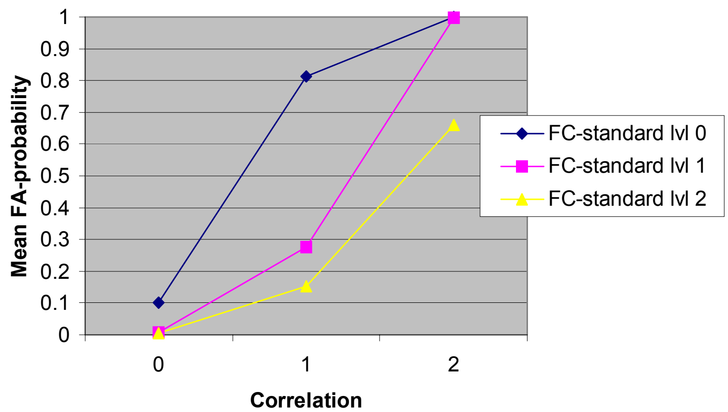

Figure C-0-1: ***Correlation – FC-standard interaction with grid size at the 0-level***

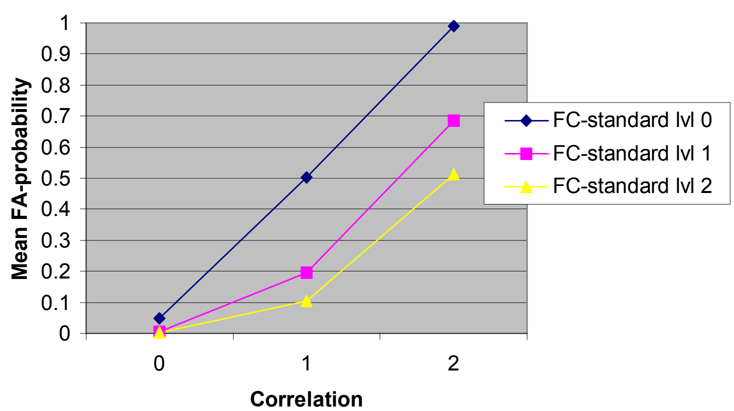

Figure C-0-2: ***Correlation – FC-standard interaction with grid size at the 1-level***

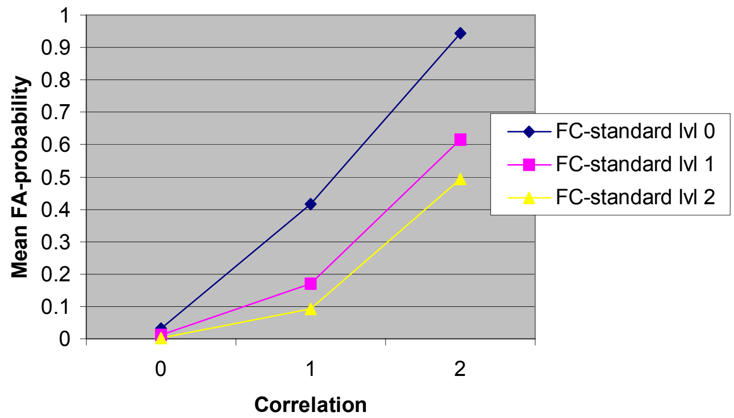

Figure C-0-3: ***Correlation – FC-standard with grid size at the 2-level***

The interaction between correlation, FC-standard, and sample size is shown in Figures C-4, C-5, and C-6. The figures indicate an interaction for strict FC-standards. Performance decreases less rapidly as correlation is changed from the 0 level to the 1 level for smaller sample sizes.

Similar analysis of the remaining three-factor interactions shows that they are not significant. Hence, a reduced model could be run where the interactions classified as insignificant are added to the error term in the same fashion as the four-factor interaction. This would further increase the mean square error and strengthen the analysis of the remaining factors and interactions.

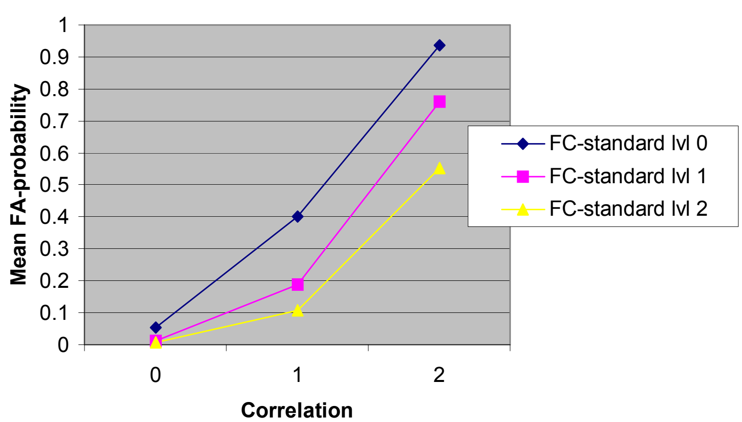

Figure C-0-4: ***Correlation – FC-standard interaction with sample size at the 0-level***

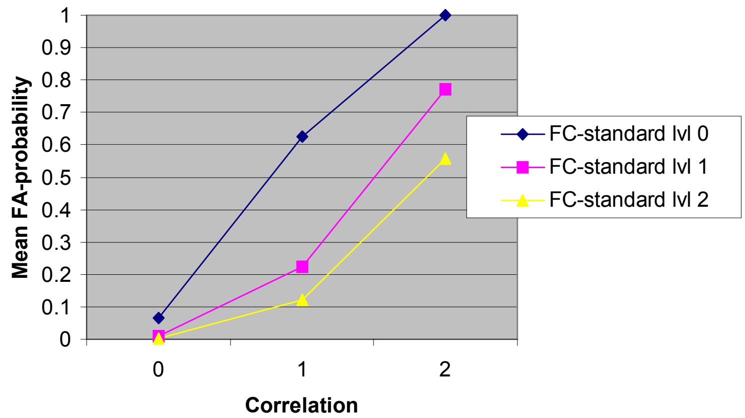

Figure C-0-5: ***Correlation – FC-standard interaction with sample size at the 1-level***

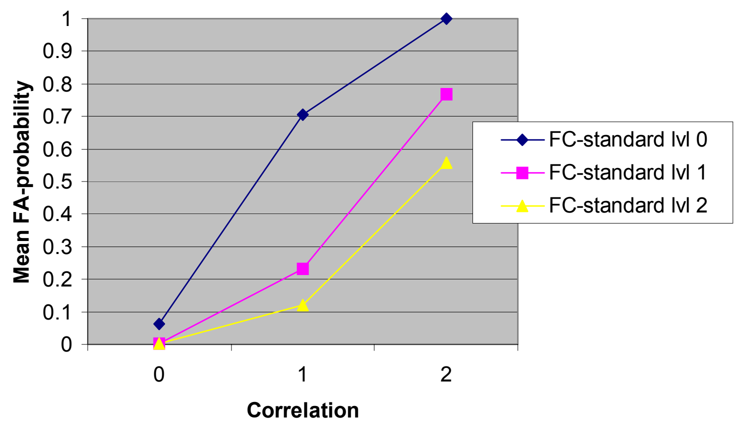

Figure C-0-6: ***Correlation – FC-standard interaction with sample size at the 2-level***

Figure C-7 shows the interaction between grid size and sample size and confirms the fact that there is no significant interaction. This graph also shows that there is essentially no difference between sample sizes of 1,000, 10,000 and 100,000; and very little difference between grid size at the intermediate and high level.

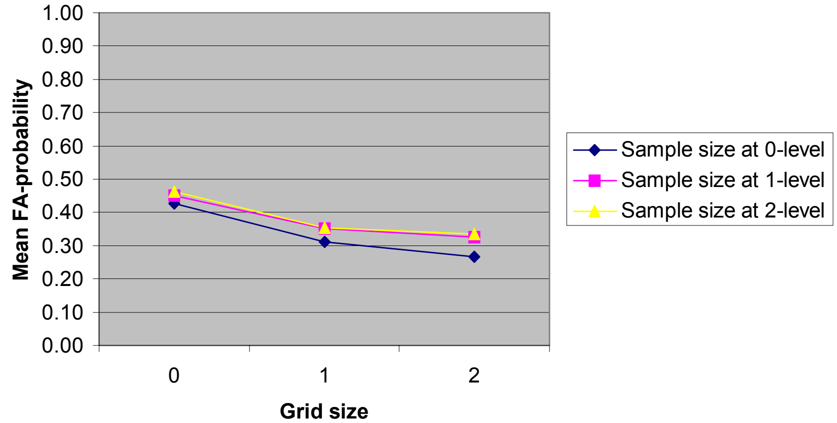

Figure C-0-7: ***Interaction between sample size and grid size***

Table C-3 shows the results of a reduced model where the three-factor interactions between sample size, grid size, and correlation and sample size, grid size, and FC-standard; and the two factor interaction between sample size and grid size has been added to the error term. The MSE is still very small which results in large F-values and inability to make conclusive statements about the significance of main factors and interactions.

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square (MS)	F_0 (MS/MSE)	P(F<X)
Main effects					
Sample size (A)	1.810	2	0.905	498.34	0.000
Grid size (B)	14.050	2	7.025	3868.02	0.000
Correlation (C)	379.880	2	189.940	1.0×10^5	0.000
FC-standard (D)	68.744	2	34.372	1.9×10^4	0.000
2-factor interactions					
AB	N/A	N/A	N/A	N/A	N/A
AC	1.980	4	0.495	272.48	0.000
AD	2.192	4	0.548	301.66	0.000
BC	4.856	4	1.214	668.47	0.000
BD	1.348	4	0.337	185.62	0.000
CD	25.355	4	6.339	3490.17	0.000
3-factor interactions					
ABC	N/A	N/A	N/A	N/A	N/A
ABD	N/A	N/A	N/A	N/A	N/A
BCD	9.591	8	1.199	660.12	0.000
CDA	2.046	8	0.256	140.79	0.000
4-factor interaction					
ABCD	N/A	N/A	N/A	N/A	N/A
Error	7.274	4005	0.002 (MSE)		
Total	519.127	4049			

Table C-3: ***ANOVA table***

Figure C-8 shows that negatively correlated sensor responses generate the lowest mean false alarm probabilities and the fact that there is no significant interaction between sample size and correlation.

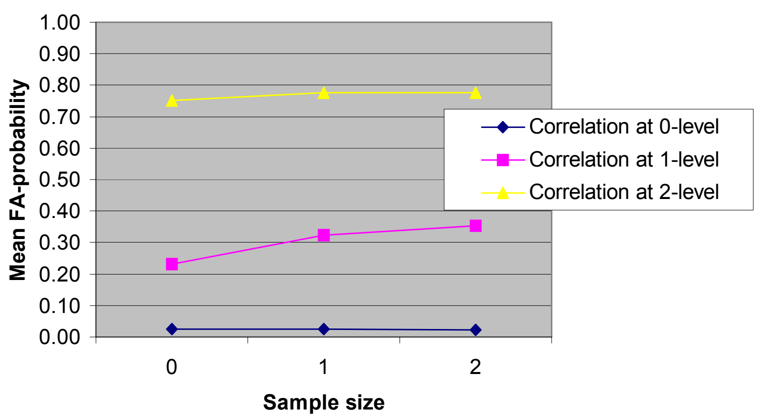

Figure C-0-8: ***Interaction between sample size and correlation***

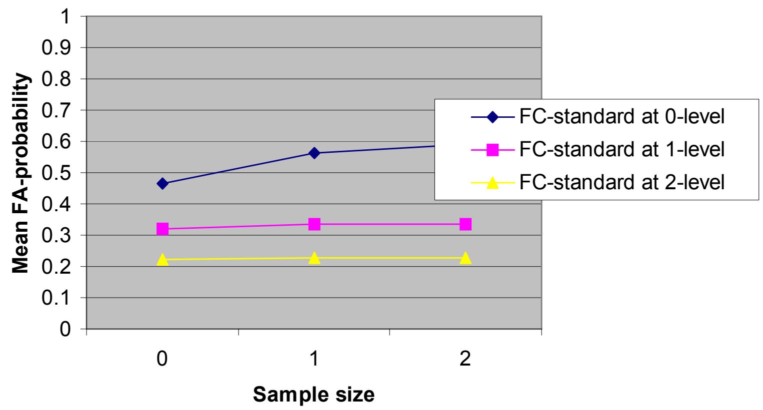

Figure C-0-9: ***Interaction between sample size and FC-standard***

Figure C-9 indicates that the two-way interaction between sample size and FC-standard is non-existing or very small. More experiments with negatively correlated sensor should be performed to confirm the lack of interaction.

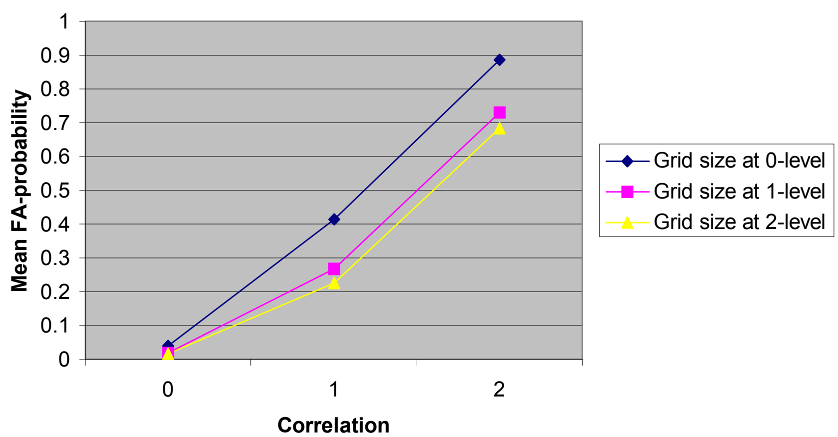

Figure C-0-10: ***Interaction effects between grid size and correlation***

Figure C-10 indicates that the interaction between correlation and grid size is not significant. The graph also shows that grid size has no effect for negatively correlated sensor responses and that there is very little difference between grid sizes at 1-level and 2-level.

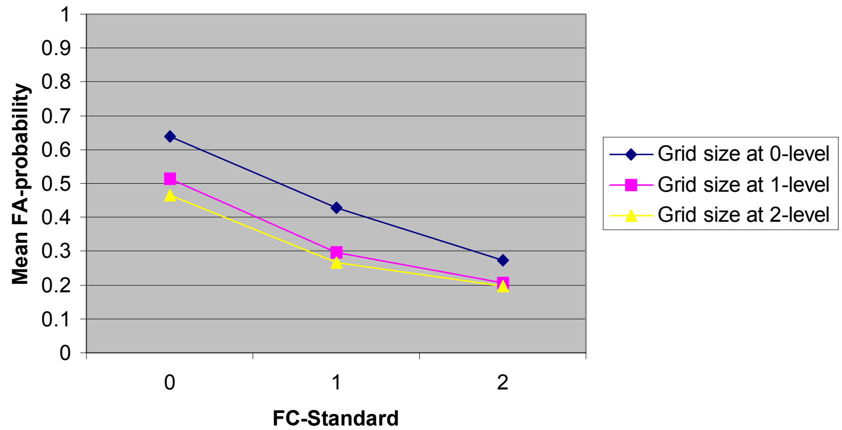

Figure C-0-11: ***Interaction effects between grid size and FC-standard***

Figure C-11 shows that the interaction between grid size and FC-standard is less significant than the interaction between correlation and grid size. Figure C-11 also shows that with the FC-standard at the high level (i.e., 0.025), there is very little or no difference in false alarm probability between the different grid sizes at 1-level and 2-level.

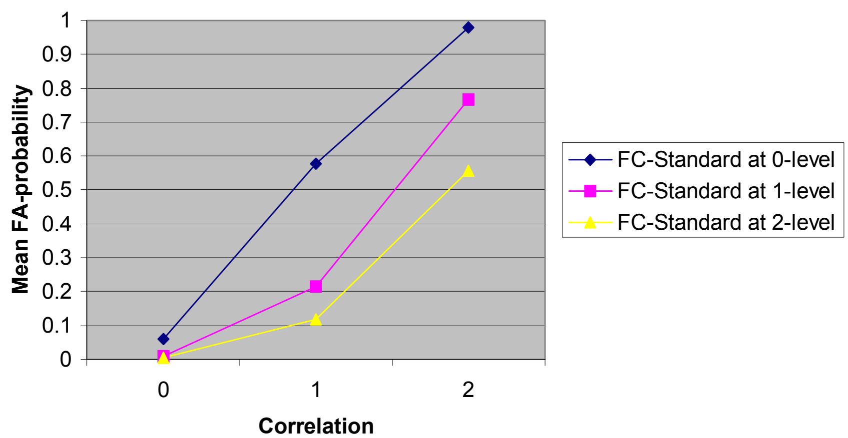

Figure C-0-12: ***Interaction effects between correlation and FC-standard***

Figure C-12 shows the most significant two-factor interaction. The interaction between correlation and FC-standard is more significant then any other two or three-factor interaction. The graph further indicates how much correlation affects the false alarm probabilities. Even though the FC-standard is a quite influential factor (second most significant main effect) its effect is minuscule for negatively correlated sensor responses.

The following models are included to show the difficulty in attempting to increase the MSE. Even if most interaction effects could be explained as insignificant the MSE remains very small which results in large F-values. Hence, it is very hard to make any conclusions about the significance of factors and any interaction effects.

Model three: The "CDA" three-factor interaction has been removed to further reduce the model.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Sample s	2	1.810	1.810	0.905	389.73	0.000
Grid siz	2	14.050	14.050	7.025	3025.02	0.000
Correlat	2	379.880	379.880	189.940	8.2E+04	0.000
FC-stand	2	68.744	68.744	34.372	1.5E+04	0.000
Sample s*Correlat	4	1.980	1.980	0.495	213.10	0.000
Sample s*FC-stand	4	2.192	2.192	0.548	235.92	0.000
Grid siz*Correlat	4	4.856	4.856	1.214	522.78	0.000
Grid siz*FC-stand	4	1.348	1.348	0.337	145.16	0.000
Correlat*FC-stand	4	25.355	25.355	6.339	2729.52	0.000
Grid siz*Correlat* FC-stand	8	9.591	9.591	1.199	516.26	0.000
Error	4013	9.320	9.320	0.002		
Total	4049	519.127				

Model four: The remaining two-factor interactions involving sample size have been removed.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Sample s	2	1.810	1.810	0.905	269.77	0.000
Grid siz	2	14.050	14.050	7.025	2093.91	0.000
Correlat	2	379.880	379.880	189.940	5.7E+04	0.000
FC-stand	2	68.744	68.744	34.372	1.0E+04	0.000
Grid siz*Correlat	4	4.856	4.856	1.214	361.87	0.000
Grid siz*FC-stand	4	1.348	1.348	0.337	100.48	0.000
Correlat*FC-stand	4	25.355	25.355	6.339	1889.36	0.000
Grid siz*Correlat* FC-stand	8	9.591	9.591	1.199	357.35	0.000
Error	4021	13.491	13.491	0.003		
Total	4049	519.127				

Model five: All three-factor interactions have been removed.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Sample s	2	1.810	1.810	0.905	157.99	0.000
Grid siz	2	14.050	14.050	7.025	1226.25	0.000
Correlat	2	379.880	379.880	189.940	3.3E+04	0.000
FC-stand	2	68.744	68.744	34.372	5999.70	0.000
Grid siz*Correlat	4	4.856	4.856	1.214	211.92	0.000
Grid siz*FC-stand	4	1.348	1.348	0.337	58.84	0.000
Correlat*FC-stand	4	25.355	25.355	6.339	1106.46	0.000
Error	4029	23.082	23.082	0.006		
Total	4049	519.127				

Model six: The two-factor interaction, "BD", between grid size and FC-standard has been removed.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Sample s	2	1.810	1.810	0.905	149.41	0.000
Grid siz	2	14.050	14.050	7.025	1159.72	0.000
Correlat	2	379.880	379.880	189.940	3.1E+04	0.000
FC-stand	2	68.744	68.744	34.372	5674.17	0.000
Grid siz*Correlat	4	4.856	4.856	1.214	200.42	0.000
Correlat*FC-stand	4	25.355	25.355	6.339	1046.43	0.000
Error	4033	24.430	24.430	0.006		
Total	4049	519.127				

Model seven: The only remaining two-factor interaction is now the interaction between correlation and FC-standard. In addition, the main effect sample size has been removed.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Grid siz	2	14.050	14.050	7.025	912.45	0.000
Correlat	2	379.880	379.880	189.940	2.5E+04	0.000
FC-stand	2	68.744	68.744	34.372	4464.38	0.000
Correlat*FC-stand	4	25.355	25.355	6.339	823.32	0.000
Error	4039	31.097	31.097	0.008		
Total	4049	519.127				

Appendix D

Appendix D includes generated ROC – curves not showed in the results section of the document.

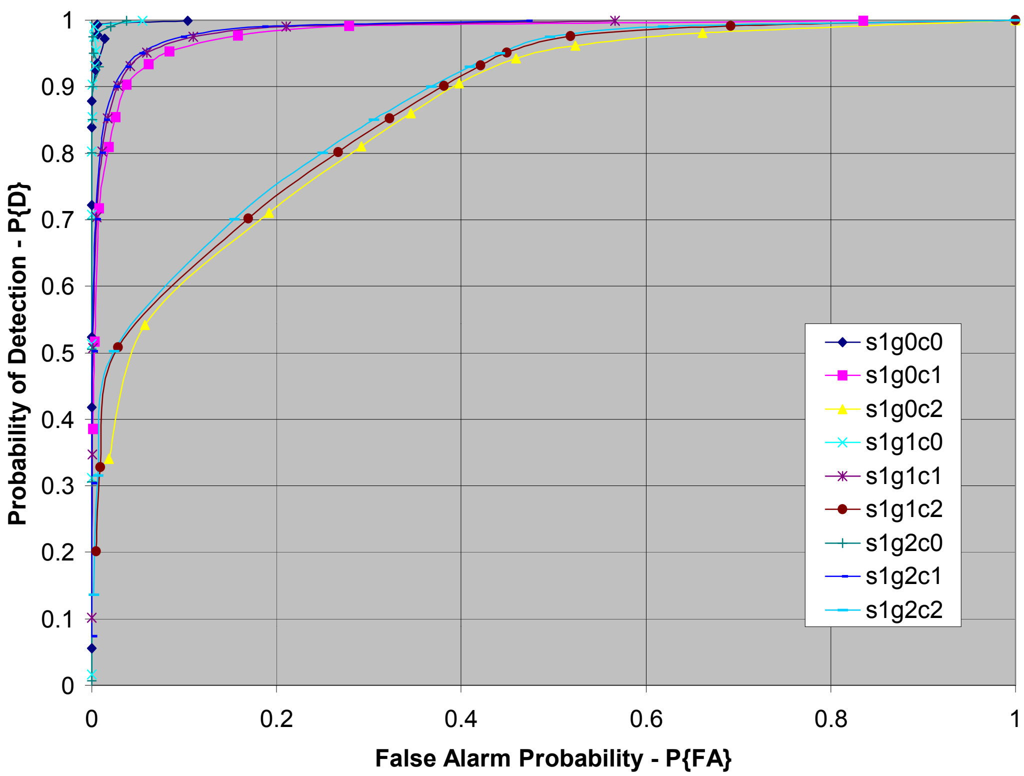

Figure D-0-1: ***ROC-curves for sample size equal to 10,000***

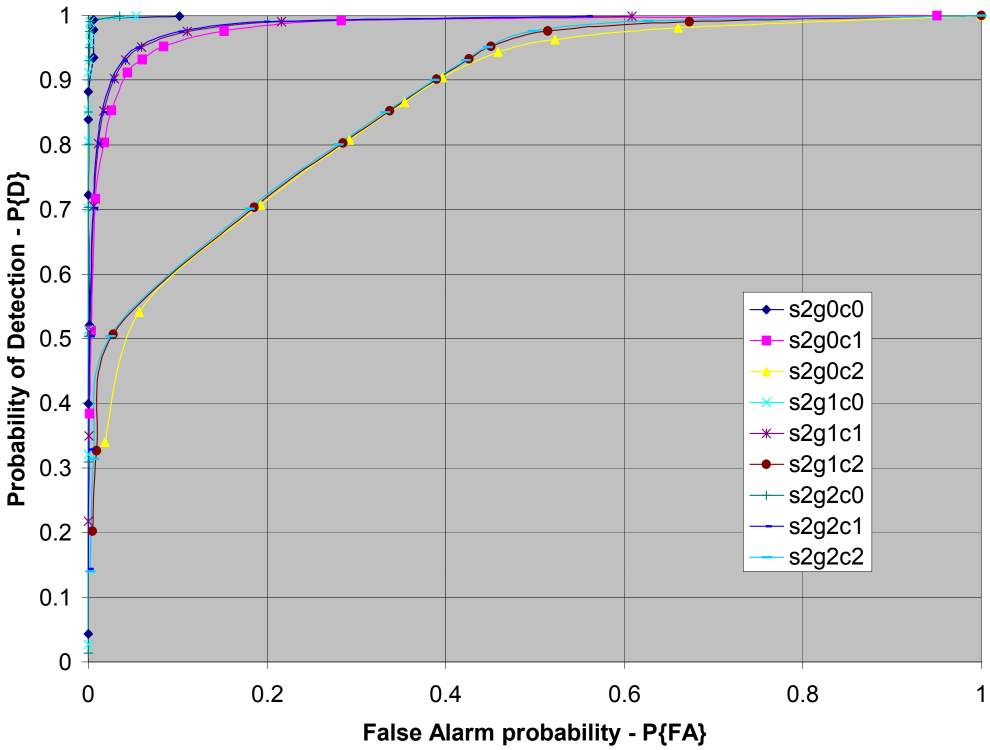

Figure D-0-2: ***ROC-curves for sample size equal to 100,000***

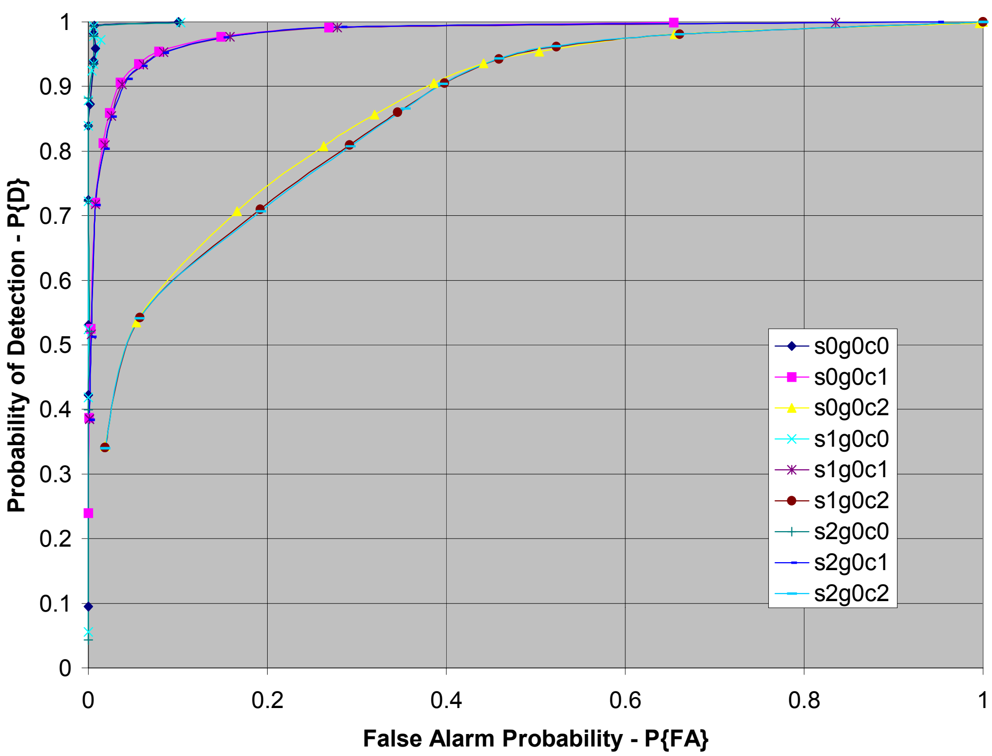

Figure D-0-3: ***ROC-curves for grid size equal to 5***

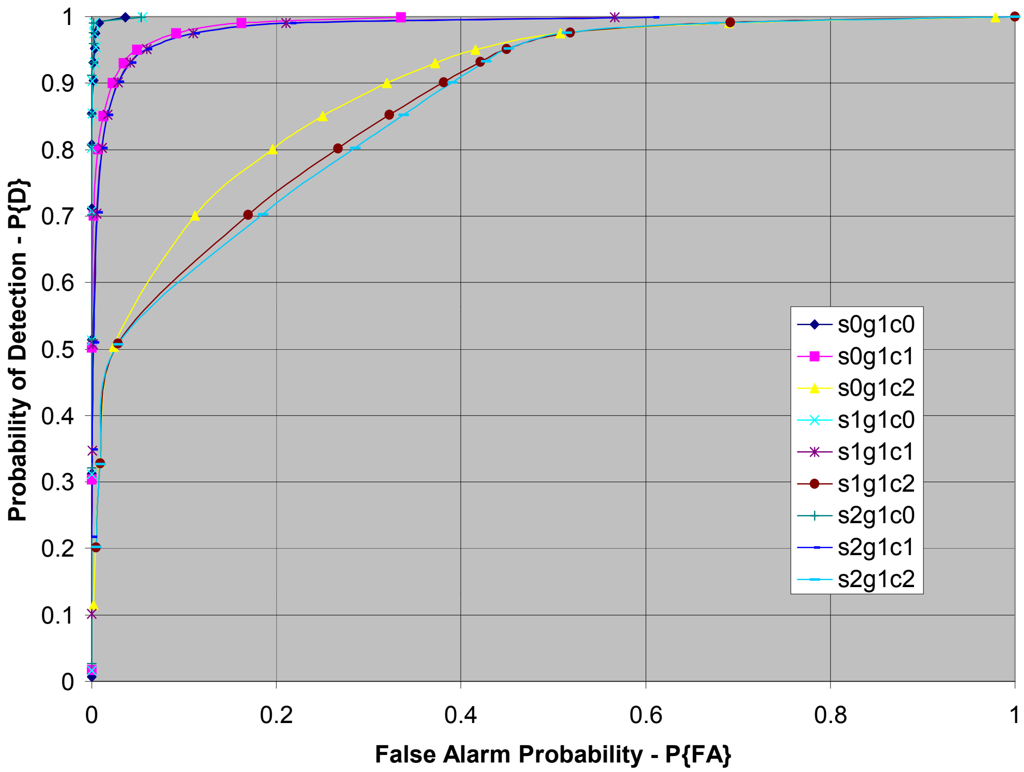

Figure D-0-4: ***ROC-curves for grid size equal to 10***

VITA

Karl-Johan Nyberg was born in Landskrona in the southern province of Sweden, Skane. He grew up and spent most of his childhood in the town of Kagerod. He is the son of Majken and Sven Nyberg. His Swedish equivalent of a Bachelors Degree in Industrial and Systems Engineering (Industriel Ekonomi) was obtained at the University of Lulea, Sweden, in May 1998. Karl-Johan received his Master of Science Degree in Industrial and Systems Engineering from Virginia Tech's Grado Department of Industrial and Systems Engineering in March of 2003.

Karl-Johan lives in Arlington, Virginia and is currently employed by Stanley Associates, Inc. as a Software Manager.