3-5-2005

Optimization of Automatic Target Recognition with a Reject Option Using Fusion and Correlated Sensor Data

Trevor I. Laine

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Signal Processing Commons

OPTIMIZATION OF AUTOMATIC TARGET
RECOGNITION WITH A REJECT OPTION
USING FUSION AND CORRELATED SENSOR
DATA

DISSERTATION

Trevor I. Laine, Major, USAF

AFIT/DS/ENS/05-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

OPTIMIZATION OF AUTOMATIC TARGET RECOGNITION WITH A REJECT

OPTION USING FUSION AND CORRELATED SENSOR DATA

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Trevor I. Laine, BS, MBA, MS

Major, USAF

25 April 2005

OPTIMIZATION OF AUTOMATIC TARGET RECOGNITION WITH A REJECT

OPTION USING FUSION AND CORRELATED SENSOR DATA

Trevor I. Laine, BS, MBA, MS
Major, USAF

Approved:

Date

// Signed //	25 April 2005

Dr. Kenneth W. Bauer (Chairman)

// Signed //	25 April 2005

Dr. John O. Miller (Member)

// Signed //	25 April 2005

Dr. Mark E. Oxley (Member)

// Signed //	25 April 2005

Dr. William F. Bailey (Dean's Representative)

Accepted:

// Signed //	27 April 2005

Dr. Robert A. Calico, Jr.   Date
Dean, Graduate School of Engineering and Management

Abstract

In many pattern recognition applications, significant costs can be associated with various decision options. Often, a minimum acceptable level of confidence is required prior to making an actionable decision. Combat target identification (CID) is one example where the incorrect labeling of Targets and Non-targets has substantial costs; yet, these costs may be difficult to quantify. One way to increase decision confidence is through fusion of data from multiple sources or from multiple looks through time. Numerous methods have been published to determine optimal rules for the fusion of decision labels or to determine the Bayes' optimal decision if prior probabilities along with decision costs can be accurately estimated. This research introduces a mathematical framework to optimize multiple decision thresholds subject to a decision maker's preferences. The decision variables may include rejection thresholds to specify Non-declaration regions and ROC thresholds to explore viable true positive and false positive Target classification rates. This methodology yields an optimal class declaration rule subject to decision maker preferences without using explicit costs associated with each type of decision.

This optimization framework is demonstrated using various generated and collected sensor data. The experiments using generated data were performed to gain insight of the potential effects of fusing data with various degrees of correlation. The optimization framework is then applied to assess two competing fusion systems across four test sets of radar data. The fusion methods include Boolean logic and probabilistic neural networks for the fusion of collected 2-D SAR data processed via 1-D HRR moving target algorithms. Excursions are performed by varying the prior probabilities of Targets

and Non-targets and varying the correlation between multiple sensor looks. In addition to

optimizing thresholds according to decision maker preferences, an objective function is

presented to facilitate comparison between CID systems, where the time associated with

each look is incorporated.

Acknowledgements

I would like to thank many individuals who supported me in this challenging, but rewarding, Ph.D. endeavor. I begin with inexpressible gratitude to my wife, whose support and sacrifices were essential to my success. I am also grateful for our son and his ability to make me laugh, take time to play, and to keep things in perspective during times when the road ahead seemed rough.

I am indebted to my research advisor, Professor Kenneth Bauer, Jr., without whom I may have never considered the challenges of a Ph.D. program. His expertise, guidance, and enthusiasm for research enabled me to reach my academic potential. He is truly a complete advisor. I am also grateful to my research committee, Professors Mark Oxley and J. O. Miller for all of their positive feedback and support.

I would also like to thank the AFOSR, ACC/DR and AFRL/SN for sponsoring this research. Specifically Mr. Chuck Sadowski provided a warfighter perspective for Combat ID, which helped to focus this research effort. I owe Dr. Kevin Priddy thanks for providing an attractive ATR data set and for his help and encouragement along the way.

In addition, I need to acknowledge my friends and colleagues in the BARF (Behind AFIT Relocatable Facility). The synergy amongst the Ph.D. students in the BARF cubicles has led to increased research efficiency. I owe specific thanks to Major Tim Albrecht for his help to process HRR features. The BARF also provided a place for fellow Ph.D. students to relax and ponder various issues from academia, the Air Force, AFIT, or life in general.

Finally, I offer a special thanks to my parents. Their love, guidance and encouragement prepared and enabled me to accomplish this endeavor – thank-you.

- Trevor I. Laine

Table of Contents

List of Figures

List of Tables

INVESTIGATION OF FUSION FOR ATR WITH NON-DECLARATIONS AND

CORRELATED INPUT DATA

I. Introduction

1.1 Combat ID Background

With recent technological advancements in precision engagement and stealth, "if

the enemy's key targets, target sets, or COGs (centers of gravity) can be found and

identified, they are usually within airpower's reach" (Dept. of AF 2000: 42). Combat

target identification (CID) is hence identified by *Air Force Doctrine Document (AFDD)*

2-1: Air Warfare, as one of the limiting factors in our ability to engage the enemy. An

assessment of the current state of CID by Haspert (2000) concurs with this assessment of

CID and goes on to state, "CID is often viewed as the weakest part of the military's kill

chain." The links in the complete kill chain may include: search, detect, track, classify,

identify, assign, fire control calculations, weapons launch, mid-course guidance, target

acquisition by the weapon, terminal homing, fuse, target damage, and battle damage

assessment. With good Combat ID hostile targets may be engaged with a minimal

probability of fratricide and with limited unintentional collateral damage of neutral

forces. In a recent *Air Force Magazine Online* article, Cahlink (2004) quotes Lt. Gen.

Leaf, the USAF liaison to the land component commander during Operation Iraqi

Freedom (OIF) who states, "in terms of fratricide, zero is the only good score, and we're

not there yet." Cahlink goes on to state, "preliminary analysis showed that fratricide of

all types accounted for about 11 percent of 115 US battle deaths" in Gulf War II (OIF).

This number of fratricides is lower than those obtained in Desert Storm where, "fratricide was blamed for 35 of 148 U.S. battle deaths," which is about 24 percent (Cahlink, 2004). A related article by Hebert (2004) quotes Army Brig. Gen. Robert W. Cone, who led Joint Force Command's (JFCOM's) lessons learned from Gulf War II. He states, "In terms of CID, I don't think we've made a lot of progress in the last 10 years" (Hebert, 2004). Hebert goes on to state, "DoD identified fratricide prevention as its top priority," and "eliminating fratricide requires two advances: accurate CID and better blue-force tracking," (Hebert, 2004). Thus, Haspert's statement of CID being considered one of the weakest parts of kill chain is currently supported in the DoD community and improvement in Combat ID is top research priority for the Department of Defense.

1.2 Introduction to Automatic Target Recognition

Combat ID includes the identification of potential targets using both cooperative systems and non-cooperative identification methods. One example of cooperative Combat ID includes a direct question-and-answer identification, friend or foe (IFF) system. This system may be used to interrogate a potential target using electronic communication between two friendly systems. When feedback is not obtained, the Combat ID must be made using non-cooperative means. The non-cooperative means may include a man-in-the-loop to make a final decision of whether or not the potential target is indeed a hostile. One potential man-in-the-loop method of Combat ID is the visual verification of a ground target by the pilot prior to engagement. If a non-cooperative Combat ID is performed autonomously by an identification system, it is considered to be an automatic target recognizer (ATR). Automatic target recognition

may include tasks of detecting, tracking, and classifying potential targets. Such a system may be referred to as an Automatic Target Detection/Recognition (ATD/R) system. With the emergence of an increased volume of electronic sensor data, along with an increase in the communication bandwidth between platforms, Combat ID research specifically aimed at improvements in ATR may have substantial benefits. For example, Hebert (2004) states, "senior officials have noted that some assets, such as Global Hawk are so effective at collecting intelligence that they can't be used at full capacity." Improved ATR systems would help streamline the Combat ID process and allow the USAF to use Global Hawk at more than the one-third capacity used during Operation Iraqi Freedom (Hebert, 2004). Thus, while great improvements have been made for the operational use of unmanned aerial vehicles (UAVs) to perform reconnaissance in support of the search phase of the military kill chain, the current intelligence processing methods are not able to utilize the full capacity of these assets. Hebert (2004) goes on to note, the Link 16 now transmits targeting information electronically rather than through voice communication. This electronic communication may occur from the Air Operations Center (AOC) to an Airborne Warning and Control System (AWACS) to a strike aircraft. Thus, as electronic communication capabilities increase and the volume of data grows, the requirement to fuse data automatically from multiple sources is likely to grow. This sharing and fusion of data from multiple sources is a key to netcentric warfare.

As identified in the Draft Capstone Requirements Document for CID, "Combat Identification is the process of attaining an accurate characterization of detected objects in the joint battlespace to the extent that high confidence, timely application of military options and weapons resources can occur." An example of a notional ATR system is

provided as Figure 1.1. The primary goal of this system is to provide better battlespace

characterization from which actionable decisions can be made by the warfighter.

Decisions may include engagement of Hostile targets, a new allocation for sensors to

identify a new Region of Interest (ROI) after non-targets have been identified, etc. Such

a system could use data from multiple sensors, denoted as A and B in Figure 1.1, in the

attempt to identify a potential target located in a ROI.

$$E \xrightarrow[\text{\textit{Sensor}}]{s} D \xrightarrow{p} F \xrightarrow[\substack{\text{\textit{Classifier or}}\\\text{\textit{additional}}\\\text{\textit{Processor}}}]{c \text{ or } p} L \text{ or } F \xrightarrow[\text{\textit{Fusion}}]{f} L$$

Labels or
Refined Features

At a minimum, the
final ATR decision
Is **trichotomous**

Data · Features

Region of Interest (ROI)	Sensor A data at time t	p_1	Sensor A features at time t	c_1 or p_3	Sensor A label or features at time t

Final Decision "Label"

Something Observable at time t

fusion database

f · fusion rule

"Target" "Non-target" or "No declaration" at time t

Sensor B data at time t	p_2	Sensor B features at time t	c_2 or p_4	Sensor B label or features at time t

Fusion process stores and incorporates previous ROI information

Figure 1.1 Notional ATR System with Sensors A & B Collecting Data through Time

The ROI was selected by at least one sensor, where enough evidence was

obtained to suggest that a desirable target is likely to be located in the general area. The

two sensors may be hosted on the same or different platforms and more than two sensors

may be used. As identified in USAF doctrine, the ATR process must obtain enough

evidence to reach a desired level of confidence in the labeling of the object, prior to

making a shoot decision (Dept of AF, 1999, 2000). Thus, if enough confidence is not

obtained a "Non-declaration" is a desired output label to the warfighter (Sadowski, 2004). In these situations, the ATR system may continue to acquire new information from additional looks by one or more sensors. This new data should then be fused to obtain an updated decision for the correct labeling of the object. With a "Non-declaration" always a desired label option, the ATR system would be required to have a minimum of three output labels including "Target," "Non-Target," and "Non-declaration."

Since the ATR system would be employed against potentially moving hostile targets, a real-time capability to acquire additional data after a "Non-declaration" label is generated is desired. Since no other platforms may be available to help in the ID process, the current ATR system may be forced to take multiple looks of the same potential target in a limited time. These multiple looks across limited differences in viewing angles, would likely contain similar information, and may likely be highly correlated. The assessment of fusion methods with data representative of different correlation structures is desired to help understand the potential effects of collecting sensor data across various correlation levels.

As will be defined in Chapter 2, one common assessment technique for ATR systems is the use of a Receiver Operating Characteristic (ROC) curve (Alsing, 2000). The ROC curve shows the trade-off between two performance measures of interest, including the probability of true positive target declaration and the probability of false positive target declaration, as a decision threshold is varied. Yet, the standard ROC curve only provides insight of a dichotomous decision and does not show any temporal relationships. Although the ROC curve is widely used in the ATR community, the additional impact of "Non-declarations" and the ATR system time required to obtain a

traditional ROC curve true positive performance values is not readily visible to facilitate the comparison of ATR systems. Thus, to determine a preferred operational ATR system, the warfighter not only needs to know the relative true positive and false positive rates, but the associated number of re-looks associated with "Non-declarations" and the associated time required by the system (Sadowski, 2004).

Thus, a goal of this research is to develop a ROC-like measure of performance for Combat ID ATR systems. This measure of performance should help evaluate competing fusion systems and be inclusive of both time measures and rejection parameters. Further, it is highly desired to perform such evaluation without determining the explicit costs associated with incorrect classifications (Sadowski, 2004). While the literature reviewed includes methods of determining an optimal system with respect to misclassifications and rejections or "Non-declarations," this is accomplished by use of a cost function, where equivalent units are required for both misclassifications and "Non-declarations." For example, the relative costs for the misclassification of a friend as a "Hostile," which may contribute to a fratricide is difficult to place in the same cost units as the cost of a "Non-declaration," which simply triggers a re-look of an ROI. Therefore, the evaluation of an ATR system without use of explicit costs is highly desired. Once a methodology is determined, evaluation can then be performed using data with various degrees of correlation. This assessment should help determine some of the effects of fusing independent data vs. fusing data that may be correlated across sensors or within a sensor as it obtains multiple looks through time.

1.3 Contributions of this Research

This research makes several contributions by addressing the research goals outlined in the previous section. First, a comprehensive review of the literature is performed to capture the current performance measures used for ATR system assessment. From this review, no current ROC-like methodology was found that allowed for a temporal assessment of a classification system, inclusive of "Non-declarations." Further, most assessments of classification accuracy with a "Non-declaration" option were optimized to be either compliant to a predetermined number of unlabeled objects, or to minimize the overall risk of a Loss function associated with the classification system. The Loss functions require estimates of target class prevalence along with costs associated with each type of incorrect output label, where "Non-declaration" costs must be placed in comparable cost units to all other feasible misclassifications.

A mathematical framework is developed to determine an optimal ATR system, inclusive of a developed temporal objective function. This objective function extends traditional ROC curve analysis, by offering identification of preferred ROC points, using assessments of both time and "Non-declarations." This measure includes the evaluation and optimization of a minimum of two variable thresholds used to make "Target," "Non-target," and "Non-declaration" decisions. This is accomplished without use of explicit costs through a mathematical formulation. This mixed variable programming framework is developed as a new and flexible evaluation method for ATR. The optimization framework is then used to assess and compare different fusion methods using generated data. In performing some of these experiments, a synthetic classifier fusion test

environment was established. This synthetic fusion test environment includes the

generation of multidimensional Gaussian data with desired correlation structures both

across simulated sensors and through multiple looks in time. Justification for use of

multivariate Gaussian data to represent sensor features is also provided. The mixed

variable framework was then demonstrated across two fusion methods with two

polarimetric channels of radar data fused across time. This experiment demonstrates the

utility of the optimization framework on a new data set collected in 2004 and obtained

from the Air Force Research Lab's Sensor Directorate (AFRL/SN).

While the research contained within this document maintains a focus on DoD

military target applications, this research is applicable across a wide range of

classification applications. For example, significant ROC curve research has been

performed in the medical community, where ROC curves are a commonly employed

decision tool (Swets *et al*., 2000; Metz, 1986, 1989). Medical data may also be derived

from multiple sensors (X-ray, CT scan MRI) or from multiple diagnostic tests. This data

may then be combined to obtain a best fused diagnosis for a patient. A majority vote is

one common technique used to fuse independent results for a given disease (Kuncheva,

2004). Current medical research also seeks to automatically assess imagery data for the

determination of cancer vs. benign growths. As with military applications,

misclassification costs may significantly outweigh "Non-declaration" costs. For a

medical application, a "Non-declaration" may have a small time and monetary cost

associated with another diagnostic test or image, while a false negative classification may

lead to substantial lost treatment time and a false positive may lead to substantial

emotional stress of a patient. In addition to medical applications, similar applications of

classification tasks requiring data to be fused through time may be found in many other areas. This list may include applications for automatic system prognosis, robotics, and environmental monitoring, among others (Hall and Llinas, 2001).

1.4 Organization of this Document

The remainder of this document is organized as follows. Chapter 2 provides a review of the pertinent literature for the investigation of fusion with unknown class designations and correlated input data. This review contains four main sections. They include an introduction to fusion and fusion process models for ATR, a background of sensor features and the potential levels of correlation found in sensors, an introduction to some of the models used for sensor fusion, and an overview of potential measures of performance used to assess ATR systems. Chapter 3 provides a methodology for the mathematical framework used to compare ATR systems including a proposed objective function inclusive of time. Chapter 4 presents some examples of the optimization framework using generated data. This chapter also includes a multivariate Gaussian data generation method and justification for its use. Chapter 5 presents an illustrative example of the optimization framework to compare two competing fusion systems using two channels of collected radar data. This chapter also includes significant sensitivity analysis across variables of interest. The final chapter presents a summary of contributions and findings along with thoughts for the continuation of related research.

II. Literature Review

This literature review is arranged with the following primary sections. Section 2.1 is an introduction to data fusion for Combat ID and Automatic Target Recognition (ATR). Section 2.2 provides a review of the sensor environment expected for ATR systems. Section 2.3 provides an overview of some of the methods to perform fusion. Section 2.4 presents common techniques used to assess ATR performance. A summary of the findings from the literature is the included as Section 2.5.

2.1 Introduction to Data Fusion for Automatic Target Recognition

This section provides a basic introduction to key components of Automatic Target Recognition as a subset of Combat ID. Intelligence data sources are first described, followed by a discussion of fusion for Intelligence, Surveillance, and Reconnaissance (ISR) applications. Definitions are then presented for different types of correlation that may be found within the sources of sensor data to be fused in the ATR process. An overview of sensor fusion process models is then presented, followed by a discussion of the relationship between ATR and these fusion models.

2.1.1 Intelligence Data Sources

The complete set of ISR images available for analysis and target identification over a specific area of interest are likely to be comprised of a mix of sensors collected from different ISR platforms. The intelligence derived from visual photography, infrared sensors, lasers, electro-optics, and radar sensors is collectively known as imagery intelligence (IMINT), as defined in Air Force Doctrine Document (AFDD) 2-5.2,

Intelligence, Surveillance, and Reconnaissance Operations (Dept. of AF, 1999). Typical

sensor types include electro-optical (EO), infrared (IR), synthetic aperture radar (SAR),

high resolution range (HRR) radar, and radar used for moving target indication (MTI),

along with the more recent addition of multispectral (MSI) and hyperspectral imagery

(HSI). While EO, IR, and radar data provide a single image, MSI and HSI data contain

multiple images of the same region obtained in different frequency bands. MSI data is

typically comprised of data in 5-12 spatially disjoint electromagnetic frequency bands

covering the visible and infrared spectrum, while HSI data may contain upwards of over

200 frequency bands (Langrebe, 1998) across the same electromagnetic frequencies. The

collection of this spectral data for a Region of Interest (ROI) is often referred to as a data

hypercube. Analysis of these IMINT sources may integrate or fuse information from two

or more IMINT sources or other intelligence sources to increase the accuracy of the

intelligence assessment. Other intelligence sources include signature intelligence

(SIGINT), measurement and signature intelligence (MASINT), human resources

intelligence (HUMINT) and open-source intelligence (OSINT). SIGINT includes

communications intelligence (COMINT), electronic intelligence (ELINT) and foreign

instrumentation signals intelligence (FISINT); MASINT includes scientific and technical

intelligence derived from sensor types used for IMINT and SIGINT; and OSINT includes

all publicly available information, such as newspaper, radio and television broadcasts.

Further discussion of intelligence sources can be found in AFDD 2.5-2 and AFP 14-210,

the *USAF Intelligence Targeting Guide* (Dept. of AF, 1998).

2.1.2 Principle of ISR Fusion

AFDD 2-5.2 identifies and defines 11 guiding Air Force ISR principles as:

- General
- Integration
- Accuracy
- Relevance

- Timeliness
- Fusion
- Accessibility
- Security

- Unity of Effort
- Interoperability
- Survivability,
  Sustainability, and
  Deployability

These principles are each defined in operational terms with illustrative examples and discussion. A common theme to all 11 ISR principles is the need for them to work synergistically to provide optimal information with maximum utility to commanders and decision makers. Thus, to fully optimize the principle of *fusion* other principles must also be considered. The *USAF Intelligence Targeting Guide* (Dept. of AF, 1998: 22) defines *fusion* as, "the process of combining multisource data into intelligence necessary for decision making," and goes on to state:

> Due to limitations inherent in any collection system, and because other
> countries strive to misinform or deny information to intelligence gathering
> agencies, intelligence normally should not be based on single source data.
> Intelligence becomes more useful and more reliable when information
> from all possible sources is collected, combined, evaluated, and analyzed
> in a timely manner.

From the above statement, the principle of *fusion* works in concert with the other principles, such as *timeliness, accuracy, integration, etc*. to provide optimal information to the commanders and decision makers at all levels.

While both AFDD 2-5.2 and AFP 14-210 clearly state ISR derived information shall be combined, evaluated, and analyzed to produce accurate intelligence, neither document provides details on how to accomplish this fusion. To further complicate intelligence analysis, growth in the total volume of information available and the

resulting dimensionality of data requiring fusion continues to grow. Technical advances

in our current "information age" have led to growth in the total data available for fusion,

where increased sensor resolution, increased bandwidth to share information, increased

ISR platforms including UAVs and satellites, and new sensor types like MSI and HSI by

their very nature can add significant amounts of data for any particular region of interest.

Dasarathy, a leading information fusion researcher, also points out *temporal* fusion

increases the dimensionality of the fusion process, and the spectral fusing of information

acquired across a period of time has not been well recognized (Dasarathy, 1997: 27).

Two other leading information fusion researchers Hall and Llinas, also recognize the

growing dimensionality of data available for fusion, where object recognition or target

identification is dominated by methodologies using a feature vector derived from sensor

data to represent an object or potential target in a feature space with defined class

boundaries (Hall and Llinas, 1997: 19-20). While many techniques for pattern

recognition using feature vector input are available to the analyst, Hall and Llinas note:

> …the ultimate success of these methods depends upon the ability to select
> good features. (Good features are those which provide excellent class
> separability in feature space, while bad features are those which result in
> greatly overlapping areas in feature space for several classes of targets.)

They then remark, "…more research is needed to guide the selection of features and to

incorporate explicit knowledge about target classes," (such as other intelligence

information). Guidance for the selection of features can be found in *Pattern Recognition*

using Neural Networks, (Looney, 1997: Ch 10 Feature and Data Engineering) with three

goals for mapping data into a feature space summarized as:

1. Retain as much relevant information as possible
2. Remove as much redundant information and extraneous noise as possible
3. Render the measurement data to variables more suitable for decision making

To accomplish goal 2 from above, the estimated linear correlation is typically used to measure the degree of association and linear dependence between any two random variables or features. This linear correlation between features is a primary measure used to indicate possible redundancy or dependence between features, where an increased number of independent features can provide greater discrimination power. However, in the presence of noise an increased number of highly correlated features can actually decrease the ability to discriminate between objects.

Thus, as identified in current literature, both fusion of time series data and feature selection are two areas of research where advancements in current methodologies could aid in the fusion process to derive optimal intelligence information given a set of collected data.

2.1.3 Types of Correlation

The linear correlation between two data variables of interest (raw data, target signatures, or refined features) can be used as a measure to identify linear dependence between data sources. The Pearson product-moment correlation coefficient, ρ, is a unitless value within the continuous interval of [-1.0, +1.0], with perfect linear correlation indicated by a value of +1.0 or -1.0 (Wilson and Keating, 1994: 75). For two continuous random variables (RVs) x and y to be considered independent, the joint probability distribution of x and y, denoted $f(x,y)$, must equal the product of the two marginal probability distributions defined as $f_1(x)$ and $f_2(y)$, such that $f(x,y) = f_1(x)f_2(y)$ (Hogg and

Ledolter, 1987: 100). In general, if the correlation coefficient between two random RVs is not 0, then the independence relationship above is not satisfied and the variables must be dependent. On the other hand, if $\rho = 0$, the variables may or may not be statistically independent (Hogg and Ledolter, 1987: 100). In practical terms, if two highly correlated variables are used to classify an object, they are clearly dependent and knowing the value of the second variable provides only a marginal increase in information beyond the first variable being used to assess the class membership of the object being studied. The entropy, *H*, (Shannon, 1948) associated with a probabilistic distribution is one approach to quantifying the relative information provided by observations of multivariate data.

The following section will provide specific definitions of *correlation*, *autocorrelation*, and *crosscorrelation*. The Pearson-product-moment *correlation* coefficient ρ, between two RVs *x* and *y* is defined in eq. 2-1 and is also known as the correlation across variables or features.

$$\rho_{xy} = \frac{E\left[(x - \mu_x)(y - \mu_y)\right]}{\sigma_x \sigma_y} = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \qquad (2\text{-}1)$$

From eq. 2-1, μ_x and μ_y are the population means and σ_x^2 and σ_y^2 are the population variances of the two RVs and σ_{xy} is the covariance between *x* and *y,* and does not include a temporal component.

Let *z* be a random variable with stationary mean and variance sampled at uniform intervals across time. The *autocorrelation* in RV *z* across *k* uniform time lags is defined as:

$$\rho(k) = \frac{E[(z_t - \mu_z)(z_{t+k} - \mu_z)]}{\sigma_z^2}. \tag{2-2}$$

where μ_z is the population mean and σ_z^2 the population variance of z. The

autocorrelation may also be referred to as correlation within a variable or feature.

The *crosscorrelation* between RVs *x* and *y* across time lag *k* is defined as:

$$\rho(k)_{xy} = \frac{E[(x_t - \mu_x)(y_{t+k} - \mu_y)]}{\sigma_x \sigma_y} \tag{2-3}$$

From the definitions above, eq. 2-1 defines the correlation across variables as the

crosscorrelation at lag 0 between two RVs and the autocorrelation or within correlation at

lag *k* in eq. 2-2 can be derived from eq. 2-3 when $x = y$. Further, the "(*k*)" is often

dropped if $k = 0$, indicating the correlation value does not include a temporal component.

Input features derived from sensors and used for ATR may or may not be

statistically independent. For example, some features derived from passive visual or

thermal sensors and reflected radar energy each containing different noise sources may be

statistically independent. Conversely, multiple looks by a single sensor across the time

continuum are likely to contain significant correlation. If a fusion algorithm assumes

independent input data for real-time ATR, violation of this assumption may overestimate

performance when significant correlation is present. As stated by Dudgeon (1998: 22):

> The assumption of independence is often justified, but in some cases it is
> not, and it may lead to inaccurate estimates of performance. Generally,
> independence between two random variables can be used as the limiting
> case where the value of one variable has no correlation with and conveys
> no information about the value of the other.

16

2.1.4 Fusion Process Models

This section briefly introduces various conceptual models of the data fusion process and the associated definitions and taxonomy as reviewed in the literature. The primary goal for each of these models is to facilitate discussion and a common language for use by those in the data fusion community including both researchers and practitioners. With numerous common terms used interchangeably with varying contextual meaning, the models are essential to establish a common nomenclature of definitions and concepts. Prominent models found in current literature include the UK intelligence cycle model, the Boyd control (OODA) loop, the revised Joint Directors of Laboratories (JDL) model, Dasarathy's fusion model, the Waterfall data fusion process model and the Omnibus model. With the exception of Dasarathy's model, these fusion models were developed primarily for military applications with significant interest and resource support by the U.S. and UK defense communities in the 1980's and into the 1990's. In addition to the mentioned models, numerous other similar models appear specific to a literature source; yet, most can easily be mapped into the before mentioned models.

By adopting a common model of information fusion and associated definitions, advancements made within one research community can more easily be put into practice by the growing multidisciplinary data fusion community. As proposed in (Hall & Llinas, 1997, 6-7), data fusion has, "…rapidly advanced to an emerging true engineering discipline with standardized terminology." Some specific definitions of types of fusion and their associated "levels" are presented in Table 2.1. It should be noted, this list of definitions is not all inclusive, lacking reference to a commonly referenced level 0 fusion,

and these definitions still appear to be dominated by their DoD roots and directly link to

the JDL fusion model.

Table 2.1 Data Fusion Terminology as Presented by Hall & Llinas, 1997

Fusion	The integration of information from multiple sources to produce specific and comprehensive unified data about an entity.
Alignment (Level 1)	Processing of sensor measurements to achieve a common time base and common spatial reference.
Association (Level 1)	A process by which the closeness of sensor measurements is completed.
Correlation (Level1)	A decision-making process which employs an association technique as a basis for allocating sensor measurements to the fixed or tracked location of an entity.
Correlator-Tracker (Level 1)	A process which generally employs both correlation and fusion component processes to transform sensor measurements into updated states and covariance for entity tracks.
Classification (Level 1)	A process by which some level of identity of an entity is established, either as a member of a class, a type within a class, or a specific unit within a type.
Situation Assessment (Level 2)	A process by which the distributions of fixed and tracked entities are associated with environmental, doctrinal and performance data.
Threat Assessment (Level 3)	A structured multi-perspective assessment of the distributions of fixed and tracked entities which result in estimates (e.g.): - Expected course of action - Enemy lethality - Unit compositions and deployment - Functional networks (e.g. supply, communication, etc.) - Environmental effects.

Of the fusion models, the UK intelligence cycle and the Boyd control loop are

similar in design, both being functionally oriented and cyclic in nature. The UK

intelligence cycle is presented in Figure 2.1 from (Bedworth, 1999) and (Bedworth and

O'Brien, 2000) where planning and action are encompassed within the dissemination

process. The Boyd control loop or OODA loop (Observe, Orient, Decide, Act) is

presented as Figure 2.2 with additional details found in (Boyd, 1987). While first used to model the military command process, the OODA loop has been widely adopted by the U.S. intelligence community as a framework for various levels of data fusion to operate within. Within both models, information fusion can occur within any of the four "blocks" (with the exception of Act block in the OODA Loop) with each block loosely representing a different "level" of fusion.
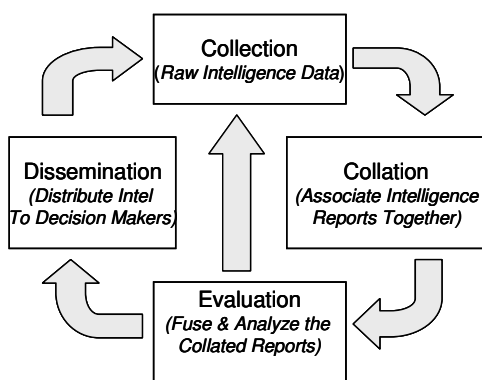
Figure 2.1 UK Intelligence Cycle
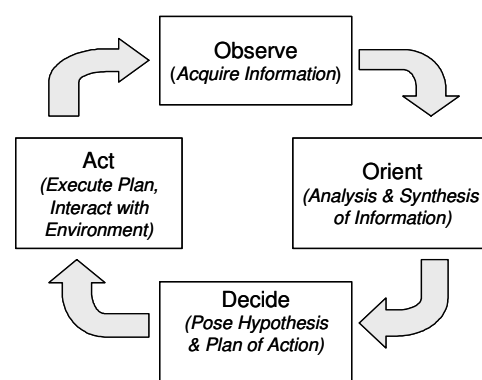
Figure 2.2 Boyd (or OODA) Loop

The JDL model was first proposed by the Data Fusion Working Group established for the study of information fusion by the DoD. This working group was established in 1986 and subsequently created the JDL model and a Data Fusion Lexicon (Hall & Llinas, 1997, 11). With an original focus on DoD applications, and emphasis on tactical targeting issues, the initial model was developed for military specific applications, but was later revised to encompass the growing nonmilitary applications such as manufacturing processes, complex system monitoring, robotics, and medical applications. Revisions to the to the JDL data fusion model are presented in (Steinberg *et al.,* 1999)

19

where the presented data fusion levels are intended to be a convenient categorization of data fusion functions, with actual data processing performed as required by an individual sensor fusion system. The revised JDL model is presented in Figure 2.3, where the data fusion domain includes Levels 0-4 and Database Management. Various sources of local input data have also been included for illustrative purposes corresponding to a military application.
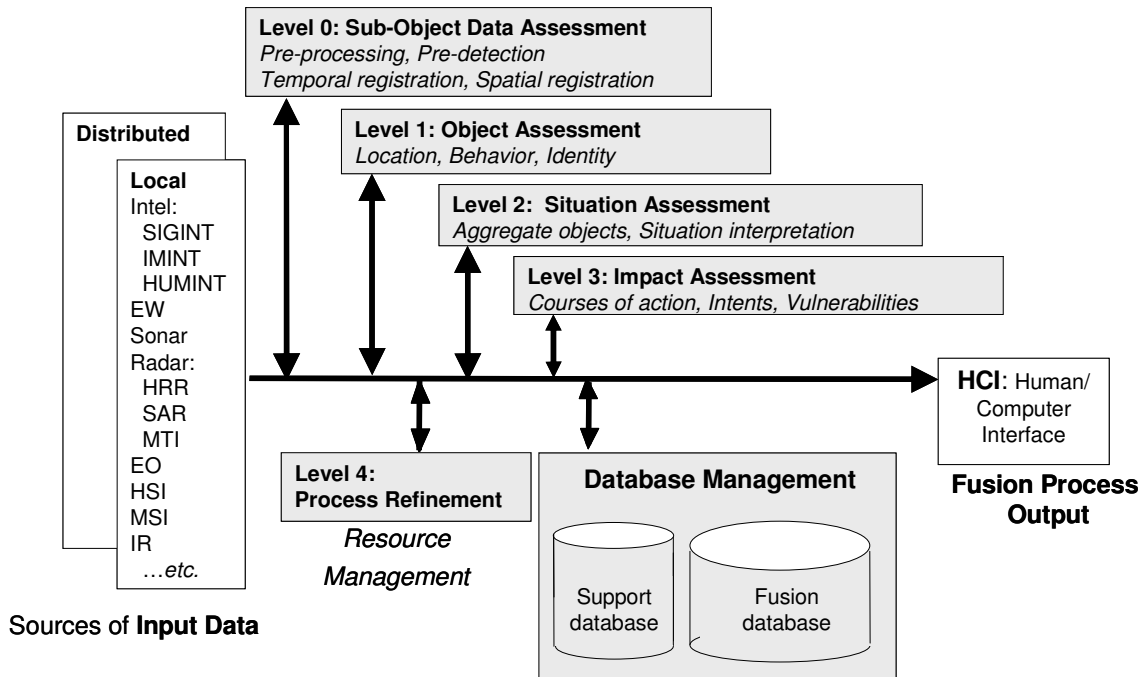
Figure 2.3 Revised JDL Data Fusion Model

Steinberg *et al.* (1999) include the following definitions for the revised JDL model levels:

Level 0 assessment involves hypothesizing the presence of a signal (i.e. of a common source of sensed energy) and estimating its state. Level 0 assignments include: signal detection on the basis of integration of a time-series of data and feature extraction from a region in imagery. A region may correspond to a cluster of closely spaced objects or to part of an object.

Level 1 assessment involves associating reports (or 'tracks' from prior fusion nodes in a processing sequence) into an association hypotheses. Each such track represents the

20

hypothesis that the given set of reports is the total set of reports available to the system referencing some individual entity.

Level 2 assessment involves associating tracks (i.e. hypothesized entities) into aggregations. The state of the aggregate is represented as a network of relations among its elements. Relations may be physical, organizational, informational, perceptual; as appropriate to a given system's mission. As the class of relationships estimated and the numbers of interrelated entities broaden, we tend to use the term 'situation' for an aggregate object of estimation.

Level 3 assessment is usually implemented as a prediction function, drawing particular kinds of inferences from Level 2 associations. Level 3 fusion estimates the "impact" of an assessed situation; i.e. the outcome of various plans as they interact with one another and with the environment. The impact estimate can include likelihood and cost/utility measures associated with potential outcomes of a player's planned actions.

Level 4 processing involves planning and control, not estimation. Similar to the formal duality between estimation and control, a duality between association and planning also exists. Level 4 assignment involves assigning tasks to resources.

Revisions to the JDL model include a generalization away from specific target tracking and target identification dominated terminology. Some noted changes include the new label for Level 3 "Impact Assessment" vs. the previous title of "Threat Refinement" and changing the "Source Pre-Processing" to the currently labeled "Level 0: Data Assessment."

The Waterfall model is similar to the JDL model in that multiple functional "levels" where data fusion can occur are clearly established. Each "level" or block represents a point in data refinement in which data from multiple sources can be combined and passed up to the next "level." Yet, it does not explicitly model feedback between levels as is included within the JDL model architecture in the data fusion domain. The Waterfall model has been adopted widely by the UK defense fusion community, but has not been significantly adopted elsewhere (Bedworth and O'Brien, 2000), possibly due to this limitation. The JDL levels and the Waterfall "levels" from

each block of Figure 2.4 are similar. The JDL level 0 corresponds to sensing and signal processing, JDL level 1 maps to feature extraction and pattern processing, JDL level 2 maps to situation assessment and JDL level 3 maps to decision making.
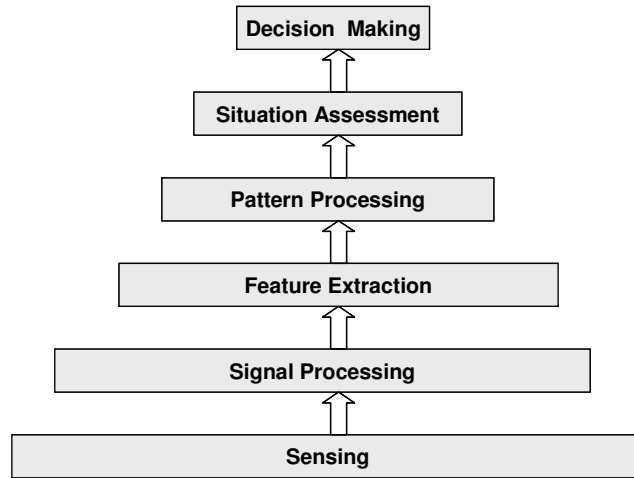
Figure 2.4 Waterfall Data Fusion Model (Bedworth, 1999)

Unlike the previous data fusion models based on the tasks or functional use of the data, the Dasarathy fusion model identifies levels of fusion based on the type of input information being fused and the resulting output, and is thus termed an I/O-based characterization model (Dasarathy, 1997). The three types of input and output include:

- Decisions: Belief values
- Features: Intermediate level values
- Data: Observed raw data with minimal manipulation

The three types of input and output lead to five distinct types of fusion, identified in Table 2.2.

Table 2.2 Five Levels of Information Fusion from the Dasarathy Model

Input	Output	Notation	Description/Analogy
Data	Data	DAI-DAO	Data-level fusion
Data	Features	DAI-FEO	Feature selection; Features extraction
Features	Features	FEI-FEO	Feature-level fusion
Features	Decisions	FEI-DEO	Pattern recognition; Pattern processing
Decisions	Decisions	DEI-DEO	Decision-level fusion

An assumed complexity for data fusion problems necessitates some level of data refinement into features before a decision can be made, thus DAI-DEO level fusion is excluded. The various levels of fusion can be combined to generate a flexible architecture for data fusion starting with given input at the *Data* level and a desired output at the *Decision* level. Figure 2.5 shows an encompassing framework, whereby fusion can occur at the parallel level, which in turn can be used as input to move upward from *Data* to *Feature* and eventually *Decision* level output.
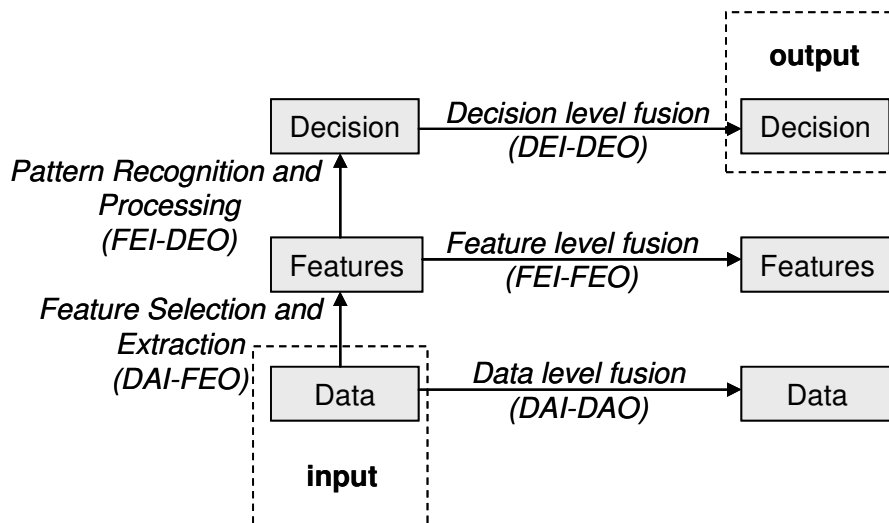
Figure 2.5 Dasarathy I/O Fusion Model, as derived from (Dasarathy, 1997)

A final model recently presented in (Bedworth and O'Brien, 2000) is the

Omnibus fusion model which is based on the Boyd OODA loop and cyclic nature of the

UK intelligence model. This Omnibus model incorporates the finer definitions from the

Waterfall model and can be mapped to both the JDL model based on tasks and can also

be mapped to the Dasarathy model based on the input/output characteristics of the fusion

occurring within any of the four Omnibus model levels of fusion to include: sensor data,

feature, soft decision, and hard decision. To note, feature level fusion is included within

the Orient process, with the selection of correct features for pattern processing identified

as one of the current limitations of feature fusion.
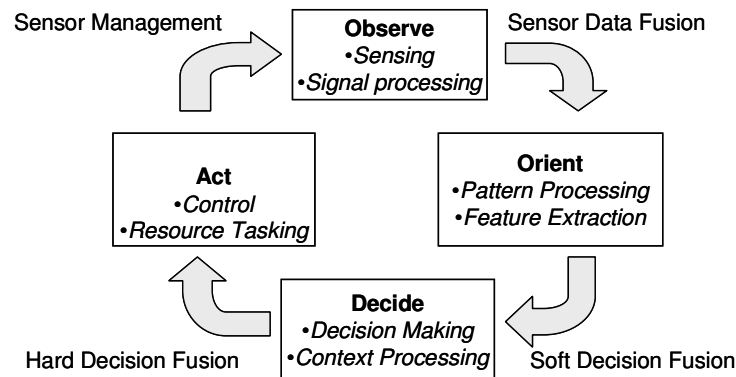
Figure 2.6 Omnibus Model for Data Fusion (Bedworth and O'Brien, 2000)

Table 2.3 is provided to compare and summarize levels where fusion occurs

within each of the described models. This table was inspired by a similar table presented

by Bedworth and O'Brien (2000) but has been modified with respect to the activity titles,

inclusion of the Omnibus and Dasarathy I/O models and minor differences in how the

fusion levels are mapped to the activities. An appropriate mapping between the activity

being performed in a decision making process was made after reviewing each model

24

individually and, if available, reviewing the mapping presented by Bedworth and O'Brien (2000). It should be noted each model is an abstract to facilitate discussion and should be used only as a guide with potential grey area between levels. Also, *impact assessment* can be viewed as a specific subset of *situation assessment* as identified in Steinberg *et al.* (1999).

Table 2.3 Comparison of Fusion Model Components as a Function of Activity

Activity	UK Intelligence Cycle	Boyd OODA Loop	Revised JDL model	Waterfall model	Dasarathy model	Omnibus model
Action	Disseminate	Act	HCI			Act
Decision making	Disseminate	Decide	Level 4	Decision making	DEI-DEO FEI-DEO	Decide
Impact assessment	Evaluate	Orient	Level 3	Situation assessment	DEI-DEO FEI-DEO	Decide
Situation assessment	Evaluate	Orient	Level 2	Situation assessment	FEI-DEO FEI-FEO DAI-FEO	Orient
Information processing	Collate	Orient	Level 1	Pattern processing / Feature extraction	FEI-DEO FEI-FEO DAI-FEO	Orient
Data processing		Observe	Level 0	Signal processing	DAI-FEO DAI-DAO	Observe
Detection	Collect	Observe	Input	Sensing	DAI-FEO DAI-DAO	Observe

2.1.5 ATD/R Models as Related to Fusion Process Models

While a requirement for military automatic target recognition (ATR) was identified 1960's, autonomous operational systems have still not been fielded (Nasr, 2003). Extensive advancements in theory and algorithms for target recognition have been made; yet by comparison, little focus has been placed on the testing and evaluation of these systems. The United States Air Force (USAF) Air Combat Command (ACC) is especially interested in objectively evaluating various ATR systems with a focus on operational goals. As discussed by Varner (2002), the warfighter is predominately

concerned with a "vertical" analysis of ATR system results, i.e. conditioned on the number of class declarations. In contrast, engineers tend to focus on a "horizontal" analysis of the system, i.e. conditioned on the number of actual objects from each class tested. The "horizontal" analysis may include receiver operating characteristic (ROC) curves which can be obtained from a confusion matrix representing the classification of all objects being tested under set classification rules (Ross *et al.,* 2002).

With prior discussion focused on information fusion models and a recent transition from military applications to a global encompassment of industrial, medical and other fields, the relationship of sensor fusion models to automatic target detection and recognition (ATD/R) models will be briefly examined. A general representation of the military ATD/R process is provided in (Schroeder, 2002) and is included as Figure 2.7. The process components in the ATD/R application should not be confused with the general processes identified in the fusion models, although some literature identifies levels of fusion based on an application specific model such as this. In general, process models of ATD/R will map into the fusion models. For example, each block of the ATR/D process model requires some decision with potential action and can be represented by a full cycle of the OODA loop. Progressing from *Detect* to *Identify*, increased levels of data resolution or data from multiple looks or sources may be required to further refine the assessment of a potential target. Therefore, as many of the previous fusion models indicate, this ATD/R application requires an iterative process, embodied within the blocks of Figure 2.7.

Detect	→	Discriminate	→	Classify	→	Recognize	→	Identify
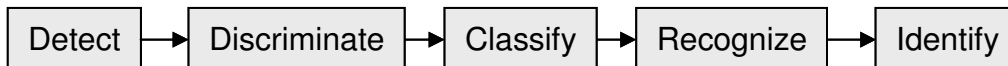
Figure 2.7 Process Model of ATD/R as Presented by Schroeder (2002)

The process blocks are defined as:

- Detect: Identify a Region of Interest (ROI) for analysis of a potential target
- Discriminate: Binary decision-target either present or not present in ROI
- Classify: Targets grouped into general class, e.g. Tank, Armored Personnel Carrier
- Recognize: Subdivision of class types, e.g. T-72 tank
- Identify: Unique identification of a target, e.g. assignment of serial number

Further, as presented by Sadowski (2001), the draft Capstone Requirements Document for CID shows five ways to characterize objects. This nomenclature includes hierarchical characterization, with "Friend/Enemy/Neutral" (FEN), refined to include "class" and "type." Yet, it also includes "nationality" and "intended mission" to provide a more complete characterization of battlespace objects.

	Air Target Example	**Ground Target Example**
Friend/Enemy/Neutral*	Enemy	Neutral
Class*	Bomber (of unk type)	Tank (of unk type)
Type*	Su-27	T-72
Nationality*	Serbia	Russia
Intended Mission*	Attacking, Moving Troops, Medical, or Defecting?	
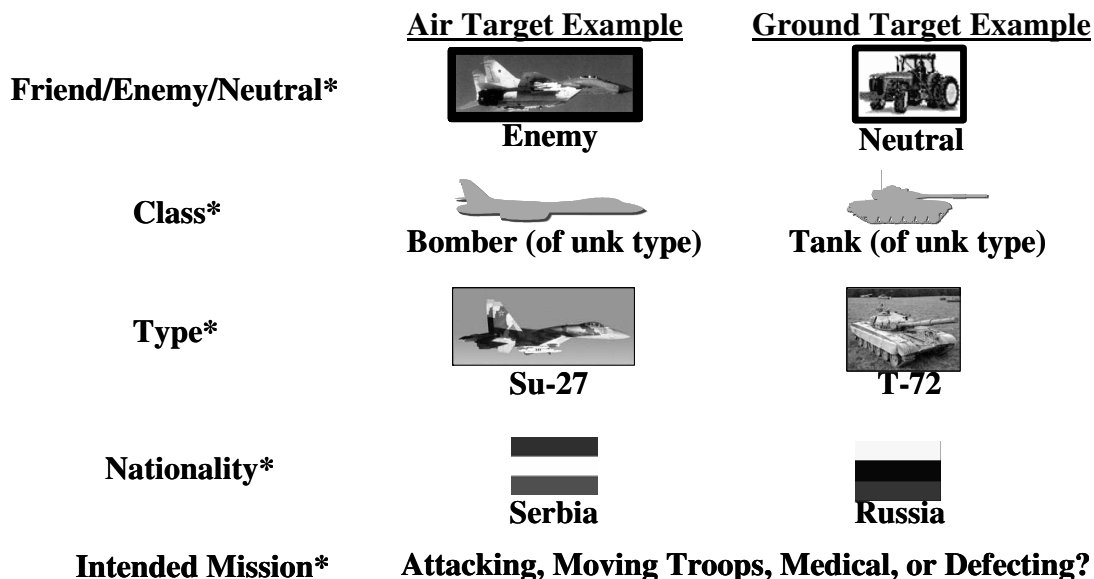
* Capstone Requirements Document for CID (Draft) and the Joint Mission Needs Statement

Figure 2.8 Five ways to "characterize objects," Sadowski (2001)

2.2 Sensor Features and Correlation for Fusion Research

This section is divided into three main themes and is then summarized. An overall goal is to obtain a general knowledge of the techniques involved with generating features from different sensors and what levels of correlation may be expected. To gain familiarity of potential feature extraction algorithms, a review of some current feature generation techniques for sensor data was undertaken and is presented in Section 2.2.1. After obtaining some working knowledge of the potential methodologies associated with feature generation, a review of Hyperspectral feature generation and challenges is addressed, since the features derived from the use of hundreds of adjacent frequency bands in the electromagnetic spectrum contain significant inherent correlation. A review of the literature was then performed in the attempt to discover what levels of correlation may be expected with different features extracted from different sensors. General conclusions are then drawn for the potential affects of correlated sensor data for ATR.

2.2.1 Sensor Features for ATD/R and Fusion

The intent of this section is to present some representative unclassified "open-literature" methods of obtaining target features from both SAR and MSI/HSI imagery data files. In doing so, an exhaustive review of feasible feature processing is not presented. This section does document some of the algorithms that may be used to generate target features for use as input data for classification algorithms. These features may be used as input data for feature level Fusion where multi-sensor features are presented to one classification model. The same input data may also be used as input for

different classification algorithms and then fused after posterior probabilities or class labels are determined for decision level Fusion.

As previously mentioned, good features should provide for optimal separability between classes, and typically involve the removal of redundant information and noise. Unlike many remote sensing applications with each individual pixel assigned to a class of land use (Langrebe, 1998, 2001), military target imagery collected from aerial platforms is likely to contain many pixels of information for each object to be classified and will be affected by the alignment and distance between sensor and target and possibly by luminance. The physical imagery differences obtained from aerial sensors and a target due to different angles and ranges from the same type of sensor, lead to the additional desired property of invariance for target features. Invariant feature generation should yield consistent feature space representation of a target image regardless of translation to a desired origin, rotation to a different axis and scale of the image. Several approaches of generating invariant features for optical and radar data are well documented in the relevant literature and include applications of the Fourier transform, the Karhunen-Loéve transform or principal component analysis (PCA), singular value decomposition (SVD), and methods of developing invariant histogram representations of 2D images. The following discussion contains a few potential feature generation methodologies.

Numerous image processing techniques incorporate the Fourier transform of an image from the spatial to frequency domain. One beneficial characteristic of a two-dimensional Fourier transform is that the frequency magnitude is invariant to translation and if a Fourier-Mellon transform in log-polar coordinates is performed, the magnitude is both rotation and scale invariant (Suvorova and Schroeder, 2002). While limited feature

extraction methodologies are presented in current literature specific to MSI and HSI data, feature extraction of visible optical data and SAR images are abundant and can be used as a starting basis to identify potential techniques that may be used for the observed energy in the smaller MSI/HSI frequency bands within the visible and IR spectrum. A Fourier based invariant feature generating technique proposed in (Wang *et al.,* 1994) is the moment Fourier descriptor (MFD) which is shown to be independent of an object's translation, rotation and scale. To generate the MFD features, *N* angularly equispaced radial vectors with an angular step of $2\pi/N$ are first generated from an object's centroid and end points located at the object's boundary. This initial mapping of the image leads to a periodic representation of the object. Moments for this periodic representation are then calculated, from which Fourier coefficients are obtained that are invariant to an object's rotation, translation and scaling. Use of MFD features is compared to traditional Fourier descriptors (FD) using classification accuracy as a metric, where fewer MSD features were required and obtained better classification accuracy for complex patterns when compared to FD features. The phase Fourier transform for invariant feature generation is described in (Paquet *et al.,* 1995) with sample range imagery classified with high accuracy by a neural network. The phase Fourier transform is used to segment planar and quadradic surfaces from a rigid object by using a limited number of normals. These normals were then grouped into a histogram to generate an invariant representation of an object. A final example of Fourier-type feature extraction from imagery is the exponential chirp transform (Bonmassar and Schwartz, 1997). The ECT is defined as a new linear integral transform combining space-invariance properties of the Fourier transform with internal image space-variant properties. In doing so, the Fourier

transform is generalized to the log-polar domain with an efficient order of operation. A unique property of the ECT is the ability to generate features which are equivalent to traditional Fourier transforms used for template matching and filtering, yet retain additional characteristics of the original image that are spatially variant, while remaining computationally efficient. Thus, while invariant features are desired, as can be obtained from Fourier transform methods, the ECT may also contain useful information for use by a more sophisticated pattern recognition technique such as artificial neural networks.

While widely used as an efficient linear projection of high dimensional data into lower dimensions, a PCA projection does not guarantee optimal separability among classes for pattern recognition applications. PCA projection is used to account for a desired amount of the observed variance by projecting the data via "rotation" into a new coordinate axis, where only the projection of data into axes accounting for a desired amount of the total data variability are retained. For example, the largest eigenvector in PCA produces the axis along which an entire image has maximum variance, but no guarantee is provided that the variance is maximized along the same eigenvector for all classes, or that a significant difference in class means will facilitate optimal discrimination between classes. Even with these drawbacks, PCA or the Karhunen-Loéve transform (KLT) has been successfully used for data reduction and feature generation in ATD/R environments. Efforts for ATR by (Schroeder, 2002) indicate KLT features are an integral part of generating invariant features from SAR data. A detailed methodology of such feature extraction is found in (Suvorova and Schroeder, 2002) where an initial step requires the calculation of rigid transformation invariant features (RTIF). The RTIF features are obtained by taking a Fourier-Mellin transform of the

image as the image is rotated through $(0, 2\pi)$ and sampled at uniform intervals. The KLT eigenvectors corresponding to the largest eigenvalues or variance are then retained as features. Experimental results indicate these features provide for good discrimination using a Mean Square Error (MSE) classifier for similar target types from the MSTAR data set including T-72 tanks, BMP-2 infantry fighting vehicles, and BTR-70 armored personnel carriers.

Similar to PCA, singular value decomposition has also been used in conjunction with Fourier transforms to produce a feature set of reduced dimensionality from imagery data. Research by (Bhatnagar *et al.*, 1998) indicates that data range-space eigenvectors account for 90% of target energy in High Range Radar (HRR) data, where radar returns for 100 frequencies were analyzed across 360° look angles sampled at 0.04^{o}. The resulting 100 x 9000 matrices for each target were transformed by taking a Fourier transform and Power transform retaining the same dimensionality (100 x 9000). This matrix was then partitioned into 2.5^{o} sectors with SVD used to generate 144 template vectors of dimension 100 x 1 corresponding to the largest singular value of the range space. Matched filters were then used for ATR target classification and were shown to be superior to using a feature extraction technique based on similar normalized mean values. Target recognition research performed using HRR and SAR radar imagery by (Cooke *et al.,* 2000) also demonstrates good discrimination between targets and clutter using SVD preprocessed radar data. While the initial radar feature preprocessing was not "documented in the open literature," reduction of these initial radar features was performed using SVD with a recursive Fisher discriminant used to select a further reduced set of 6 features derived from 27 x 27 pixel images with a false alarm rate just

over 1%. Thus, as part of an integrated feature engineering process, SVD is a demonstrated tool for feature-space reduction.

One form of invariant histograms is introduced by (Ikeluchi *et al.*, 1996) and is generated using weak invariants which are defined to be a feature generated by a pair of primitive target features. The primitive point and line features are first extracted from a SAR image with subsequent estimation of six translation features as a function of a reference angle between features. These pair-wise features include: displacement and direction, angle and slope of a bisecting line, and orthogonal direction and orthogonal distance. To increase the robustness of an ATD/R system against camouflage and surrounding noise, the authors recommend against using properties of peaks or edges, such as the maximum brightness or area or a peak intensity which may be more susceptible to concealment tactics. Template matching is used for target classification/identification with good performance reported. In particular, this method of ATR is shown to be robust to dense target environments, partially occluded targets and to targets under camouflage.

In summary, a limited review of the literature has identified several state-of-the-art approaches to generating invariant features from SAR and MSI or HSI data, many of which incorporate some form of linear transformation.

2.2.2 Current HSI Data Feature Research

One area of current research identified in the literature is the reduction of HSI features to assess the underlying dimensionality of this data and possibly indicate optimal frequency bands on the order of those collected by MSI sensors. Because high levels of

correlation are present for neighboring spectral bands, HSI data collected for hundreds of frequency bands may be no better for classification problems than MSI data (Gat *et al.* 1996). Collection of HSI data also often produces very sparse data that can be projected into lower dimensions with minimal loss of information (Landgrebe, 1997: 24). Further, from an information theoretic viewpoint, Hughes (1968) showed for a given finite sample of data, the mean classification accuracy obtainable will theoretically decrease as the number of input features increases. The Hughes phenomenon, suggests feature engineering methods to generate a reduced subset of input variables, for use by a classification model, is highly desired.

With HSI collected at a high spectral resolution, it can be easily tailored to a desired application by combining or eliminating any number of bands to generate more desirable features. One potential method of HSI data reduction is simply binning HSI bands into wide groups to enhance the signal to noise ratio. This may be performed using different numbers of initial HSI bands or by combining discontinuous bands of HSI to generate new features within the visible and IR spectrum. Research of determining optimal frequency bands may use PCA or Factor Analysis in combination with classification model feature screening techniques in attempt to determine the underlying dimensionality of those features providing for best class separation. If optimal bands within the visible and IR spectrum are determined, a multispectral system can then be used or designed that is less expensive, produces smaller datasets and has a greater signal to noise ratio (Gat *et al.,* 1996). Thus, some current research of HSI sensor data seeks to select optimal spectral band parameters (band position and widths) to reduce noise and focus on salient classification information. In summary, with the potential of MSI or HIS

data to be used by ATR systems, the investigation of fusing highly correlated data across features must be addressed for these sensors to maximally contribute to the ID process.

2.2.3 Levels of Correlation in Radar, MSI and HSI Data

A literature review was performed in attempt to find to obtain a better understanding of the "real-world" correlation levels which may be encountered. In general, the use of MSI and HSI imagery is new to the ISR community, but is very similar to more traditional electro-optical (EO) and IR sensors, with an increased ability to collect data within smaller frequency bands. A major advantage of the MSI and HSI data imagery is the simultaneous gathering of sensor data across a full spectrum of visible and IR electromagnetic frequencies for an object of interest. This reduces registration issues and the potential uncertainty that two different sensors are observing the same object. No characterization of the correlation obtained from multiple sensors across multiple looks was found in the literature. Specifically, no published work has been found addressing potential correlation levels between radar imagery and EO, IR, MSI or HSI data, although it is suspected research of this nature is currently or has been performed but is classified and potentially proprietary to DoD contractors.

The literature does present some indications of how correlation issues are addressed for multiple looks of radar data, and that potential high levels of correlation exist within HSI data. The HSI data will inherently possess significant correlation between 'close' frequency bands, will all frequency band information collected during a single look of an object. It should be noted, that the predominant published literature for MSI and HSI data analysis is for remote sensing applications, e.g. data collected by

satellites with relatively low resolution > $10m^2$ and may not be directly applicable to an Combat ID application where a warfighter is relying on an ATR system for fire control assistance, at lower elevations and with much better resolution.

As mentioned, the open literature is relatively sparse with respect to sensor correlation observed for military applications, most likely due to the fact that publishing this information may benefit US adversaries, and thus remains classified and/or proprietary to DoD contractors. DARPA's Multisensor Exploitation Testbed (MSET) program is one example of current research in the area of sensor fusion for CID. An *ITAR restricted FOUO* analysis of MSET data performed by Young *et al*. (2001) for sensor fusion across radar and MSI data was reviewed, but does not report any measures of the correlation between features. Some preliminary results of feature level fusion using SAR and MSI data is presented, where significant improvement in probability of detection and reduction in false alarms were obtained when SAR and MSI data were fused in an algorithmic ATD architecture. While results are FOUO, they are presented for three levels of occlusion (target in open, partially occluded, and heavily occluded by trees) and demonstrate the effectiveness of sensor fusion relative to varying levels of Camouflage, Concealment and Deception (CC&D).

Initial analysis of the MSET data was limited to a subset of the collected SAR and MSI imagery. The multispectral scanner collected data in 12 channels operating in visible and infrared wavelengths between 0.4 and 10.5 micrometers. The specific wavelength bands corresponding to each channel is shown in Table 2.4. While 12 MSI channels are available, a subset of 5 channels was used by Young *et al*., (2001) including channels 3, 5, 7, 9 and 10. Use of this subset of five disjoint MSI channels is

hypothesized to help produce relatively less dependent data, and avoid potential problems of fusing the highly correlated adjacent bands.

Table 2.4 Measured Multispectral Imagery (MSI) Frequency Bands

Channel	*Band*	*Wavelength (μm)*
1	Violet/Blue	0.42-0.45
2	Blue/Green	0.45-0.51
3	**Green/Yellow**	**0.51-0.59**
4	Orange	0.58-0.62
5	**Red**	**0.61-0.66**
6	Red/NIR	0.65-0.73
7	**Near-IR**	**0.71-0.82**
8	Near-IR	0.81-0.95
9	**SWIR**	**1.60-1.80**
10	**SWIR**	**2.10-2.40**
11 (alt)	MIR	3.16-5.20
11	TIR	8.28-10.67
12	TIR	8.28-10.67

No military ATR applications of HSI data were found in the open literature; but, relatively recent research of HSI data has led to a special issue of *IEEE Transactions on Geoscience and Remote Sensing* dedicated to the analysis of hyperspectral image data. An article by Serpico and Bruzzone (2001) documents the difficulty of dealing with the spectrally close HSI bands with redundant information. Similar findings are presented in (Kumar *et al.,* 2002), who state high positive correlation is to be expected in HSI frequency bands that are in very close spectral proximity. HSI object images are constructed from observed data from one frequency channel, and are similar to one of the MSI bands presented in Table 2.4, but contain a much smaller frequency range. For

example, an IR image may be generated from channel 12 of Table 2.4, the Thermal IR

frequencies (TIR). For HSI images each channel may be subdivided into 10 or even 20

sub-channels each with its own slightly different 'picture' of the object of interest. These

HSI images in spectrally close bands would appear very similar because of the

continuous nature of observed energy emitted across the continuous time and frequency

domains. Thus, the energy emitted by a potential target and sensed by HSI frequency

bands with very similar frequencies would be correlated. As stated in (Serpico and

Bruzzone, 2001), "as hyperspectral sensors acquire images in very close spectral bands,

the resulting high-dimensional feature sets contain redundant information." Kumar *et al.*

(2002) similarly state, "the response of bands that are spectrally 'near' each other tend to

be highly correlated," and go on to note that, to generate features from the bands of HSI

data, it should first be, "partitioned into groups of highly correlated adjacent bands."

This potentially indicates a practical projection back down into MSI size frequency

bands, but this projection would now be optimized for the pattern recognition task at

hand.

   With the possibility of obtaining high levels of correlation between IR frequency

ranges, Thomas (1994) used correlation values > 0.99 for adjacent MSI IR spectral bands

to determine potential targets from background clutter. This high correlation in IR

spectral bands was found to correspond to man-made objects, while significantly lower

correlation levels were obtained for natural clutter. Thus, while and HSI data sets were

not found in the open literature with reported values of correlation, the literature does

suggest high levels of positive correlation would be expected between 'close' bands of

spectral frequency data. This may be especially true for man-made targets, and exploiting this high level of correlation may help for some classification efforts.

Since radar has a longer history of use for ATR and other military uses, more literature is available, but still appears relatively 'filtered' as to not give away classified capabilities of systems. The following discussion summarizes some of the findings reported in the literature. It is not exhaustive, yet it does provide good insight as to the expected levels of correlation in multi-look SAR data and for the within feature correlation that would be observed for multiple features given by one radar sensor. In research by Chitroub *et al.* (2002), for the statistical characterization and modeling of SAR images, the authors point out that multi-looks of SAR imagery are typically used to reduce noise, and multiple images result in only a single target image. In performing such noise reduction and SAR image fusion, the authors note that if $k = k_r k_a$ pixels are averaged where k_r denotes the range direction and k_a denotes the azimuth direction, then the effective number of looks is somewhat smaller due to dependence of neighboring pixels. Related research by Gierull and Sikaneta (2002) estimate the effective number of looks in interferometric SAR data, and document adjacent pixel information obtained is statistically dependent due to the filtering process and go on to state, "the number of looks is usually smaller than the number of samples averaged." Hauter et *al.* (1997) also reported in their research of polarimetic fusion for SAR target classification, that multiple SAR imagery polarized channels are, "inherently more correlated than the sources from independent sensors." Although, they do not indicate the actual levels of correlation observed between these within SAR data polarized imagery features. Costantini *et al.* (1997) look to obtain a better knowledge of an object through fusion of SAR images by

39

fusion of different resolution SAR images. This research acknowledges the inherent

dependences between the multi-looks obtained at differing resolutions and generates a

single fused image via a least square deviation from the finest resolution image, subject

to constraints obtained from courser resolution images. Unfortunately, the process is

demonstrated for generated data and does not indicate the levels of correlation that may

be observed between true SAR images. Lee *et al.* (1994) also address SAR correlation

issues for the intensity and phase statistics of multilook polarimetric and interferometric

SAR imagery. In this research, they rigorously document how multilook processing

reduces statistical variation when combining multiple SAR images to produce a single

image of higher resolution. Some theoretical examples are presented for a correlation

level set at $\rho = 0.5$. Unfortunately, these papers are primarily theoretical, void of

observed within radar sensor correlation levels, but the EE community appears to be

addressing the SAR within sensor correlation issue by reducing the noise and producing a

better single estimate SAR image from multiple looks.

2.2.4 Summary of Sensor Data Correlation Issues for ATR

While features derived from passive visual or thermal sensors or reflected radar

energy each contain different noise sources and may be statistically independent, multiple

looks by a single ATR system across the time continuum may yield significantly

correlated scores (Jacques, 1998). Some research addresses the case of fusing correlated

probabilities (O'Brien, 1998, 1999), but a difficulty arises if a fusion algorithm assumes

independence is implemented for real-time multi-look ATR. Yet, ATR applications for

combat ID may require additional information to increase confidence after obtaining

"Non-declaration" for an object of interest (Dept. of the AF, 1998, 1999, 2000). Real time re-looks by a sensor in close temporal proximity for the same object may be the only source of new target information. These multiple looks are hypothesized to have high levels of positive correlation and may provide relatively little new information about the object. Literature from the radar community (Chitroub *et al.*, 2002; Costantini *et al.*, 1997; Lee *et al.,* 1994) indicates high correlation levels are indeed expected between SAR imagery obtained from continuous re-looks of an area. Current image processing techniques use these multiple correlated looks to refine a single image by reducing noise as additional data are obtained. While this SAR imagery refinement is primarily done for visual interpretation and methods are not presented to make class declarations, they suggest a basic framework for dealing with temporally collected sensor data. Similar to SAR image refinement, as correlated temporal information is gathered, ATR may benefit from algorithms designed to update and refine class estimates based on obtaining new, albeit correlated, information. Thus, a primary research goal is the investigation of fusion to obtain optimal class declarations when correlated input data is used.

2.3 Fusion Methodologies

To perform fusion at the various levels, numerous quantitative techniques are available. As an emerging field of research, the data fusion community does not uniformly agree as to which fusion method is necessarily best for a given application (Hall and Llinas, 2001). For example, each fusion algorithm may have its own particular limitations, challenges, and advantages for use in a given situation. Hall and Llinas (1997) list current challenges for JDL Level 1 fusion techniques to include: addressing

41

correlation and maneuvering target problems for the complex multisensor, multi-target case with co-dependent sensor observations, and the need to integrate identity and kinematics data. Other indicated challenges for object identification include difficulties created by dense target environments, rapid target movement, complex signal propagation and background clutter. Thus, research aimed at understanding the impact of correlated input data on given fusion techniques for feature or decision level fusion of an object of interest is desired.

Review of the recent literature has identified several methodologies to perform fusion in the attempt to refine a class estimate for an individual object under investigation. If feature level fusion is being performed, then a feature vector representation of the object may be used by any standard pattern recognition algorithm to obtain a class estimate (Hall and Llinas, 1997, 2001; Klein, 2004). It is assumed that the feature vector is comprised of features from at least two different sensors, or from multiple looks by the same sensor. Hall and Llinas (1997) indicate methods for estimating an object's identity are dominated by feature based approaches, which include the use of neural networks, cluster algorithms, and other pattern recognition methods. Pattern recognition techniques may include template based approaches and other statistical and probabilistic methods. If the individual sensor data if first refined to generate a class label, then Boolean voting logic is a standard fusion methodology to determine a single class estimate (Varshney, 1997; Klein, 2004; Waltz, 1990). Rule-based expert systems are also identified by Hall and Llinas (1997), with the addition of fuzzy logic and neural networks for multisensor fusion at this slightly higher level. The

voting logic may also be determined optimally by use of probabilistic methods, such as those presented by (Ralston, 1999) and (Haspert, 2000).

The next two subsections of fusion methodology include a discussion on Boolean logic fusion methods and the use of neural networks for fusion. These sections will support the experiments undertaken in future chapters of the document, where research of fusion with unknown class declarations in the presence of correlated input data is performed.

2.3.1 Boolean Fusion Methodologies

One method of combining output labels of different identification systems is to use Boolean rules. One such rule may conclude a Hostile target is present if and only if all of the sensor labels indicate the target is a "Hostile." This rule may simply be referred to as the AND rule. Another simple Boolean rule is for the system to conclude an object under investigation is a "Hostile" target if any of the sensors being fused label it as a "Hostile." This rule may be called a simple OR rule. With more than two sensors fused to generate a final output label, many combinations of simple Boolean logic are possible. For the case of fusing K sensors with two output labels, 2^{2^K} Boolean fusion rules may be obtained (Haspert, 2000).

An illustration of potential Boolean logic fusion rules is depicted in Figure 2.9 for the use of three Sensors (S^A, S^B and S^C). These diagrams are similar to those presented by Liggins (2001). The labeled areas of the Venn diagrams show seven mutually exclusive sets for the declaration of a potential target as "Hostile." Each set is identified by a two-class output label associated with sensors A, B and C. Thus, $2^3 = 8$ different

sensor labels associated with the three sensor outputs may be obtained for any given assessment of a potential target. Further, as noted by Haspert (2000), each of these sensor states may be included in a final "Hostile" declaration rule, and a logical OR combination of these sensor output states results in $2^8 = 256$ different feasible logical fusion rules. An eighth combined sensor output state may be added to each of the five Venn diagrams, where no sensor indicates the target is "Hostile." This completes the feasible sensor output states for each of the Boolean fusion rule presented in Figure 2.9. The grey areas show where a positive declaration of a "Hostile" target would result for each of the Boolean fusion rules. Logical AND and OR rules follow from the previous discussion. Majority Vote logic requires a majority of the sensors to declare the target as "Hostile." Thus, 2 or more "Hostile" labels are required for a three sensor suite to declare "Hostile." Majority Vote logic is perhaps the oldest strategy for decision making with roots tracing back to the era of ancient Greek city states (Kuncheva, 2004). The logic associated with 'sensor corroboration' requires sensor A to declare a target as "Hostile" and to corroborate this label with either sensor B or C. As described by Hill (2003), such a fusion rule may be appropriate when sensors perform different functions. For example, sensor A may represent a Moving Target Indicator (MTI) with good detection rates, but low resolution; while sensors B and C may represent cued sensors with high resolution and good target discrimination The final Boolean rule shown is 'sensor dominance.' This logic may be appropriate if sensor A is known to perform much better than sensors B and C. Thus, sensor A may have high confidence and result in a fused "Hostile" label regardless of the labels from sensors B and C. With lower accuracy, when sensor A does not indicate "Hostile," sensors B and C may only yield a

44

fused "Hostile" label if they agree. Other Boolean fusion rules may be generated for a given ensemble of sensors and classification task at hand.
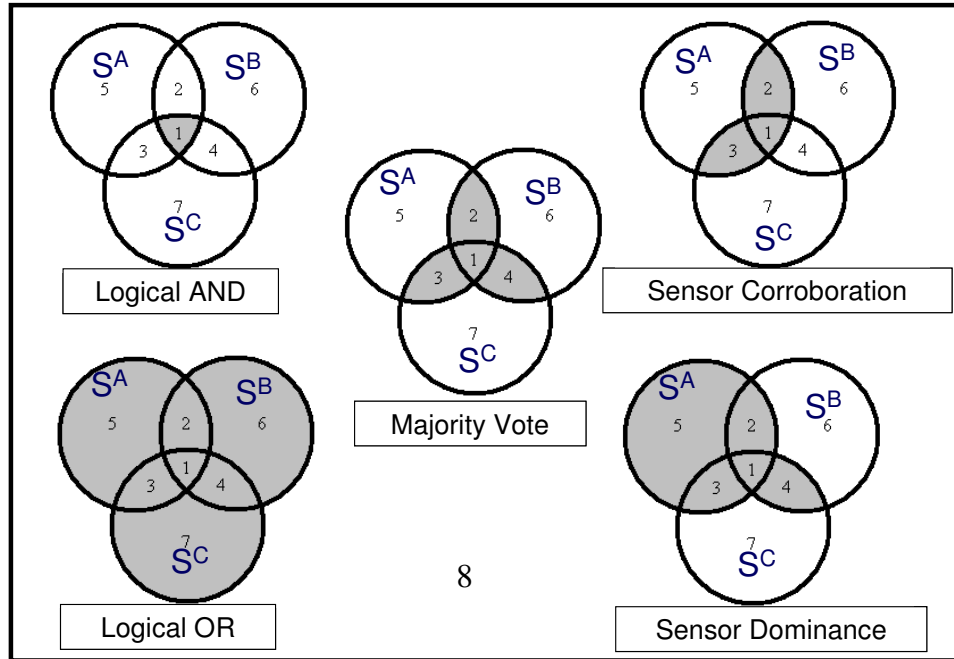
Figure 2.9 Examples of Various Boolean Fusion Rules with Venn Diagrams

Overall, current sensor and classifier fusion texts (Klein, 2004; Varshney, 1997; Kuncheva, 2004) provide significant discussion of these simple Boolean fusion rules. This focus in current fusion texts provides evidence as to their general use and acceptance as easy to implement fusion rules.

While easy to implement, Boolean logic has some significant limitations for the fusion of decision labels. Robinson and Aboutalib (1989) provide a mathematical proof showing Boolean fusion for decision labels is suboptimal for the fusion of two or more sensors when the sensors are not independent. This proof uses a cost function and known priors. In their proof, they show an optimal declaration threshold is a function of the joint

pdf of the combined sensor data. They conclude, "decision level fusion is in general suboptimal to feature level fusion in terms of classification performance." Robinson and Aboutalib (1989) go on to state, "…in order to achieve a global optimum decision, the classifier of sensor S1 should know the entire decision process for the classifier of sensor S2." If each sensor is optimized independently, as may be the case when sensors are initially developed and fielded as independent ISR assets, the fusion of the two sensors will in general be suboptimal to their combined potential. Robinson and Aboutalib (1989) also note Boolean fusion logic selection is a key to good performance. For example, if the thresholds are optimally tuned for each sensor and a particular Boolean rule, a different Boolean rule may provide better system results.

Overall, Boolean logic is a common method to perform fusion at the decision level. Yet, selecting an optimal Boolean fusion rule and tuning each individual sensor to achieve an optimal identification system across potentially correlated, dependent information remains a challenge.

2.3.1.1 Optimal Boolean Fusion Methodologies

To determine the optimal Boolean fusion rule associated with the fusion of K sensors, Ralston (1999) and Haspert (2000) use an Identification System Operating Characteristic (ISOC) curve to determine the optimal system performance associated with all potential Boolean fusion rules. This fusion method determines the optimal fusion rule when each sensor uses a set decision label threshold. The determination of the optimal Boolean logic is obtained from a novel algorithm using a likelihood ratio associated with each of the mutually exclusive and collectively exhaustive sensor label output states. The

46

likelihood ratio is generated as the ratio of probabilities associated with a desired true class, compared to all other classes, for each of the unique sensor label output states. For this algorithm, sensors are assumed to be independent and a cost associated with each type of misclassification error is required to determine the optimal point on the ISOC curve which is associated with an optimal Boolean fusion rule (Haspert, 2000). An approach to obtaining "Non-declaration" labels using a minimum cost function and estimated misclassification costs for K sensors with any number of output labels is also presented by Haspert (2000).

While Ralston (1999) and Haspert (2000) seek to determine the optimal fusion rule given sensors with set thresholds, Oxley and Bauer (2002) determine the optimal thresholds for a predetermined Boolean fusion rule across conservative to aggressive declaration labels. This novel ROC fusion methodology provides an analytical means to obtain the best fused ROC curve for logical AND and OR rules; yet, does so under the assumption of independent sensors. While Oxley and Bauer (2002) present an example of their ROC fusion for a two-class problem with two sensors, research by Hill (2003) shows the ROC fusion using AND and OR logic may be extended to include any number of classifiers and output labels. While conceivable, the inclusion of the third output label for "Non-declarations" does not appear to be a readily practicable extension of the ROC fusion. Thus, the inclusion of "Non-declarations" and potentially more than two input classes or output labels may warrant additional research to extend the current ROC fusion methodologies.

Recent research by Storm (2003), Leap (2004) and Clemans (2004) compared use of ISOC and ROC within fusion using a logical OR rule across various sensor correlation levels. While these methods assume independent sensor data, in general, the ISOC and ROC within fusion methods were found to be robust to sensor correlation (Storm *et al*., 2003; Leap *et al.*, 2004). These fusion rules did not gain significant performance improvement above the best fused sensor, but were found to mitigate the risk associated with the potential use of a poor sensor in the available sensor ensemble. This conclusion agrees with Boolean fusion research by Dasarathy (2004), where different distance measures were used by fused classification algorithms under given Boolean logic and a risk mitigation effect was also observed. While both ISOC and ROC fusion methods were robust to correlation, fusion using neural networks, without an assumption of independent input data, was found to perform better in the presence of induced sensor correlation (Leap *et al.*, 2004).

2.3.2 Artificial Neural Networks for Sensor Fusion

As identified in (Hall and Llinas, 1997) neural networks have been successfully employed for feature and decision level fusion. As identified in the next three sections, artificial neural networks may be sub-divided. Subdivisions may include feed forward multilayer perceptrons (FF MLPs), recurrent neural networks (RNNs) and the use of radial basis neural networks (RBNNs). A section is then devoted to some current methods of feature selection as applicable to MLP ANNs and RNNs.

2.3.2.1 Feed Forward Multilayer Perceptron ANNs

To perform fusion, neural network models may be selected for several reasons. Figure 2.10 represents a fully connected multilayer perceptron (MLP) ANN. While often viewed as a black box, these models are theoretically capable to perform any mathematical mapping from an input to output space with any desired degree of accuracy provided the number of hidden nodes is sufficiently large enough (Hornik *et al.*, 1989, 1990). MLP ANNs offer a nonparametric approach to generate a mapping for input data to a desired output space, with no assumed distribution or independence requirement between variables. In addition, ANNs learn and may even adapt to new training data to obtain optimal parameter settings. Some drawbacks of ANNs include the expense of an available training data set fully representative of desired input and output spaces, along with the computational complexity of the training process, and a lack of decision insight. Yet, because they do not require assumptions of the input data structure, they are fully capable of integrating sensor features, estimated class probabilities and binary class labels, each of which may contain significant correlation between and across features. Thus, ANNs allow for flexible sensor fusion via a one big net model.
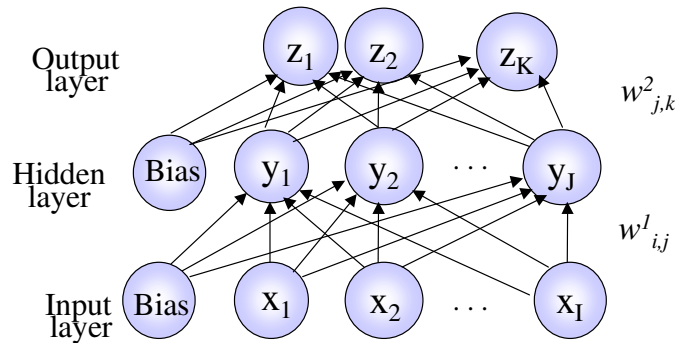
Figure 2.10 Multilayer Perceptron (MLP) Artificial Neural Network (ANN)

The output from such a MLP ANN for the *n*th input vector (z^n) can be computed as:

$$k^{\text{th}} \text{ neural network output} = z_k^n = f\left(\sum_{j=1}^{J} w^2_{j,k} x_j^1\right) \tag{2-4}$$

where,
- *J* is the number of hidden nodes
- $f(a) = 1/(1 + e^{-a})$ is a typical sigmoidal activation function
- $w^2_{j,k}$ is the weight from hidden node *j* to output node *k*
- x_0^1 is the hidden layer bias term and is set equal to 1
- $x_j^1 = f(\Sigma\, w^1_{i,j} x_i^n)$ is the output of hidden node *j* and is summed from *i* = 1 to M
- *M* is the number of input features
- $w^1_{i,j}$ is the weight from input node *i* to hidden node *j*
- x_0^n is the input layer bias term and is set equal to 1
- x_i^n is the *i*th input feature of the *n*th input vector

MLP ANNs are typically trained using a nonlinear optimization algorithm in which the error gradient is estimated from the current model parameters for training data with known desired output. A standard approach for training ANNs uses the error in an iterative fashion to adjust the connection weights of the ANN until a stopping criteria has been reached (Bishop, 1995). These algorithms are commonly referred to as backpropagation training algorithms. Additional background for FF MLP ANNs may be found within (Looney, 1997) and (Bishop, 1995).

2.3.2.2 Recurrent Neural Networks (RNNs)

While an ANN with proper architecture has been proven capable of universal function approximation, it may only explicitly model temporal relations in static time. Since a strong temporal component may be hypothesized for many pattern recognition applications, such as financial forecasting or target tracking and identification, an ANN model may be desired that allows for the implicit encoding of time. The Elman RNN

50

includes internal feedback and the ability to model temporal patterns (Elman, 1990).

With an architecture similar to ANNs, an Elman RNN adds internal feedback to the

model with each hidden layer output from time *t* included as input model at time *t+1*.

Figure 2.11 shows an Elman RNN, with I input features, J context nodes, J hidden nodes

and K outputs, where feedback is accomplished by the context nodes in Figure 2.11.
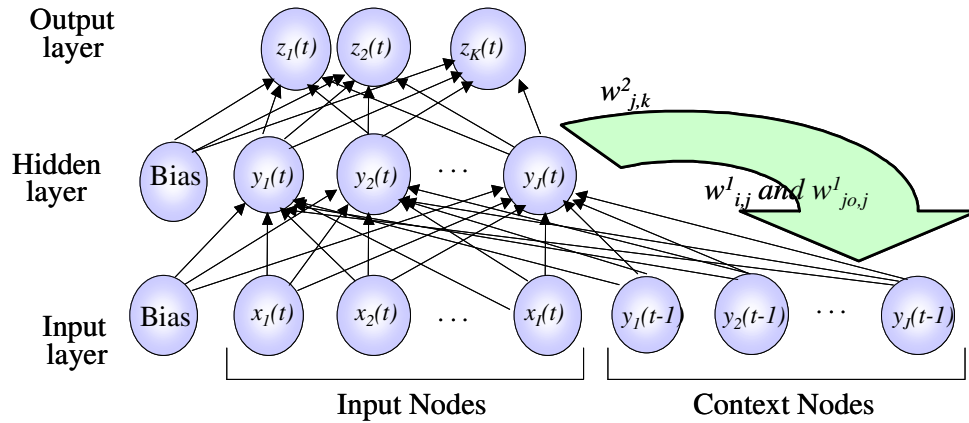
Figure 2.11 Elman Recurrent Neural Network (RNN)

Similar to MLP ANNs, Elman RNN hidden and output layer perceptrons have

associated activation functions, typically nonlinear sigmoid, hyperbolic tangent, or linear

depending on the application. The hidden layer output is included as context node input

for the next data observation to facilitate a dynamic memory for temporal patterns. By

having internal feedback, the Elman RNN implicitly models temporal patterns (Elman

1990) and has been proven to have the computational power of any finite state machine

given a sufficiently large enough architecture (Giles & Omlin, 2001; Kremer, 1995).

Further, the Elman RNN has increased modeling flexibility over another common RNN,

the Jordan RNN, which uses the final network output from time *t* as context node input at

51

time $t+1$ (Calvert and Kremer, 2001). Further details for the training of RNNs along with use of a similar time delayed neural network (TDNNs) to model temporal patterns may be found within (Kolen and Kremer, 2001).

2.3.2.3 Radial Basis Function (RBF) Neural Networks

Radial Basis Function (RBF) neural networks (NN) are commonly used neural networks to perform classification tasks. Unlike standard MLP ANNs and RNNs with sigmoidal or linear activation functions, RBF NNs use activation functions with an exponential neuron response which is not supported by biological neural systems (Wasserman and Nostrand, 1993). They may require more neurons to perform a given classification task as compared to FF MLP ANNs, but because they may be trained using deterministic methods, the associated training time may be far less than that of MLP ANNs (Bishop, 1995). RBF neural networks may be designed as exact interpolation functions with an activation (basis) function associated with every training exemplar. Perhaps the biggest difference between RBF NN and MLP ANNs is use of activation functions with local vs. global influence. A typical basis function used by these networks is,

$$f(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - x_i\|^2}{2\sigma^2}\right), \tag{2-5}$$

where σ is the spread or variance associated with each basis function, x_i is the location of each of i basis functions and $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is an input vector of dimension n. The spread may be adjusted, where larger values have more global influence, and smaller values limit influence and cause these functions to behave in a nearest neighbor fashion

(Demuth and Beale, 1998). The RBF NNs use a set of distributed basis functions each with a radially uniform symmetric local response. A training algorithm then adjusts a weighted response for each of the basis functions to estimate the underlying function of the input data (Bishop, 1995). A basis function may be used for every training exemplar, or may be added in an iterative manner until a desired level of performance is achieved (Demuth and Beale, 1998). Two common neural networks using radial basis functions include the generalized regression neural network (GRNN) and the probabilistic neural network (PNN). The general regression neural network (GRNN) is a probabilistic neural net (PNN) augmented by a normalizing factor (Specht, 1991) and may be used for function approximation for arbitrary non-linear functions (Specht, 1991).

The probabilistic neural net (PNN) is an ANN implementation of the Parzen windows method. The output is a weighted sum of all training features, where the weighting is exponential according to the distance to given training point (Specht, 1990). The PNN is based upon work performed in the 60's by Specht, but due to computational limitations has only recently been implemented for a variety of classification problems (Wasserman and Nostrand, 1993). The PNN offers many advantages for classification compared to a FF MLP ANN. These advantages include rapid training performed in one pass of the training data, robustness to noise, and guaranteed convergence to Bayes-optimal decision boundaries given enough training data (Specht, 1990). One disadvantage is a large computational storage requirement by the PNN, with a basis function included for every training exemplar. As presented by Wasserman and Nostrand (1993: 52), Figure 2.12 shows the architecture of a PNN for a two-class decision.
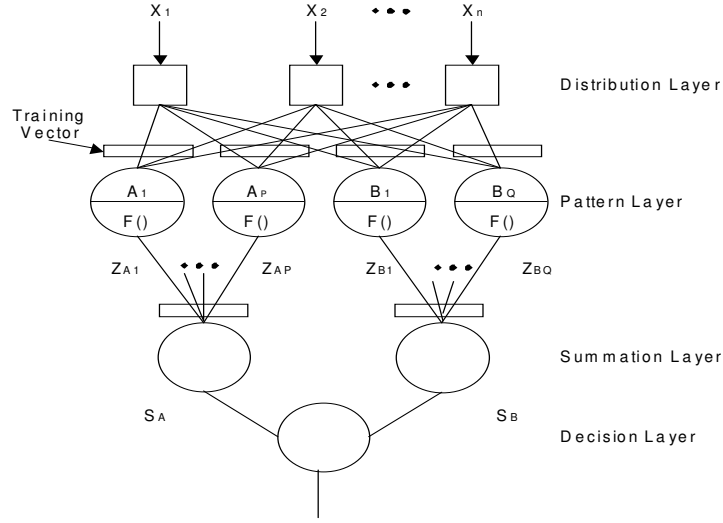
Figure 2.12 Probabilistic Neural Network (Wasserman and Nostrand, 1993: 52).

Starting at the top of Figure 2.12, a normalized input vector $X = (x_1, x_2, ..., x_n)^T$ with n-input features is presented to the PNN. The distribution layer is a connection point and no calculations are performed (Wasserman and Nostrand, 1993). The set of weights for the pattern layer neurons are equivalent to the components of the training input vectors, and are grouped by the known labels. Each pattern layer sums the weighted inputs from the distribution layer neurons and applies the nonlinear basis function to produce output Z_{ci}, where the appropriate activation function is given as,

$$Z_{ci} = \exp\left[\frac{\left(X_{Ri}^t X_i - 1 \right)}{\sigma^2} \right] . \tag{2-6}$$

For this calculation, $X_{Ri} = (x_{R1}, x_{R2}, ..., x_{Rn})^T$ is a training exemplar, where i is an index for the number of training exemplars, R indicates a training vector, and c denotes the class (Wasserman and Nostrand, 1993). The summation layer sums all Z_{ci} for each class. The output of this layer for class c is given as follows (Wasserman and Nostrand, 1993),

$$S_c = \sum_{i=1} \exp\left[\frac{\left(X^t X_{Ri} - 1\right)}{\sigma^2}\right].$$
(2-7)

The decision layer compares S_c for all classes and assigns the input vector to the class with the largest corresponding S_c. In summary, the PNN assigns a new input exemplar to the decision label with the largest probability of membership. A PNN can model any number of classes, and the probabilities of class membership may be obtained from the values associated with the class summation layers.

Overall, the PNN has been found to be an effective method for the fusion of multivariate Gaussian data across different correlation structures. These correlation structures may represent the correlation between sensor features as presented by research performed by Storm (2003), Clemans (2004), Leap (2004) and Mindrup (2005). In each of these investigations, PNN fusion was found to be equivalent or preferred to Boolean fusion techniques where decision labels were forced. A "Non-declaration" implementation was applied by Mindrup (2005), in which similar performance was obtained by use of a PNN for fusion across correlation structures as compared to a preferred 'optimal' Boolean fusion rule identified by a heuristic approach.

2.3.2.4 Feature Selection for Artificial Neural Networks

While properly configured neural networks can approximate any function, they are dependent on the quality of input data from which they learn or adjust their weight parameters. For statistical pattern recognition applications, it is well documented that too many features may decrease classification performance, since the number of observations must grow exponentially as the number of features increases to maintain the same

sampling density. This "curse of dimensionality" (Bishop, 1995) phenomenon parallels

findings by Hughes (1968) and suggests feature reduction should be performed to

improve results when limited data observations with sparse, high-dimensional input data

are collected (Jackson and Landgrebe, 2001).

In order to improve a neural network model's accuracy, a reduced feature set

representative of the underlying salient input feature space is desired. Feature

engineering includes the extraction of salient features by finding a mapping to project P-

dimensional input data onto M-dimensional space where $M < P$. A review was first

undertaken to identify methodologies for RNN feature selection, with feature selection

defined as a special case of feature extraction whereby the M-dimensional space

corresponds to a subset of P collected potential input features. Research by Greene

(1998), Greene *et al.* (1997, 2000), Utans *et al.* (1995) and Moody (1998) use RNN

saliency metrics based on model weights and output error associated with input features.

Since limited RNN saliency methods were identified, a broader review was undertaken of

recent ANN feature selection techniques that may be applicable to RNNs. Similar to the

methods of Greene *et al.* and Moody *et al.*, other recent research is divided between

techniques using ANN model weights (Castellano and Fanelli, 2000; Lazzerini and

Marcelloni, 2002; Mak and Blanning, 1998) or model output (Feraud and Clerot, 2002;

Kwak and Choi, 2002; Piramuthu, 1999; Verikas and Bacauskiene, 2002; Zhang and Sun,

2002) with entropy measures associated with model output used by Piramuthu (1999) and

Verikas & Bacauskiene (2002) and a tabu search based on observed model output

employed by Zhang & Sun (2002).

Two representative feature screening techniques for use with ANNs inclusive of Elman RNNs are the Signal-to-Noise Ratio (SNR) feature screening introduced for ANN use by Bauer *et al*. (2000) and first applied to an Elman RNN by Greene (1998) and Sensitivity Based Pruning (SBP) as developed by Moody and presented in (Moody 1998; Utans *et al.* 1995) for general neural network use. These methods represent proven network parameter and output based saliency measures.

The SNR saliency measure is computed using the first layer weights of a trained RNN as,

$$SNR_i = 10 \cdot \log_{base10}\left(\frac{\sum_{j=1}^{J}\left(w_{i,j}^1\right)^2}{\sum_{j=1}^{J}\left(w_{N,j}^1\right)^2} \right), \tag{2-8}$$

where SNR_i is the value of the SNR saliency measure for feature i, J is the number of hidden nodes, $w_{i,j}^1$ is the first layer weight from input node i to hidden node j, and $w_{N,j}^1$ is the first layer weight from an injected noise input node N to hidden node j. All feature inputs, including the randomly generated noise, are normalized. The scaled logarithmic transformation of the ratio converts the saliency measure to a decibel scale. The idea behind the SNR saliency measure is relevant features will have a SNR_i significantly greater than 0, while noise-like features will have a SNR_i saliency value close to or less than 0. The SNR saliency measure provides a way to rank order features from most relevant to least relevant and has been shown to be is statistically equivalent (Greene 1998) to Ruck's partial derivative based saliency measure (Ruck *et al.,* 1990) and Tarr's weight based saliency measure (Tarr 1991) for ANNs. In addition, SNR feature selection has been successfully employed for fusion of correlated features derived from multiple

57

sensors with an ANN (Laine *et al.,* 2002; Greene, 1998), and feasibility has been

demonstrated for time delayed neural nets (TDNNs) and RNNs by Greene (1998).

Overall, the use of weight based saliency measures are well documented in the literature

and have Bayesian foundations as shown by Priddy *et al.* (1993), who demonstrate

effective weight based Bayesian selection of salient forward looking infrared (FLIR)

features for a combat ID application.

Like the SNR saliency measure, Sensitivity Based Pruning (SBP) associates a

saliency measure to each input feature. The sensitivity measure S_i for each of *i* features is

calculated by assessing the effect of replacing each input feature with the mean value of

that feature (Moody 1998; Utans *et al.* 1995) and can be calculated once a classification

model is trained as,

$$S_i = MSE\left(\overline{x}_i\right) - MSE\left(x_{ip}\right) , \tag{2-9}$$

where $MSE(x_{ip})$ is the mean square error of the RNN for all *p* exemplars and $MSE(\overline{x}_i)$ is

the MSE when an average value is assigned to input feature *i*. If using the average value

of a feature for all exemplars increases the *MSE*, S_i will be positive and considered

salient, and the feature associated with the largest value of S_i is deemed the most salient

feature. Thus, S_i values can be used to rank order the relative saliency of input features

for any classification model. If the input features have been normalized with a mean of

zero prior to training the network, S_i can be computed simply by evaluating the trained

ANN and setting each input feature to 0. Applications of SBP by Moody (1998) for

continuous financial time series prediction compute S_i for the training data, iteratively

train and remove a feature from the network, then seek to select a parsimonious subset of

features that minimizes prediction risk of a forecast. The SBP metric may be implemented similar to the SNR measure, with S_i calculated from the training-test set to provide a measure of an ANN's ability to generalize well to new patterns and CA is calculated as:

$$CA = \frac{\text{Number exemplars classified correctly}}{\text{Total number of exemplars}}.$$ (2-10)

Therefore, instead of prediction risk, CA may be used to determine a final set of parsimonious salient features to retain for effective discrimination between two or more target classes.

SNR and SBP screening methods use the saliency metrics as previously defined to obtain parsimonious sets of salient features while retaining good classification accuracy as features are removed stepwise backward. Following are the steps to determine reduced salient feature sets:

1. Introduce a Uniform (0,1) noise feature, x_N, to the initial features (for SNR only).

2. Preprocess all features with mean zero and unit variance.

3. Initialize the RNN weights via the Nguyen & Widrow (1990) method.

4. Initialize input layer weights as uniform [-0.01, 0.01] (for SNR only).

5. Train the RNN and retain the weights that minimize the MSE of the test set.

6. Identify the least salient feature with the lowest SNR_i or S_i saliency metric.

7. Remove the least salient feature from the ANN.

8. Repeat steps 5, 6, and 7 until all features in the initial set have been removed.

9. Plot the training-test set classification accuracy (CA) as individual features are removed.

10. Retain the first feature whose removal caused a significant decrease in the training-test set CA, as well as all features removed after the first salient feature was identified.

Both screening methods seek to find a parsimonious set of input features representative of the underlying input feature space dimensionality. This is accomplished by reducing the features used to discriminate between classes, such as removing one of two highly correlated input features. In previous research the SNR screening method has produced a reduced number of input features for an ANN while maintaining or improving classification accuracy for independent validation sets (Bauer *et al.,* 2000; Greene *et al.*, 2000; Laine *et al.*, 2002).

2.3.3 Other Methods for Sensor Fusion

As mentioned in the introduction to this section of fusion methods, numerous other quantitative methods are available for fusion at the feature or decision level for an individual object under investigation. Some of these methods include Bayesian Techniques, parametric statistical modeling, non-parametric techniques, support vector machines, Hidden Markov Models (HMMs), etc. Current sensor fusion texts by Hall and Llinas (2001), Klein (2004), and Varshney (1997) and a classifier fusion text by Kuncheva (2004) provide overviews of many of the quantitative techniques that may be applied for fusion at this level. In addition, from their *IEEE Proceedings* article, "Introduction to Multisensor Data Fusion," Hall and Llinas (1997), indicate any pattern recognition techniques may be applicable for feature level fusion. Thus, all methodologies for performing pattern recognition of an object may also be used to perform fusion if the input data for the algorithm is derived from multiple sensors. An

overview of pattern recognition techniques may be found in texts by Duda *et al.* (2001) and Fukunaga (1990).

2.4 Measures of Performance for Classification Algorithms

A review of the literature indicates most classifier metrics do not provide an efficient framework for optimization of conservative to aggressive decision strategies when more than two output labels are possible for a fusion system. While the traditional ROC curve does facilitate optimization across decision thresholds, it is only applicable for a two-class assignment problem with forced decisions (Alsing and Bauer, 1998). This research seeks to extend the use of ROC like performance indicators inclusive of "Non-declarations." A limiting component to most of the metrics available is that only a single metric is reported for a given classification system and comparisons between systems are then made based on a single set of thresholds. These thresholds or parameters may have been chosen optimally for each sensor individually, based on a particular test data set, but may not be optimal for the system as a whole (Robinson and Aboutalib, 1990). For example, recent research by Haspert (2000), Varner (2002) and Dasarathy (2003, 2000b) all provide a framework for non-forced decisions, but do not provide for an optimization of the Non-declaration thresholds associated with each sensor.

While the literature is dominated by metrics to assess ATR systems performing a two-class decision, many of these are not applicable to a three or more output labels. Recent research performed at AFIT provides many sources documenting potential methods for the comparison of competing classification algorithms. Included in this list are technical reports by Alsing and Bauer (1998 and 1999) along with the literature

review and subsequent research of ROC curve metrics and use of a multinomial selection procedure (MSP) documented by Alsing (2000). A more recent literature review of current ATR evaluation techniques is included in (Bassham, 2002). As presented in Section 2.3.2 "Automatic Target Recognition Performance Measures" (Bassham, 2002), measures of classifier evaluation include the following visual techniques: confusion matrices, error-reject curves, error histograms and classification trees. Statistical techniques are also summarized by Bassham (2002) and include: confidence intervals, hypothesis testing, ROC curve performance measures, the multinomial selection procedure, linear goal programming, and decision analysis. From the classification metrics above, further discussion will follow for the potential use of classification accuracy (CA) as related to confusion matrices (CMs), and ROC curves. A limited review of fuzzy logic, Dempster-Schafer analysis, the multinomial selection procedure (MSP), linear goal programming (LGP), and decision analysis (DA) is also presented. All of these methods may be applicable to the required trichotomous ATR decision.

In addition to these measures of classifier performance accuracy, Blasch *et al.* (2004) suggest other measures of performance should also be included for a fusion system. They state, "it is important to develop metrics as part of a test and evaluation strategy," and suggest, "a minimum set should include feasible metrics of accuracy, confidence, throughput, timeliness and cost" (Blasch *et. al*, 2004). Thus, while the classification performance of an ATR system is important, the temporal and monetary costs along with system confidence and efficiency are important as well.

As mentioned, many performance measures require misclassification cost and other information to determine optimal "Non-declarations." For example, Ralston (1999)

presents an approach for the Boolean fusion of labels inclusive of "Non-declarations."

His strategy is premised on prior knowledge, including prior probabilities of true class

membership and decision maker costs associated with all possible decisions. Thus, a cost

associated with each type of correct and incorrect classification along with the prior

probabilities and likelihoods of class membership must be specified (Ralston, 1999;

Haspert, 2000). As pointed out by Mahler (2001), while many algorithms claim to be

Bayes-optimal, they may be incorrectly doing so since the true likelihood ratios to be

encountered may not be sufficiently characterized. Finally, when Non-declaration labels

are implemented for classification systems, use of standard performance metrics tend to

just report the percentage of objects correctly, incorrectly, or rejected for classification.

These values may be presented in a confusion matrix (CM). As the number of potential

class labels increases and if a parameter associated with "Non-declarations" is allowed to

be adjusted, visual analysis to compare CMs, or to perform confidence interval testing of

just a handful of the reported accuracies would quickly become overwhelming. As a

research goal, a ROC-like metric is desired for ATR applications where "Target," "Non-

target" and "Non-declaration" are valid outputs.

2.4.1 Confusion Matrices (CMs)

One limitation to CMs is that they represent the classification accuracy and the

misclassifications obtained when the classifier uses set rules or decision thresholds, often

at a Bayes optimal point. With CMs, each object is uniquely labeled into one of any j-

output labels. The matrix can then be examined to see where misclassifications are most

likely to occur. Within each cell of the CM the number of correct classifications and/or

the associated percentage of classifications are included. The confusion matrix cells can be used to estimate the probability of true Target detects and the percentage of Friends misclassified as Targets, which are the two measures needed to produce a standard ROC curve. A ROC curve could be generated by varying a decision threshold between conservative to aggressive parameter values to obtain a sequence of points used to estimate the ROC curve associated with a two-class pattern recognition algorithm. Figure 2.13 extends standard "Target" and "Non-target" output labels to include "Non-declarations." Visual analysis of CMs like this, or with an extended number of true classes and output labels, as seen in Figure 2.14, may provide insight to an analyst to compare competing classification systems and to identify where misclassifications are likely to occur. This may help facilitate determining what classifier parameters may be adjusted to produce more desirable results. Summary measures to assist in the evaluation of CMs are presented by Ross *et al*. (2002) with a general discussion for ATR confusion matrix evaluation.

If training data sets are fairly well balanced and if an adequate number of training examples are available, then many classification algorithms, inclusive of neural networks will train and approach a Bayes optimal error rate (Bishop, 1995). If competing models are all trained approaching the optimal error rates, visual comparison of CM elements may help discriminate between classifiers or identify significant deficiencies. Yet, a warfighter may not be interested in the Bayes optimal values for an ATR system since the cost of certain misclassifications and prior ratios may change depending on the specific mission.

Classifier "Labels"

True Classes		"Target"	"Friend"	"Non-declaration"	Totals
	Target	# Targets labeled "Target"	# Targets labeled "Friend"	# Targets labeled "Unknown"	# Targets evaluated
	Friend	# Friends labeled "Target"	# Friends labeled "Friend"	# Friends labeled "Unknown"	# Friends evaluated

Figure 2.13 Sample Confusion Matrix with Rejection (Unique Values are Typical

for Each Threshold, θ)

When performing confusion matrix analysis, the warfighter is interested in good horizontal classification accuracy as reported by most research efforts, but is more concerned with a vertical analysis of the output labels. The vertical analysis of a confusion matrix is conditioned on the output label declarations of a classification system, from which actionable decisions are made by the warfighter (Sadowski, 2001, 2004). Varner (2002) discusses the "horizontal" and "vertical" analysis of a confusion matrix for ATR systems using this philosophy. A sample confusion matrix with two different "Non-declaration" options is presented as Figure 2.14 as presented by Sadowski (2001). A row is associated with each true class and a column is used for each Combat ID system output label. For most applications, engineers perform "horizontal" confusion matrix analysis, independent of true class prior probabilities, as depicted by the probabilities summing to 1 for each row of Figure 2.14.

| Truth | FEN | Library | CID System Output | | | | | |
| | | | Report: | | | | | |
Target	1-Friend 2-Neutral 3-Ambig 4-Enemy	1-In 2-Not In	"F15"	"F16"	"Mig 29"	"Su 27"	Not in Lib "Unknown"	No Rpt
F15	1	1	**0.800**	0.040	0.010	0.000	0.010	0.140
F16	1	1	0.010	**0.850**	0.010	0.030	0.010	0.090
Mig 29	3	1	0.010	0.010	**0.820**	0.010	0.030	0.120
Su 27	4	1	0.035	0.010	0.005	**0.850**	0.030	0.070
Mig 21	4	2	0.001	0.000	0.001	0.001	0.200	0.797
Mig 15	4	2	0.300	0.300	0.050	0.010	0.300	0.040

Figure 2.14 Sample Confusion Matrix with 2 Types of Non-declarations: "Not in Lib Unknowns" and "No Report"

The "vertical" analysis of the confusion matrix will yield estimates conditioned on the probability of label declarations. For example, horizontal analysis shows the system's F15, classification accuracy is $0.8/(0.8+0.04+0.01+0.00+0.01) = \sim 93\%$. In contrast, vertical analysis would report the CID system label accuracy of "F15" as $0.8/(0.10+0.10+0.035+0.001+0.30) = 0.80/1.156 = 69\%$, given the prevalence of each true target type is equal. This "vertical" analysis may reveal different performance with different prior probabilities of the true target types. The same estimate of "F15" output label accuracy may be computed for different true target prior probabilities. For example, first let P_F be the prior probability of Friendly fighters (F15s or F16s) and P_H is the prior probability of Hostile fighters (Mig 29, Su 27, Mig 21 and Mig 15). Let the probability across fighter types for the Friendly and Hostile classes be equal. Then, since the label events are mutually exclusive across true classes and collectively exhaustive, using the total Law of Probability, vertical analysis may be performed using different prior probabilities. Vertical analysis to estimate the label accuracy (*LA*) of a given

66

"output" yields the following for $j \in \{"F15", "F16", "Mig29", "Su27", "Unknown"\}$ and

$n = 6$ true target types,

$$LA("label\ j") = \frac{P(true\ type_i)P(true\ type_i | "label\ j")}{\sum\limits_{i=1}^{n} P(true\ type_i)P(true\ type_i | "label\ j")}. \quad (2\text{-}11)$$

With a hostile sparse ratio of $P_F : P_H = 4:1$, the priors for each true target type are (0.4, 0.4,

0.05, 0.05, 0.05 and 0.05). This provides an estimate of the LA("F15") as,

$$LA("F15") = \frac{(0.4)(0.8)}{(0.4)(0.8+0.1)+(0.05)(0.10+0.035+0.001+0.30)} = \frac{0.32}{0.36+0.0218} = 84\%.$$

Using the reversed probability of $P_F : P_H = 1:4$ representative of a hostile rich environment

yields,

$$LA("F15") = \frac{(0.1)(0.8)}{(0.1)(0.8+0.1)+(0.2)(0.10+0.035+0.001+0.30)} = \frac{0.08}{0.09+0.0872} = 45\%.$$

Thus, this small example shows the classification accuracy of a system yields reasonably

good results of 93% from horizontal analysis for the classification of a F15 as a "F15."

Yet, vertical analysis indicates a less favorable evaluation of the system. With

warfighters acting on the output decision labels of the Combat ID system, a label of

"F15" may be inaccurate over 50% of the time if operating in a hostile rich environment.

This label accuracy of the system is shown to vary from 45% to 84% by varying the

ration of $P_F : P_H$ priors from just 1:4 to 4:1.

2.4.2 Classification Accuracy

As presented for use by some ANN feature saliency identification methods,

classification accuracy may be calculated as

$$CA = \frac{\text{Number exemplars classified correctly}}{\text{Total number of exemplars}}.$$ (2-12)

Alsing (2000, chapter 7) offers insight for the interpretation of classification accuracy as a measure of classifier performance. *CA* is typically, reported for the Bayes optimal point, where, "the Bayes optimal point is the decision threshold for which the total misclassification error (1-*CA*) is a minimum" (Alsing, 2000: 7-2). The *CA* is the average of all objects being classified and may not be applicable to an ATR system where the cost of misclassifying a Friendly as a Target is extremely high. While *CA* may not be the best measure for comparing competing ATR systems, it is still frequently presented in the literature as a simple metric to show the performance of a system. For example, Simone *et al.* (2002) only present the optimal mean classification accuracy obtained for each of three classes when using image fusion techniques for remote sensing applications. Catlin, *et al.* (1999) also use the probability of correct ID's for the evaluation of ATR systems. Similarly, for a three-class pattern recognition effort (Laine *et al.,* 2002) simply report the classification accuracy when making comparisons of competing classifiers based on differing feature sets. Yet, these average *CA* measures were obtained from confusion matrices with insight obtained for the specific types of misclassifications between three classes observed from the Confusion Matrix cells, which are included within the original research in (Laine, 1999).

2.4.3 Confidence Testing

In addition to the *CA* obtained from the diagonal elements of the *CM*s, the probabilities of misclassification are also obtainable from the off diagonal cells. A second order statistic of variability would add value to these estimated mean *CA* and

68

probability of misclassification values. With a measure of the expected variance about the mean *CA* obtained, a classifier with high variance may be undesirable, even if it were to have a higher mean *CA* than a competing classifier. For two class problems, Classification Accuracy estimates can be modeled as binomial random variables. This facilitates the calculation of confidence intervals on this random variable, where an approximate $(1-\alpha)$ confidence interval for random variable $\hat{p}$ may be calculated from n samples as:

$$\hat{p} \pm Z_{(1-\alpha/2)} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \qquad (2\text{-}13)$$

where a normal approximation is assumed for large sample size with $n > 30$ (Wackerly *et al.,* 1996). For other measures of performance, confidence intervals may be obtained using multiple replications of an experiment. From the experimental replications, the mean and variance of a desired measure of performance may be estimated. A confidence interval could then be generated using,

$$\bar{Y} \pm Z_{(1-\alpha/2)} \frac{\sigma}{\sqrt{n}}, \qquad (2\text{-}14)$$

where, $\bar{Y}$ is the estimated mean value of some performance value, and σ is the observed standard deviation for $\bar{Y}$ (Wackerly *et al.*, 1996).

Some standard measures of performance associated with an ATR include the probability of true target declaration, false target declaration and with the inclusion of "Non-declarations," the probability of rejecting to declare, or the related probability of declaration. These standard measures of performance are estimated as follows:

- Probability of True Positive ($\hat{P}_{TP}$): probability an object is declared "Target"

69

given it is a Target,

$$\hat{P}_{TP} = \frac{\text{number of Targets declared "Target"}}{\text{total number of Targets declared}} \ . \qquad (2\text{-}15)$$

- Probability of False Positive ($\hat{P}_{FP}$): probability an object is declared "Target" given it is a Non-target,

$$\hat{P}_{FP} = \frac{\text{number of Non-targets declared "Target"}}{\text{total number of Non-targets declared}} \ . \qquad (2\text{-}16)$$

- Probability of Declaration ($\hat{P}_{Dec}$): probability an object is not too difficult to label and is declared as "Friend" or "Target", 1- probability of rejection, $\hat{P}_{Dec} = 1 - \hat{P}_{REJ}$:

$$\hat{P}_{Dec} = 1 - \hat{P}_{REJ}, \text{ where } \hat{P}_{REJ} = \frac{\text{number of objects not declared}}{\text{total number of objects evaluated}} \ . \qquad (2\text{-}17)$$

Along with a binomial approximation to estimate the associated variance and confidence intervals for each of these measures of performance, other methods may also be used. For example, the associated variance may be estimated through repeated training of a classifier if the parameters are determined stochastically, as is the case with some ANNs. Resampling techniques could also be used to create a stochastic process by training and or evaluating the classifiers with different sets of validation data. In addition, Bishop (1995: Ch 10, Bayesian Techniques) offers a means of computing the variance of a trained neural network function using a Bayesian approach that could be used to place a confidence interval about the estimate of class prediction for a specific model input. Further, if a region of feature space is of particular interest where misclassifications are likely to occur, the confidence intervals could be computed at

designed points to compare competing models and would provide a measure of their robustness across the input feature space.

Overall, when reports of a single measure of performance associated with a particular classification system are reported, confidence intervals provide additional information to determine if one system is statistically different from the other at a desired level of confidence.

2.4.4 ROC Curve Analysis

ROC curve analysis is a common evaluation tool for ATR systems and has been extensively applied to many dichotomous decision problems (Swets, 1964; Swets *et al.*; 2000a, 2000b). Given a finite data set, a standard ROC curve, f, can be thought of as a function of estimated performance measures (Alsing, 2000). A typical ROC curve illustrates the estimated feasible range of false positive, $\hat{P}_{FP}$, vs. true positive, $\hat{P}_{TP}$, detection probabilities. A ROC curve, f, can be generated empirically by varying θ over its range, Θ, as shown in eq. 2-18:

$$f = f(\theta) = \left\{ \left(\hat{P}_{FP}(\theta), \hat{P}_{TP}(\theta) \right) | \theta \in \Theta \right\} \qquad (2\text{-}18)$$

The resulting set of points $\left\{ \hat{P}_{FP}(\theta), \hat{P}_{TP}(\theta) \right\}$ start at the lower left corner and move toward the upper right corner as a decision threshold varies through a range of conservative to aggressive values. A notional ROC curve is shown in the following figure.
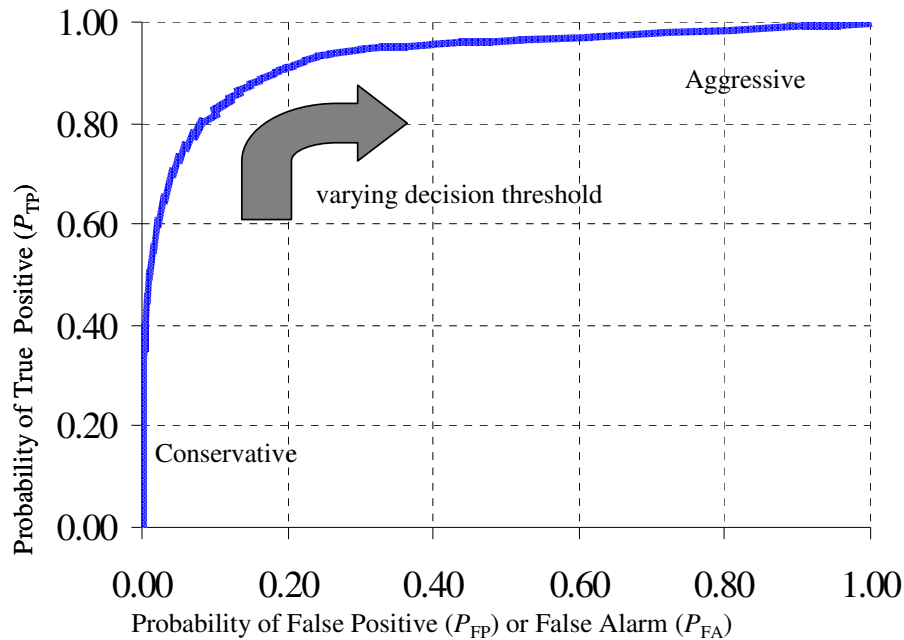
Figure 2.15 Typical Receiver Operating Characteristic (ROC) Curve

ROC analysis was developed from statistical decision theory as a tool for electronic signal detection (Peterson *et al.*, 1954) and has been extensively applied to decision making problems (Swets, 1964; Swets and Picket, 1982; Swets *et al.,* 2000) and is commonly used in biomedical research (Metz, 1986, 1989). Alsing (2000) provides a comprehensive review of the use of ROC curves in ATR research. The interested reader is referred to Alsing *et al*. (1999) for an example of generating a standard ROC curve, Lloyd (2002) for determining a theoretic ROC function, while an in-depth discussion of ROC curves is presented by Egan (1975) and Swets & Pickett (1982).

Standard ROC curves can be generated from conditional data labels, where values of P_{TP} and P_{FP} are estimated only using non-rejected data. Yet, these curves do not indicate the associated number of "Non-declarations." If a collection of ROC curves are generated with different rejection levels, ROC curve may be compared using the area

under the curve (AUC). The AUC may be approximated using a trapezoidal

approximation of the area, where a larger area indicates a ROC curve may be preferred

(Egan, 1975; Bradley, 1997), although it does not account for rejection. The average

metric distance developed by Alsing (2000) is an alternative means to compare ROC

curves. The average distance measure, MD, as developed by Alsing (2000) is,

$$MD = \frac{\sum_{i=1}^{n} \left\| \left(P_{TP}(\theta_i), P_{FP}(\theta_i) \right) - (\theta_i, \theta_i) \right\|_1}{n}, \qquad (2\text{-}19)$$

where $\left(P_{TP}(\theta_i), P_{FP}(\theta_i) \right)$ is the ith data point sampled from the ROC curve and $\left\| \cdot \right\|_1$ is the

1-norm. But, as stated by Alsing and Bauer (1998: 18) ROC curves have limitations:

> ROC curves are only generated for the simple two class "top layer" problem of
> differentiating between clutter and targets. Our literature review failed to find any
> research into the use of ROC curves for the "lower layer" problems of target
> group classification and specific target identification.

Thus, from the statement above, the standard ROC curves does not appear readily

applicable to ATR identification efforts where three or more output labels may be

desired, and standard comparison techniques do not account for "Non-declarations."

2.4.4.1 ROC Curve Extensions to Multiple Classes

While development of ROC analysis was can be traced back to the 1950's

(Peterson *et al.*, 1954), limited research has been identified in the literature for ROC like

analysis for more than two variables of interest. Mossman (1999) and Hand & Till

(2001) suggest the volume under a ROC like surface using three performance estimates

may be of value to compare trichotomous decision models. Yet, neither incorporates a

rejection option nor a means to determine the associated optimal threshold values.

Varner (2002) introduces use of two thresholds with a reject option to generate a family of ROC curves, but does so via a predetermined number of test objects to be labeled "Non-declaration" for each ROC curve. Threshold assessment is accomplished through analysis of numerous confusion matrices, which may be partially summarized by ROC curves.

Current literature does offer some extensions to the standard ROC curve with either trajectories or surfaces extending into 3-dimensions. Alsing *et al*., (1999) introduce use of a third measure to be plotted to generate a ROC trajectory. This trajectory shows the traditional ROC performance with a system allowed to reject hard to classify objects. While the traditional ROC curve is only suited to facilitate a 2-class target recognition classifier assessment, Hand and Till (2001) present a methodology to extend ROC area under the curve (AUC) for multiple classes. They provide a generalization of the area under the ROC curve for multiple class classification problems. This extension is made by averaging pair-wise comparisons between class assignments. Another possible performance metric for a three-class, "Target," "Non-target," "no-declaration," trichotomous decision task is the use of the Volume Under the Surface (VUS) obtained from a three-way ROC surface as presented by Mossman (1999). The surface is obtained by plotting the correct identification rates obtained from a contingency table or Confusion Matrix under a set classification rule. By systematically changing a single decision threshold for a given classifier a three-way ROC surface could be generated to facilitate visualization of the possible correct classification accuracies. The volume under such a surface is then suggested to use as a metric of comparison.

If used to plot the classification accuracy for "Target," "No-target" and "no-declaration" labels, the optimal point on the plot would then be at 100% Target declaration, 100% No-target declaration and 0% No-declarations. Thus, as presented, the volume under the surface may not be an appropriate metric, but modifications could probably be made to include false positives and no-declaration rates, rather than just the classification accuracy percentages. This may then overcome one shortcoming of the two volume metrics, i.e. the limited plotting of the true positive rates without including false positive rates within the 3-D plots. Such a 3-D ROC surface may then be useful for decision-making when the consequences of a false positive could be substantial including friendly fire.

2.4.4.2 ROC Curve Analysis under Uncertain Costs

In general, selection of a unique optimal point suggests perfect knowledge of priors and the associated costs of correct and incorrect decisions. On the other hand, use of a metric, such as the Area Under the Curve (AUC) or average metric distance (MD) suggests no prior information on the relative costs of errors or the prior probabilities of class types likely to be encountered. For most classification problems, including Combat ID, the assumption of perfect or no *a priori* information is probably poor at best, since some information with respect to costs and priors can probably be obtained or deduced. While certainly not perfectly known, the prior probability of Friend, Enemy and Neutral targets within a given area of interest can probably be identified within an order of magnitude using existing Intelligence data. Similarly, the associated costs for different potential classification error can probably be at least rank-ordered, where the cost of

misclassifying natural clutter as an Enemy target is certainly less than misclassifying an Enemy target as a Friend which may likely be less costly than classifying a Friendly target as an Enemy.

Similar findings have been recently addressed in the machine learning community. For example, Fawcett (2001) investigates different strategies for evaluating rule sets to maximize ROC performance, when class distributions are skewed and the costs associated with misclassifications are unequal. These flexible rule sets provide a means to determine a combined or fused ROC curve, where one classification method or a combination of methods may be preferred across a given range of P_{FP}. More details for the development of a hybrid classifier are found in (Provost and Fawcett, 2001), where the combination of two or more classifiers is assessed using a ROC curve. Multiple classifiers may be combined to generate a preferred ROC convex hull. The convex hull will then yield a P_{TP} value as good, or potentially better, than each individual classifier across the range of P_{FP} values. This is useful for real world applications, where the misclassification costs along with the prior probabilities of true classes are uncertain. This type of ROC fusion may be useful for the fusion of sensors and associated classifiers, which were initially developed independently. As environmental information about costs and priors is refined, a preferred classification model may then be identified. While only presented for a two-class identification effort, the authors note a potential for extension to multiple classes (Provost and Fawcett, 2001). Additional examples and background are found within (Provost *et al.*, 1998) and (Provost and Fawcett, 1997). Within these works the authors note, "Often in real-world domains there are no 'true'

target costs and class distributions" (Provost *et al.*, 1998), and in fact, estimation of such costs may change across time and across different situations.

To address some of the issues associated with comparing classifiers when misallocation costs are uncertain, Adams and Hand (1999) suggest use of a loss comparison (LC) index. This LC index ranges from -1 to +1, with a value of +1 indicating preference for one classifier across all feasible cost values. Since the AUC implicitly assumes implies equal misclassification costs, Adams and Hand (1999) suggest at a minimum, a subject matter expert may help to determine potential costs by estimating the minimum, maximum, and most likely cost ratios associated with a classification effort. By incorporating this cost information via a triangular distribution, the LC index may then show preference for one classification system across the feasible range of costs. Further suggestions for improving the practice of classifier performance assessment are contained within (Adams and Hand, 2000). Within this article, Adams and Hand state, "in many applications, assessment criteria are chosen that do not match the problem very well." In addition to presenting discussion against use of the AUC due to its inherent assumptions of equal and unknown costs, they suggest it is likely that costs may change over time. They also note, even when the AUC is used, further complications may arise if appropriate confidence bounds are not used. Thus, any performance measure used to determine a preferred classification model should not only incorporate all relevant decision information, such as the best estimates of costs, but should also estimate the variance associated with the measures being used, to ensure a reported difference is significant.

Overall, review of the literature for ROC curve analysis under uncertain costs indicates determining a preferred classifier via analysis of ROC curves may be performed. This analysis is typically dependent on either implied equal costs or a range of costs should be considered. Further, if the true class environment is uncertain with respect to the prevalence of different classes, then similar arguments may be used to suggest evaluation across a range of prior probabilities. Such analysis may try to incorporate available information to define a triangular or another parametric distribution associated with the priors of each class, or excursions may be performed to assess competing classifiers across a range of potential prior class probabilities.

2.4.5 Other Potential Evaluation Methods

The following sections will briefly introduce other potential methods of performing classifier assessments with the required addition of "Non-declarations." These methods include fuzzy logic, Dempster-Shafer (DS) analysis, multinomial selection procedure (MSP), linear goal programming (LGP), and decision analysis (DA).

2.4.5.1 Fuzzy Logic and Dempster-Shafer (D-S) Analyses

One possible way to model the new inclusion of "Non-declarations" is to retain a traditional binomial decision, "Target" or "Non-Target," but now if desired confidence is not obtained a "Non-declaration" is made. The use of fuzzy logic or Dempster-Shafer (DS) theory could be applied to this type of classification effort. While literature supports binomial classification performed in this manner, it does not appear to offer significant metrics to compare competing systems across various conservative to aggressive ROC-like thresholds. Many articles simply report the classification accuracy

78

(CA) obtained for two classes and also report a single measure for the number or percentage of Unknowns that were not included as one of the two default classes.

An overview for the use of fuzzy logic can be found in (Clutz, 2003). Magnus and Oxley (2002) present use of fuzzy logic in their investigation of the fusion and filtering of "arrogant classifiers." The three-value logic presented is an example of fuzzy logic with values associated with an object being classified in {false, uncertain, true}. They also present four-value expertise logic which further divides the uncertain class between {uncertain interpolation and uncertain extrapolation}. Thus, three or four class logic can be applied to indicate areas of uncertainty, possibly more useful than the two-value logic forced decision between {false, true} or {Target, Friend}. In summary, fuzzy or three-value expertise logic and the four-value logic presented could be used to expand a 2-class ATR label set to include "Non-declarations." These "Non-declaration" could then be reported in an appropriate confusion matrix and associated ROC curves may be generated to compare competing ATR classification systems.

Dempster-Shafer (D-S) analyses may also be applicable to classification problems where modeling uncertainty is desired. As stated by Simone *et al.* (2002: 6), "the Dempster-Shafer evidence theory…has been applied to classify multi-source data by taking into account the uncertainties related to the different data sources involved." Milisavljevic *et al.* (2003) apply D-S analysis in an iterative manner in their research to improve mine recognition through Dempster-Shafer fusion of ground penetrating radar data. Within this research, imaged objects are first screened as definitely Friendly, with high confidence, or as potential mines. Further pattern recognition analysis is then performed to classify an object as a mine, with the goal being to correctly identify 100%

of the mines while minimizing the false positives. As with the fuzzy logic encountered in

the literature, typically only the mean classification values are presented.

Overall, fuzzy logic and D-S theory can be used to model uncertainty and the

inclusion of "no-declarations." Both fuzzy and D-S theory may be applicable to an ATR

system to indicate when more information is needed to make a decision. The reduced

classification of Targets and Friends could be reported conditioned on declarations and

traditional ROC curves could still be produced, yet optimization between declarations,

P_{TP} and P_{FP} would still need to be determined.

2.4.5.2 Multinomial Selection Procedure (MSP)

A multinomial selection procedure for evaluating competing classifiers is

presented by Alsing *et al.* (2002) and summarized in (Kuncheva, 2004: 34-35). While

only two-class identification efforts are presented within (Alsing *et al*., 2002), feasible

use of a MSP procedure is demonstrated and provides a metric of the strength of

conviction or "probability of being the best" among competing classifiers. Bassham

(2002: 2-56) notes that MSP may be used to compare *k* competing classifiers across *n*

classes, and is not limited to the binomial declaration of target/non-target. Kuncheva

goes on to state,

> …the objective of the MSP is to find the best system, given a limited amount of
> data, which is most likely to be the best performer in a single trial among systems,
> rather than identifying the best average performer over the long run.

Kuncheva (2004:35) further states MSP has been demonstrated to, "be very sensitive in

picking out the winner, unlike the traditional error-based comparisons." Thus, use of

MSP inclusive of the assessment of "no-declaration" labels may provide a reasonable comparison procedure to assess competing ATR identification systems.

2.4.5.3 Linear Goal Programming

A good review of linear goal programming (LGP) as applicable to ATR system evaluation is found in (Bassham, 2002). The objective of LGP is to solve multivariable, multigoal problems, which is applicable to determining an optimal ATR system. In determining the optimal ATR system, trade-offs must be assessed between different declaration levels of Hostiles as "Targets" and Friendlies as "Targets" and "Non-declarations." An objective function must be specified, as applicable to ATR evaluation task at hand, and prioritized goals must be determined. For accurate assessment of the system, the prioritized goals require subject matter expert or decision maker input, which may be difficult to obtain or reproduce in a consistent manner.

2.4.5.4 Decision Analysis

Significant contributions for the use of decision analysis (DA) including a framework of assessing measures of effectiveness (MOEs) obtained from a combat model were developed by Bassham (2002) for the comparison of competing ATR systems. Like goal programming, decision maker input is required and may lead to biased comparisons based on a particular decision maker's preferences. Such differences were seen through a differing value structure obtained from decision makers in different AF MAJCOMS (ACC and AFMC). Since the DA framework used metrics from the MOEs obtained from a combat model, the lower level measures of performance (MOPs) were not directly compared. The resulting MOEs may only be useful for a particular

combat engagement with set concepts of operation (CONOPS) and where a predetermined Hostile environment is used. Sensitivity analysis was also presented to characterize the robustness of the DA value model parameters for comparison of the ATR systems. Of significant applicability is that the DA framework can be used with any ATR system incorporated into a combat simulation, where measures of combat effectiveness (MOEs) derived from the lower level decisions of declaring an object of interest as "Target," "Friendly" or "Non-declaration" show a net effect on the battlefield.

2.4.6 Classifier Performance with an Error-Reject Tradeoff

While ROC analysis is a standard tool for ATR research evaluation (Alsing, 2000), an operational ATR system should, at a minimum, provide two output labels plus a "reject to declare" option. A rejection parameter establishes a region where samples are considered too difficult to classify (Chow 1970); thus, declared "unknown." A classification algorithm for N–true classes and D_i decision labels with $i = 1,2,....,N$ seeks to assign patterns from true class ω_i to decision space D_i, by maximizing the classification accuracy. As presented by Fumera et $al.$ (2000), this accuracy is given as,

$$Accuracy = P(correct) = \sum_{i=1}^{N} \int_{D_i} p(x \mid \omega_i) P(\omega_i) dx,$$

(2-20)

where $P(\omega_i)$ is the prior probability of true class ω_i, and x is a pattern to be classified. Similarly, the goal of classification systems may be stated as the minimization of classification error as defined by,

$$P(error) = \sum_{i=1}^{N} \int_{D_i} \sum_{\substack{j \neq i \\ j=1}}^{N} p(x \mid \omega_j) P(\omega_i) \, dx \qquad (2\text{-}21)$$

(Fumera *et al.*, 2000). These decision rules may be referred to as Bayes-optimal, since

they assign each pattern x to the class with the maximum a posteriori probability,

$P(\omega_i \mid x)$. Rejection offers a means to obtain an increase in the classification accuracy,

with an associated decrease in misclassification errors. This performance improvement

may be obtained by allowing for the "Non-declaration" of difficult to identify patterns

(Chow, 1970). While rejection offers classification improvement, this performance

improvement comes at a cost. This cost includes a trade-off between ID system accuracy

and the cost of obtaining more information and lengthening the classification process if

an initial "Non-declaration" is made. By Using Chow's rule, a pattern x is rejected if,

$$\max_{k=1,2,\dots,N} P(\omega_k \mid x) = P(\omega_i \mid x) < \theta \qquad (2\text{-}22)$$

for $\theta \in [0,1]$. Patterns are accepted to be labeled as other than "Non-declaration," if,

$$\max_{k=1,2,\dots,N} P(\omega_k \mid x) = P(\omega_i \mid x) \geq \theta . \qquad (2\text{-}23)$$

Chow (1970), shows if all misclassification costs, C_e, rejection costs, C_r, and correct label

costs, C_c, are equal for all K classes, then the optimal θ may be obtained as,

$$\theta = \frac{C_r - C_c}{C_e - C_c} . \qquad (2\text{-}24)$$

where, typically $C_e, > C_r, > C_c$. By noticing that the prior probabilities associated with a

declaration of each of the true classes may vary according to the rejection thresholds and

may vary for estimates obtained from sampled data sets, Fumera, *et al.* (2000), propose

the use of multiple thresholds. Equations 2-22 and 2-23 are now slightly modified to

allow for different rejection thresholds for each true type. These equations thus become:

$$\text{reject if: } \max_{k=1,2,\dots,N} \hat{P}\left(\omega_k \mid x\right) = \hat{P}\left(\omega_i \mid x\right) < \theta_i, \quad \text{and} \tag{2-25}$$

$$\text{label as class } \omega_i \text{ if: } \max_{k=1,2,\dots,N} \hat{P}\left(\omega_k \mid x\right) = \hat{P}\left(\omega_i \mid x\right) \geq \theta_i. \tag{2-26}$$

In these equations, $\hat{P}\left(\omega_i \mid x\right)$ is the new estimate for the posteriori probability associated

with pattern vector x for class ω_i. In addition, by using these class related thresholds,

Fumera *et al.* (2000), have proven that the classification accuracy achieved for any

rejection rate is equal to or higher than use of a single rejection thresholds as presented by

Chow (1970). To determine the best values of the class related thresholds, a constrained

maximization problem is proposed to maximize the overall classification accuracy. The

constraints simply include the maximum total rejections allowed across all classes.

Further optimization of class-related rejection thresholds, may involve the optimization

of a risk or cost function. To minimize the total risk, a sum of all costs associated with

correct, incorrect and rejection of each true class may be optimized across all class-

related rejection thresholds. Thus, optimal rejection rates may be determined by an *a*

priori defined percentage of rejection allowed, or through the use of costs and a risk

function, where the costs may be difficult to quantify. In addition to presenting the

theoretical framework for use of class-related rejection thresholds, Fumera *et al.* (2004)

apply class related thresholds for the rejection and classification for text categorization.

Fumera and Roli (2004) also show the utility of such analysis when combining multiple

classifiers. In this research the classifiers are fused using simple averaging and linear weighting, where the error-reject trade-off is shown to improve. Some practical guidelines for combining classifiers with a reject option are presented, whereby individual classifier performance without a rejection option may be used to help assess reasonable weights for the linear combination of the same classifiers with a reject-option (Fumera and Roli, 2004). Thus, as desired by the warfighter, current error-reject research is being performed to allow for Non-declaration of potential targets with low levels of identification confidence.

By varying the rejection thresholds associated with a classification system, a family of ROC curves associated with different rejection criteria may be generated. These ROC curve present a visual means to see the improvement obtained via use of an error-reject option. If assessments are made across a range of all feasible thresholds, then the point associated with the optimal error-reject thresholds will be contained on the current plot of ROC curves. Several authors: Chow (1970), Devijer & Kittler (1982), Fumera *et al*. (2000) and Haspert (2000) use a Bayes optimal classification strategy to determine preferred classification and rejection rules by minimizing a Loss function. This will simply identify a single point on the ROC curve that is defined as best. A Loss function may include costs of rejection, correct and incorrect decisions all in equivalent units and incorporates prior probabilities of class membership. However, since ATR systems are likely to operate in a variety of conditions, the expected prior probability of Targets to Non-Targets may vary greatly (Ross *et al.,* 2002). Further, costs of "Non-declarations," that initiate ATR re-looks may be difficult to place in comparable cost units to false positive target IDs, which may lead to friendly fire. Thus, a Loss function

85

may not be appropriate to determine optimal classifier settings, and an alternative measure of effectiveness is sought.

Some rejection strategies set a predetermined number of objects as "non declared" based on *a priori* decision to have a set percentage undeclared. This method was used for the generation of 2-threshold ROC curves by Varner (2002), and is suggested by Fumera *et al.* (2000). Rather than assuming a certain percentage should be rejected for declaration, perhaps, a more appropriate strategy may use the posterior model class estimates to determine whether enough confidence is obtained to make a class label declaration. In addition to finding an optimal rejection level, research has been identified to incorporate more than 2 classes for ROC like analysis. Hand and Till (2001) and Mossman (1999) suggest the volume under a ROC like surface may be an appropriate metric to compare classifiers. From their plots, an increased volume may generally indicate robustness, but since "Non-declaration" labels are desired, the volume of these surfaces may not be an effective measure. Similar to the 3-D trajectory presented by Alsing *et al.* (1999) a 3-D ROC surface, which extends the standard ROC curve by adding a third performance measure that reflects the ability of the ATR algorithm to "reject" unknown or difficult to classify objects may be a useful aid in ATR analysis. This analysis includes use of two or more thresholds to tune an ATR system for the minimum trichotomous decision, with performance measures estimated using Test data.

Research by Dasarathy (2003, 2000b) includes sensor fusion with "Non-declaration" in a sensor system inclusive of re-looks, but is limited to the assumption that all sensor data is *independent*. The research was performed for the fusion of sensor decisions with designed $P_{correctID}$, $P_{falseID}$, $P_{no\ ID}$ levels, such that $P_{correctID} + P_{falseID}, + P_{no\ ID}$

= 1 for each sensor. Asymptotic properties associated with sensor "re-looks" are then reported where a re-look was triggered if the final output label from a suite of sensors was "no ID." The primary performance measures presented were plots with the percentage of correct ID and false alarm by the number of re-looks for sensors with the three predetermined probability characteristics. Asymptotic properties could then be observed after a finite number of looks, usually less than 10, for any given sensor characteristics.

2.5 Literature Review Summary

Overall, relevant background was presented to provide a foundation for the investigation of ATR system performance when "Non-declaration" labels are always an option and when sensor data may be correlated. Methods of feature extraction were reviewed, where limited observed correlation levels were found documented in the literature. A high level of desired confidence associated with ATR system labels was identified, where use of fusion is a prescribed means to increase identification confidence. Use of Boolean rules and neural networks to perform fusion were then introduced. While many analytical techniques are available to assess ATR and fusion algorithms with a "Non-declaration" option, some methods may be preferred. The preferred performance assessments should require minimal variation from more traditional ATR analysis. In particular, the use of confusion matrices and ROC curves are prevalent in the literature for the assessment of current ATR research. Thus, modification to these methods, by including warfighter "vertical" preferences and "Non-declaration" output labels is sought.

III. Mixed Variable Programming Formulation

As identified in the literature review, AF doctrine requires a certain level of confidence prior to declaring a target may be engaged. Thus, ATR systems must at a minimum make trichotomous decisions, where an object under consideration can be labeled as a "Target", "Non-target", or "No-declaration." Review of literature has not found methodologies that seek to optimize such a decision, without use of explicit cost information. Figure 1.1 is presented again to show this general process, where more than two sensors may be used and need not collect data at the exact same time. For an ATR system, a fusion rule should be chosen to combine data from two or more sensors, or determine an output label for a single sensor at each instance in time. The process may continue until a declaration is made or some upper time constraint or number of looks has been reached to label an ROI as a class other than "Non-declaration."
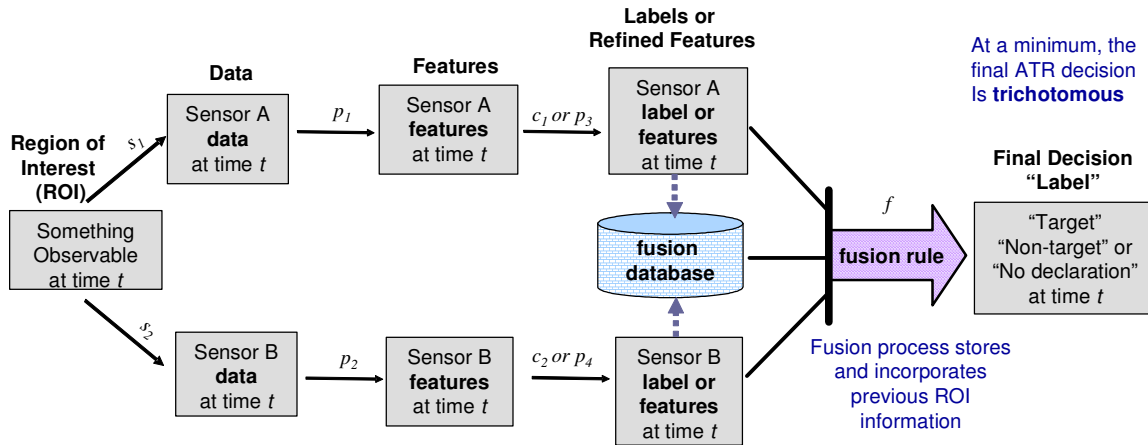
Figure 3.1 Notional ATR Process Model for Two Sensors through Time

3.1 Introduction to 3-D ROC Surface Generation via a Reject Option

By expanding the 3-D trajectory presented by Alsing *et al*. (1999), a 3-D ROC surface, extends a standard ROC curve by adding a third measure to reflect ATR declaration performance when difficult to classify objects are rejected. For a given finite data set, a 3-D ROC surface, *s*, is a function of θ represented in 3-space by three estimated probabilities: true positive detection, $\hat{P}_{TP}$, false positive detection, $\hat{P}_{FP}$, and rejection, $\hat{P}_{REJ}$. The relations for the estimated performance measures are given as follows:

$$\hat{P}_{TP} = \hat{P}_{TP}(\theta), \quad \hat{P}_{FA} = \hat{P}_{FP} = \hat{P}_{FP}(\theta) \quad \text{and} \quad \hat{P}_{Dec} = \hat{P}_{Dec}(\theta) = 1 - \hat{P}_{REJ}(\theta) \tag{3-1}$$
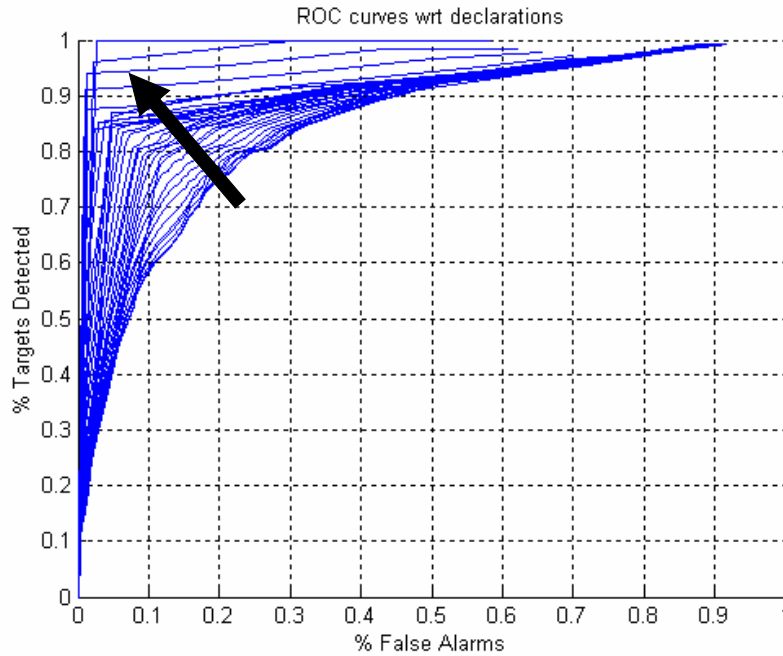
Figure 3.2 Family of ROC Curves Generated with Increased Rejection. The Arrow Pointing to the Upper NW Corner of the Plot Indicates General Performance Improvement as a Rejection Window is Increased

89

where $\hat{P}_{Dec}$ is the estimated probability of declaration and $\hat{P}_{FP}$ is equivalent to false

alarms, $\hat{P}_{FA}$. The three performance measures are functions of $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ may be

defined as $\boldsymbol{\theta} = (\theta_{low}, \theta_{up})^T$. A 3-D ROC surface, s, is generated empirically by varying $\boldsymbol{\theta}$

over its entire range, $\boldsymbol{\Theta}$:

$$s = s(\boldsymbol{\theta}) = \left\{ \left(\hat{P}_{TP}(\boldsymbol{\theta}), \ \hat{P}_{FP}(\boldsymbol{\theta}), \ \hat{P}_{Dec}(\boldsymbol{\theta}) \right) | \boldsymbol{\theta} \in \boldsymbol{\Theta} \right\} . \tag{3-2}$$

Rejection should improve ATR performance by only declaring those objects with high

likelihood of class membership (Chow, 1970). Classification can be delayed until

additional data are obtained and efficient sequential analysis (Wald, 1947) is performed

to limit data requirements. The 3-D ROC surface may be a useful tool for understanding

tradeoffs between $\hat{P}_{TP}$, $\hat{P}_{FP}$ & $\hat{P}_{Dec}$. To generate a 3-D ROC surface the ATR thresholds,

$\boldsymbol{\theta}$, needs to be further defined. For a trichotomous decision, $\boldsymbol{\theta}$ may include the size or

width of the rejection zone along with a conservative to aggressive ROC threshold. To

illustrate use of the varying rejection zone, consider the two-class detection problem

between hostile Targets and Friendly non-targets. Let ATR outputs, *ppT* & *ppF*, be

estimated posterior probabilities for the Target and Friend classes, such that:

$$ppT + ppF = 1. \tag{3-3}$$

Since $ppT + ppF$ sum to one, decisions may be made based on just *ppT*:

$$label = \left\{ \ "T" \ \text{if} \ ppT \ > \ \theta_{up}, \ "F" \ \text{if} \ ppT \ < \ \theta_{low}, \ "ND" \ \text{if} \ \theta_{low} \le ppT \ \le \ \theta_{up} \right\} \tag{3-4}$$

where θ_{up} and θ_{low} are upper and lower thresholds. These thresholds are functions of the

ROC threshold, θ_{ROC}, and a rejection threshold, θ_{REJ}, as shown in Figure 3.3 and defined

by following equations:

$$\theta_{low} = \theta_{ROC} \quad \text{and} \quad \theta_{up} = \theta_{ROC} + \theta_{REJ}. \qquad (3-5)$$

An example of the ROC and rejection thresholds is included in Figure 3.3. This shows

the declaration labels for a set of two-class data represented by the histograms of different

grayscale along an x-axis corresponding to the posterior probability of "Target."
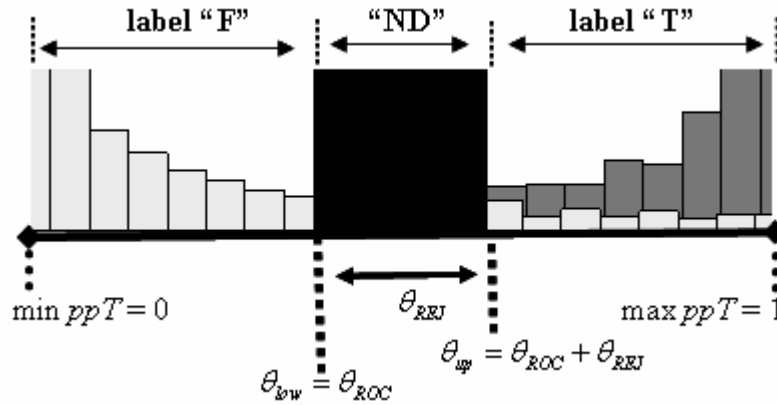
Figure 3.3 Example Relations and Labels for given Values of θ_{low} and θ_{up}

To generate a 3-D ROC surface, θ_{REJ} is varied from 0 (no rejections) to some

upper limit, $\theta_{REJ} \leq 1.0$, for estimated label probability scores. The ROC threshold, θ_{ROC},

is then systematically varied from 1- θ_{REJ} down through 0.0. This facilitates evaluation of

the full conservative to aggressive ROC trade space associated with a given rejection

window. Thus, for a dichotomous decision plus rejection, a 3-D ROC surface reflects the

available performance across a threshold decision space, as can be seen in Figure 3.4.

91
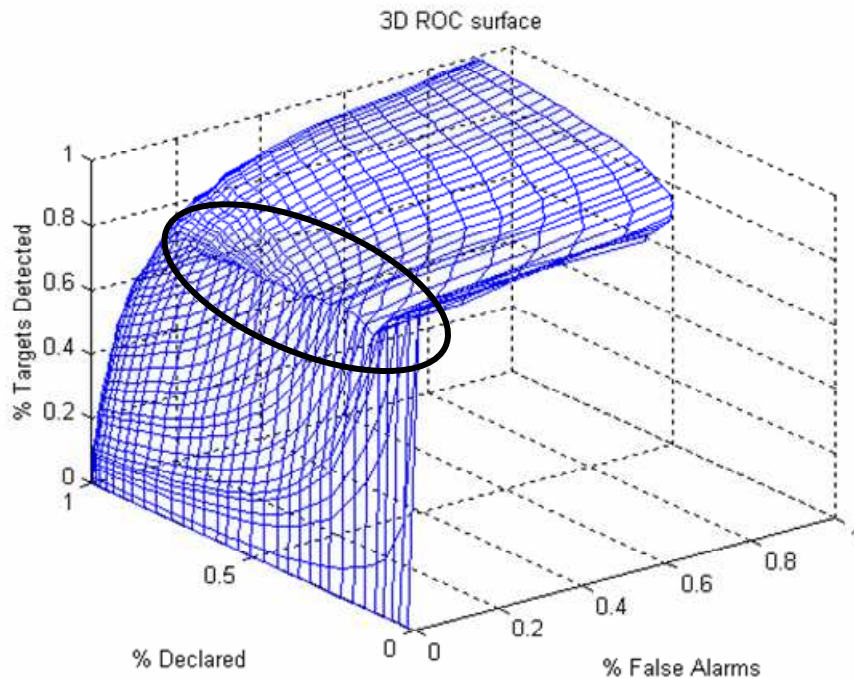
**Figure 3.4 Family of ROC Curves Plotted with the % Declared, 1 indicates 100%
Declaration Rate with 0% Rejected**

While a classical knee in the curve can be seen in the above 3-D ROC surface,
determining the single optimal point associated with specific threshold values is visually
difficult. Further, no methods to determine the associated optimal point were identified
in the literature, which do not include the use of explicit costs or a predetermined
maximum rejection probability. A predefined maximum level of rejection would always
yield a given percentage of objects to be "non declared," even if enough confidence was
available to make label decisions. On the other hand, use of explicit costs requires all
misclassifications, including "no-declarations" to be placed in equivalent cost units,
which this research seeks to avoid.

3.2 Background Definitions and Assumptions for ATR System Evaluation

New research is suggested to use nonlinear optimization of the feasible decision space generated across all potential classifier label mappings associated with variable thresholds. An optimization strategy may be performed that is required to meet certain requirements, such as minimum error rates, and could then seek to maximize the percentage of Targets correctly identified. The number of total targets being labeled as "Non-declaration" offers one degree of freedom to meet the minimum error rate, by allowing some targets to be rejected when confidence is low. This may also help to obtain fewer false negative declarations, where targets that look similar to non-targets are "non declared." Non-linear optimization may be performed across multiple time periods with sensors allowed to acquire multiple looks of a target. The fusion strategy may allow for multiple looks if a "Non-declaration" is made, or may force a minimum number of looks to help achieve a required level of confidence prior to making a decision. For example, the basic framework to determine the optimal rejection and ROC thresholds settings could be obtained by maximizing the percentage of true positive target declarations subject to other error constraints, without the use of explicit cost information. This is similar to using a Neyman-Pearson criterion for ROC curve analysis (Varshney, 1997), in which an acceptable false positive probability of error is established, and the parameter settings associated with the corresponding maximum percentage of true positive declarations are used by the system.

3.2.1 Definitions

Before the formal development of the mathematical programming formulation, it is useful to first define some of the key terminology to be used. The following is a list of basic definitions used to be later used in the development of a mathematical programming formulation for the optimization of a Combat ID ATR systems with fusion.

- ATD/R: Automatic Target Detection and Recognition. ATD/R includes the task of initially detecting a region of interest (ROI) that may potentially have a target of interest.

- ATR: Automatic Target Recognition without a man-in-the-loop. Use of an ATR system assumes time critical identification is being performed. At a minimum, an ATR system is defined by a fusion rule to combine data, the sensors used to collect data and the associated parameters or thresholds used at either the sensor level or at the fusion level to make output label declarations.

- Class: A desired level of fidelity to group true objects of interest by the identification system. For example, the set of {Friend, Enemy, Neutral} identifies three true classes.

- Clutter: natural objects that may degrade sensor performance including the environmental background consisting of foliage, rocks, etc.

- Confuser: man-made objects with similar feature space representation to in-LIB Targets and Friends.

- Extended operating condition (EOC): Physical or environment settings significantly different from the data used to train an identification system. For example,

radar data collected at a depression angle not within the database used to train a classifier, physical modifications of targets to include different configurations of a T-72 Main Battle Tank (MBT) not included with training data, or even different levels of environmental concealment by foliage, mud, etc., which may alter the data derived from a given sensor.

- In-LIB (in-Library): Samples of known target types similar to the representation of targets in the data set used to train an ATR identification system.

- Label: A desired level of fidelity to specify output decisions of interest by the identification system. For example, the set of {"Hostile," "Friend," "no-declaration"} identifies a minimum set of potential output labels.

- NiL (Not in-Library): Target types significantly different than those used to train an ATR system. These targets of interest may be detected by an ATR system, but are sufficiently different from the list of known target types to match against. These targets should be labeled as "NiL," "Non-declaration" or "unknown" by an ATR system.

- ROI: Region of Interest. The area under investigation for the identification task-at-hand after a positive cue for a man-made object of interest is communicated to an ATR system or made by the ATD/R system.

- Target Type: Classification of an object of interest based on physical properties at a high level of fidelity for discrimination. Objects of the same target type will only vary slightly by serial number, tail number, etc. For example, all variations of a T-72 MBT are considered the same target type as are all variations of an F-15. As such, similar feature vector representations may be used to represent objects of the same target type.

- $\boldsymbol{\theta}^{ij}$ = Vector of all decision thresholds associated with traditional ROC (P_{TP} and P_{FP} tradeoffs), rejection and other decisions. These thresholds are unique to a specific application ij, where the application includes all thresholds associated with an entire fusion system i and the associated individual sensor j, where either i or j may be dropped, if the threshold is clearly associated with only the fusion rule or sensor.

3.2.2 Initial Background Assumptions

It is assumed for this research that an ATR system under evaluation is developed sufficiently to meet certain initial requirements. First, a Combat ID ATR system is assumed to filter naturally occurring background clutter sufficiently to only provide positive detection of an ROI if a man-made object is present. In other words, natural clutter ROI's should not be detected as an ROI under question, although confuser classes may be considered. Further, individual sensors being fused are assumed to be mature with reasonable performance accuracy. As such, fusion of two sensors should yield new information for the classification task-at-hand. Next, from the previous definitions, an ATR system is only trained with representations of in-LIB target types. As depicted in Figure 3.1, the ATR system relies on a sequential process with "Non-declarations." A feature-space vector, posterior probabilities, or an output label associated with each ROI may be reused or updated by an ATR system, after an initial "Non-declaration" label. The sequential process updates are made after the acquisition of new data. Further, it is assumed, at some level in the ATR system, a continuous value associated with the desired label decisions may be assessed to estimate the posterior probabilities of label membership prior to making a label assignment. This may be performed at either an

individual sensor level or after fusion of multiple sensors or looks has occurred for some fusion methods.

In order to determine an optimal point on a 3-D ROC surface, or to determine the preferred ROC and rejection thresholds, it is first important to assess the different levels of error as they impact the mission. As viewed by the warfighter, Combat ID errors may be deemed as Critical or Non-critical errors (Sadowski, 2003). When identification errors do not meet the requirements of being a Critical or Non-critical error, the Combat ID errors may be defined as Lesser errors. The primary similarity between these error labels is an associated actionable decision, which may be analyzed through subsequent vertical analysis of the ATR system output labels. Thus, in contrast to evaluating a system's performance using standard ROC measures such as P_{TP} and P_{FP}, the prevalence of each true target class will affect the performance of a system, as would be the case when an ATR system is fielded operationally. Brief definitions and examples of the error types follow:

- **Critical Errors:** These errors are characterized by an incorrect positive or incorrect negative shoot decision. They have the potential to lead to grave consequences and may contribute to undesirable "CNN events." Examples include a Friend labeled as "Enemy" leading to fratricide, Neutral/Civilians labeled as "Enemy" leading to collateral damage, or the lost opportunity to engage the enemy and preempt an enemy strike.

- **Non-Critical Errors:** These errors are characterized by less than optimal use of weapons and sorties without the potential of grave consequences. Examples include weapons expended on non-desired targets of the day that do not lead to the loss of lives

for non-combatants or coalition forces, a weapon suboptimally matched to a target, or expending weapons on decoys, etc. In some cases an Enemy labeled as "Friend," without the display of imminent hostile intent, may still be correctly labeled by a future CID system. Thus, in some circumstances an incorrect negative shoot decision may be deemed as a non-critical error, depending on the situation and associated risk.

- **Lesser Errors**: These errors do not fall into the critical or non-critical definitions and are characterized as having little or no-impact to a warfighter decision (e.g. friend classified as "clutter" is still a non-shoot decision). Lesser errors may also include lucky classification, such as correctly declaring any NIL target types as an appropriate label, even though the ATR system has not been trained to recognize them. Lesser errors are not directly analyzed.

3.2.3 Analysis of Confusion Matrices with Unknown Class Labels

Initial analysis is performed at the Friend, Enemy and Neutral (FEN) level to assess the impact of "Non-declaration" labels, and to determine the contributions of the different Critical and Non-critical misclassification errors. From the confusion matrix presented next, misclassification of Friends as "Neutral" or vise versa yields minimal performance impact with no direct analysis of this Lesser error.

98

Classifier "Labels"

True Classes	*"Enemy"*	*"Friend"*	*"Neutral"*	*"No declaration"*	Horizontal Totals
Enemy	Enemy labeled "Enemy"	Enemy labeled "Friend"	Enemy labeled "Neutral"	**Enemy labeled "ND"**	**Enemy evaluated**
Friend	Friend labeled "Enemy"	Friend labeled "Friend"	Friend labeled "Neutral"	**Friend labeled "ND"**	**Friend evaluated**
Neutral	Neutral labeled "Enemy"	Neutral labeled "Friend"	Neutral labeled "Neutral"	**N labeled "ND"**	**Neutral evaluated**
Vertical Totals	**"Enemy" declared**	**"Friend" declared**	**"Neutral" declared**	**"ND" declared**	

Legend	Contribution	Analysis
	Correct ID	Horizontal
	Critical Error	Vertical
	Non-Critical Error	Vertical
	Non-Declaration	Horizontal
	Lesser Error	No direct

Figure 3.5 Confusion Matrix Error Assessments for FEN Classes

Further, if Critical errors include both the incorrect labeling of Friends or Neutrals as "Enemies," and the incorrect labeling of Enemies as "Friends" or "Neutrals," non-critical errors are not obtained for these given output labels. Since, misclassification of an Enemy may not directly have a potentially grave consequence based on the location and threat of the Hostile force, a revised confusion matrix at the FEN level including both Critical and Non-critical errors may be developed as shown in Figure 3.6. Analysis of this confusion matrix provides a two-class problem with representations of both a Critical error and a Non-critical error obtained when the Enemy targets do not present an imminent threat.

Classifier "Labels"

True Classes	*"Enemy"*	*"FN"*	*"No declaration"*	**Horizontal Totals**
Enemy	Enemy labeled "Enemy"	**Enemy labeled "FN"**	**Enemy labeled "ND"**	**Enemy evaluated**
Friend or Neutral	F or N labeled "Enemy"	F or N labeled "FN"	**F or N labeled "ND"**	**F or N evaluated**
Vertical Totals	**"Enemy" declared**	**"FN" declared**	**"ND" declared**	

Legend	Contribution	Analysis
	Correct ID	Horizontal
	Critical Error	Vertical
	Non-Critical Error	Vertical
	Non-Declaration	Horizontal
	Totals	H or V Analysis

Figure 3.6 Revised Confusion Matrix Error Assessments for FEN Classes

If labeling a hostile Enemy as a "Friend or Neutral," should be included in the calculation of critical error, due to an increased threat to friendly forces, then the Enemy class must be subdivided to obtain both Critical and Non-critical errors. This is shown in Figure 3.7, where The Enemy class is divided between a desired Target of the Day (TOD) and Other Hostile (OH) Targets. The calculation of Non-critical error now requires discrimination between types of enemy targets, where a specific ground target such as a high threat Surface-to-Air missile may be the primary Target of the Day, to be neutralized by current air tasking order (ATO) sorties.

True Classes	*"TOD"*	*"Other Hostile"*	*"Friend / Neutral"*	*"No declaration"*	Horizontal Totals
TOD	TOD labeled "TOD"	**TOD labeled "Other Hostile"**	TOD labeled "FN"	**TOD labeled "ND"**	**TOD evaluated**
Other Hostile	**Other Hostile labeled "TOD"**	Other Hostile labeled "Other Hostile"	Other Hostile labeled "FN"	**Other Hostile labeled "ND"**	**Other Hostile evaluated**
Friend or Neutral	F or N labeled "TOD"	F or N labeled "Other Hostile"	F or N labeled "FN"	**F or N labeled "ND"**	**F or N evaluated**
Vertical Totals	**"TOD" declared**	**"Other Hostile" declared**	**"FN" declared**	**"ND" declared**	

Legend	Assessment	Analysis
	Correct ID	Horizontal
	Critical Error	Vertical
	Non-Critical Error	Vertical
	Non-Declaration	Horizontal
	Totals	H or V Analysis

Figure 3.7 Confusion Matrix Error Assessments for Multiple Hostile Classes

Similar confusion matrix analysis may be performed, where the labeling of each true class as one of the classifier output labels maps to a correct ID, Critical error, Non-critical error, or Lesser error. Consolidated classes may be determined as was done with the combination of Friend and Neutral classes. This occurs for limited cases where the incorrect label between true classes is a Lesser error and all other errors are the same for the two true classes. An example of adding a new class is as follows. If enemy decoys or confusers are desired to be added as a fourth true class and fifth ATR classifier output label, then assignment of the TOD as "enemy confuser" may result in a Critical error with a lost opportunity to engage a high threat Enemy. The misidentification of an Other Hostile as "enemy confuser" would likely be considered a Lesser error, since the current sortie would not engage an Other Hostile or enemy Decoy, and would continue with the current mission to seek the desired TOD. Misidentification of a Friend or Neutral as

"Decoy / Confuser" may be a Non-critical error, since no substantial impact was made with respect to the shoot decision, but potential incorrect intelligence information may be shared to generate faulty battlespace awareness. This battlespace awareness may add risk to future missions. Analysis may continue and yield a confusion matrix as shown in Figure 3.8. Since the errors obtained for Other Hostile and Decoy or Confuser classes include symmetric Lesser errors, they are a candidate for class consolidation. Yet, because the errors are not equivalent for "TOD" or "Friend or Neutral" labels, they may not be combined.

Classifier "Labels"

True Classes	*"TOD"*	*"Other Hostile"*	*"Decoy / Confuser"*	*"Friend / Neutral"*	*"No declaration"*
TOD	TOD labeled "TOD"	**TOD labeled "Other Hostile"**	TOD labeled "DC"	TOD labeled "FN"	**TOD labeled "ND"**
Other Hostile	**Other Hostile labeled "TOD"**	Other Hostile labeled "Other Hostile"	Other Hostile labeled "DC"	Other Hostile labeled "FN"	**Other Hostile labeled "ND"**
Decoy or Confuser	**D or C labeled "TOD"**	D or C labeled "Other Hostile"	D or C labeled "DC"	**D or C labeled "FN"**	D or C labeled "ND"
Friend or Neutral	F or N labeled "TOD"	F or N labeled "Other Hostile"	**F or N labeled "DC"**	F or N labeled "FN"	**F or N labeled "ND"**

Legend	Assessment	Analysis
	Correct ID	Horizontal
	Critical Error	Vertical
	Non-Critical Error	Vertical
	Non-Declaration	Horizontal
	Lesser Error	Vertical

Figure 3.8 Confusion Matrix Error Assessments for Four True Classes

In addition, it is of importance to note, an "Unknown" or "Non-declaration" would be the correct label for all NIL objects being assessed. Further, ATR systems should be validated against both in-LIB and NIL samples to examine system robustness and the ability to sufficiently make "Non-declarations."

3.3 Development of a Mathematical Programming Formulation

To develop a mathematical programming framework, the primary goals and objectives of an ATR system are first defined. The primary goal of an ATR system can be considered to help neutralize the enemy more efficiently by delivering more "bombs-on-target." Any system helping with Combat ID should also help to minimize friendly-fire and collateral damage, as may be associated with some critical errors. Thus, either maximizing bombs-on-target or minimizing critical error might be the primary objective for an ATR system, depending on the specific situation and rules of engagement. Because the warfighter will act on the ATR output labels, optimization of an ATR system should be performed to support the warfighter via vertical analysis of the Critical and Non-critical errors. The initial optimization formulation will focus on maximizing bombs on target, given acceptable wartime constraints, as shown in Table 3.1. Initial values for the presented goals reflect a general order of magnitude desired and are not official requirements. These values are a reasonable estimate obtained from communication with a Combat ID Principal Systems Architect (Sadowski, 2004) at Air Combat Command (ACC) and may be considered reasonable assessments by an ATR subject matter expert.

Table 3.1 Initial Mathematical Programming Formulation Goals and Objectives

Goals:	Implementation:	Objective function impact:
Maximize $TPR(\boldsymbol{x})$ the True Positive Rate per time/look	max $TPR(\boldsymbol{x})$	maximum TP's per time to obtain more bombs-on-target attempt to quantify TP vs. time to ID relationship

Other goals accomplished by meeting constraints	Desired order of magnitude	Impact for constraints:
Minimize Critical Errors	$E_{CR} < \sim 0.02$ (< a few %)	limits feasible label declarations obtained by different ROC thresholds though vertical analysis of true class prior probabilities and error estimates
Minimize Non-critical Errors	$E_{NC} < \sim 0.05$ (< a few %)	secondary concern to critical errors, further restricts feasible operating space of traditional ROC curve via vertical analysis
Maximize system declarations (for in-Lib) targets	$P_{Dec} > \sim 0.70$	allows system to reject difficult to identify objects with low classification confidence so long as a minimum declaration level is achieved

3.3.1 Mathematical Program Decision Variables

To determine the best ATR system through optimization and mathematical programming, decision variables must first be defined. The following is a description of key decision variables for the optimization of ATR Combat ID systems. With multiple looks required to gain confidence in a decision prior to engagement, a fusion rule must be selected. Let F_i be an indicator variable associated with the selection of the ith of f total fusion rules under consideration. Then, $F_i \in \left\{ F_1, F_2, ..., F_f \right\}$, where $F_i = 1$ if the fusion rule is selected, and $F_i = 0$ if the fusion rule is not selected. Next, assume that s total sensors may be selected for use by the ATR system. Let S_j be an indicator variable

associated with the selection of the jth of s total sensors under consideration. Then,

$S_j \in \{S_1, S_2, ..., S_s\}$, where $S_j = 1$ if sensor j is selected, and $S_j = 0$ if the sensor is not used.

The selection of a limited number of sensors may be imposed by different design

constraints. Design constraints may also include a minimum number of desired sensors.

The design of an ATR system should also address obtaining a minimum level of

confidence, prior to generating output labels. Fusion of sensor data will be used to

increase this confidence. A minimum number of looks may *a priori* be unknown to

obtain this confidence for systems under development. Thus, when evaluating ATR

systems, where the required minimum number of looks, *ML,* is unknown, a constraint

may be added to assess systems using different required minimum-forced looks. The

required value of *ML* may be varied as a categorical variable and may be considered part

of a fusion rule. This particular parameter associated with a fusion rule is highlighted,

because depending on the operational mission and environment, it is assumed different

levels of confidence may be required and the fusion of multiple looks is a key to

obtaining this confidence (Dept. of AF, 1998, 1999). Different costs may be associated

with the fusion rule and sensor variables, and included as design constraints. These

design constraints may include the monetary costs associated with the lifecycle of the

ATR system and include research and development (R&D), procurement of ATR fusion

systems and sensors, along with the cost of maintaining the system (Feuchter, 2000).

Physical cost constraints may also be imposed. These may include a maximum weight of

a sensor ensemble, size associated with the sensors, the communication bandwidth

requirements associated with sensor and fusion rule combination, etc.

A final set of decision variables includes all continuously valued thresholds and parameters used by a fusion rule or by a sensor. The use of thresholds may assume an available posterior probability estimate in $[0,1]$ can be obtained within the ATR system for all desired output labels. These posterior probability estimates may be obtained at the sensor or fusion rule level within an ATR system. Let $\theta^{ij}_{decision}$ indicate a threshold associated with a specific fusion and sensor application, denoted by ij, along with a specific decision. By convention, let sensor $j = 0$ denote those thresholds associated at the fusion algorithm level. These thresholds include the ROC threshold for conservative to aggressive Target declarations and a rejection threshold to determine a region to make "Non-declarations." In summary, each threshold may be associated with a unique fusion rule, sensor, and decision.

For n-correct output labels, the inclusion of "Non-declarations" yields $n+1$ total output labels. Label decisions may be made using n thresholds to obtain $n+1$ labels. Starting with a minimum of three output labels, any "Non-declaration," "Friend" or "Target" labels may be further divided. For example, it may be of value to subdivide the "Non-declarations," as definitely "NiL," or "potentially in-LIB," for those cases when separation between two classes is not sufficient to make a decision. Another example of a hierarchical subdivision includes the separation of "Targets" as "Target of the Day" (TOD) or "Other Hostile" (OH) output labels. The threshold θ_{TOD} may be used to determine "TOD's", with labels determined as:

$$label = \{ \ "TOD" \text{ if } ppTOD \geq \theta_{TOD}, \ "OH" \text{ if } ppTOD < \theta_{TOD} \}, \qquad (3\text{-}6)$$

where *ppTOD* is an appropriate estimated prior probability of membership to "*TOD*."

Using appropriate estimates, a sequential strategy could be used to further divide all

"Friends" and "Targets" into *n* total labels. Each of these $\theta_{decision}$ thresholds may be varied

to make a dichotomous decision using estimated posterior probabilities of any two

classes, or of any two consolidated classes. In addition to these thresholds, other

parameters may be included in the formulation. For instance, some continuous valued

threshold associated with identifying out-of-library objects could be varied to trade-off

performance for the initial detection of an in-library target being in the Region of Interest

(ROI). Or, as used by Fumera and Roli (2000), a different rejection threshold may be

associated with each output label.

3.4 Mixed Variable Programming (MVP) Formulation

To perform non-linear optimization of the fusion systems using mixed variable

programming, decision variables must be further defined. Let *x* define a vector of all the

decision variables, which will be partitioned into continuous and discrete parts, x^c and x^d,

respectively as defined by Audet and Dennis (2000) for their pattern search algorithm

used to solve mixed variable programs. Next, let n^c and n^d denote the maximum

dimensionality of continuous and discrete variables. Since the dimensionality of (x^c, x^d)

may vary within a given formulation, let $x^c \in \Re^{n^c}$ and $x^d \in Z^{n^d}$ be the maximum

dimensionality of continuous and discrete variables. By convention (Abramson, 2002),

simply ignore unused variables where $X^c \subseteq \Re^{n^c}$ and $X^d \subseteq Z^{n^d}$. Thus, the decision

variable space may be defined as $X = X^c \times X^d$. For the continuous threshold space, let

X^c be equivalent to the threshold-space, Θ. The maximum dimensionality of Θ is then

n^c. The discrete decision space is defined as X^d and includes the available fusion-space,

F, $\times$ sensor-space, **S**. If only optimizing across fusion models and sensors, $X^d = \mathbf{F} \times \mathbf{S}$.

Further categorical variables may be included. One example is the predetermined

number of minimum looks required to obtain confidence in a target label prior to making

a final decision. In this case the categorical variable decision space is simply expanded to

include $\mathbf{F} \times \mathbf{S} \times \mathbf{ML}$. In this specific example, **ML** includes all potential levels of

minimum looks by an ATR system. Equivalently, the number of minimum looks could

be subsumed by different fusion algorithms under consideration, where a different

number of *ML* would be considered a different fusion algorithm. Thus, for these decision

variables, the best solution of $x \in Z^{n^d} \times \Re^{n^c}$; is obtained by optimization across all

feasible $\mathbf{F} \times \mathbf{S} \times \mathbf{ML} \times \Theta$.

The primary goal of the mixed variable programming is the determination of the

optimal fusion rule, with the optimal selection of sensors, forced looks and thresholds.

This goal is obtained through the optimization of a desired objective function for the

ATR system, such as maximizing the probability of True Positive Target declarations

across time, via an estimated True Positive Rate ($T\hat{P}R$). A secondary assessment of a

given ATR system, as defined by the categorical combination of fusion rule, sensor, and

minimum looks, may be to identify the range of feasible operating thresholds and other

internal system variables. The assessment of feasibility across decision variables may

help show system robustness across assumptions of priors, EOC's and NIL targets. A

mixed variable formulation to assess competing ATR systems involving the different

categorical variables as described above is then defined as follows:

Objective Function:

$$\max_{x \in X} \quad T\hat{P}R(x) = \frac{\hat{P}_{TP}(x)}{\hat{E}(time_{TP}(x))} \quad \text{maximize } TPR(x), \text{ the True Positive Rate} \tag{3-7}$$

Subject to:

Initial Warfighter Operational Constraints:

$$\hat{E}_{CR}(x) < \Pi_1 \quad \text{limit incorrect fire decisions (vertical analysis)}$$

$$\hat{E}_{NC}(x) < \Pi_2 \quad \text{limit lower impact incorrect decisions (vertical analysis)}$$

$$\hat{P}_{Rej}(x) < \Pi_3 \quad \text{limit Non-declarations (horizontal analysis)}$$

Fusion Rule constraint:

$$\sum_{i=1}^{f} F_i = 1 \qquad \text{limit selection of a single Fusion Rule}$$

$$\text{where } F_i = \begin{cases} 1 \text{ if } i \text{ th Fusion Rule used} \\ 0 \text{ otherwise} \end{cases}$$

Sensor Selection constraints:

$$\sum_{j=1}^{s} S_j \leq s \qquad \text{limit selection of Sensors selected to those available}$$

$$\sum_{j=1}^{s} S_j \geq 1 \qquad \text{ensure selection of at least 1 Sensor}$$

$$\text{where } S_j = \begin{cases} 1 \text{ if } j \text{ th Sensor is selected} \\ 0 \text{ otherwise} \end{cases}$$

Minimum Look Constraint:

$$ML \geq \min Looks \qquad \text{require minimum looks prior to label declaration}$$

Monetary Budget Constraints:

R&D Budget Constraint:

$$\sum_{i=1}^{f} C_{R\&D}^{F_i} F_i + \sum_{j=1}^{s} C_{R\&D}^{S_j} S_j \leq B_{R\&D} \qquad \text{limit R\&D costs}$$

where, $C_{R\&D}^{F_i}$ is the R&D cost associated with fusion system i

$C_{R\&D}^{S_j}$ is the R&D cost associated with sensor j

Procurement Cost Budget Constraint:

$$\sum_{i=1}^{f} C_{PC}^{F_i} F_i + \sum_{j=1}^{s} C_{PC}^{S_j} S_j \leq B_{PC} \qquad \text{limit Procurement Costs}$$

where, $C_{PC}^{F_i}$ is the procurement cost associated with fusion system i

$C_{PC}^{S_j}$ is the procurement cost associated with sensor j

Operation and Maintenance (O&M) Budget Constraint:

$$\sum_{i=1}^{f} C_{O\&M}^{F_i} F_i + \sum_{j=1}^{s} C_{O\&M}^{S_j} S_j \leq B_{O\&M} \qquad \text{limit O\&M costs}$$

where, $C_{O\&M}^{F_i}$ is the procurement cost associated with fusion system i

$C_{O\&M}^{S_j}$ is the procurement cost associated with sensor j

Physical System Constraints:

Physical Weight Constraint:

$$\sum_{i=1}^{f} C_W^{F_i} F_i + \sum_{j=1}^{s} C_W^{S_j} S_j \leq B_W \qquad \text{limit physical weight of ATR system}$$

where, $C_W^{F_i}$ is the weight associated with fusion system i

$C_W^{S_j}$ is the weight associated with sensor j

Physical Space/Size Constraint:

$$\sum_{i=1}^{f} C_{SZ}^{F_i} F_i + \sum_{j=1}^{s} C_{SZ}^{S_j} S_j \leq B_{SZ} \qquad \text{limit size of ATR system}$$

where, $C_{SZ}^{F_i}$ is the size associated with fusion system i

$C_{SZ}^{S_j}$ is the size associated with sensor j

Communication Bandwidth Constraint:

$$\sum_{i=1}^{f} \sum_{j=1}^{s} C_{BW}^{F_i S_j} F_i S_j \leq B_{BW} \qquad \text{limit communication bandwidth}$$

where, $C_{BW}^{F_i S_j}$ is the bandwidth requirement for fusion system i using sensor j

Threshold Constraints:

For the top-level decision, depicted in Figure 3.3 by "Target," "Friend," or "Non-declaration" decisions, specific constraints for these thresholds may be written as:

$\theta^{ij}_{low} \geq 0 \;\; \forall \; i, j$ satisfy lower threshold requirement

$\theta^{ij}_{low} \leq \theta^{ij}_{up} \;\; \forall \; i, j$ satisfy ordinal requirement

$\theta^{ij}_{up} \leq 1 \;\; \forall \; i, j$ satisfy upper threshold requirement

Basic constraints for any pair-wise decision using posterior probability estimates of two desired labels or consolidated groups of labels may be included as:

$$\theta^{ij}_{decision} \geq 0 \quad \forall \; i, j \qquad \text{satisfy lower threshold requirement}$$

$$\theta^{ij}_{decision} \leq 1 \quad \forall \; i, j \qquad \text{satisfy upper threshold requirement}$$

Other constraints may be developed specific to each continuous parameter. For example, it may be desired to add threshold constraints associated with time as either a continuous or discrete value. Variable thresholds across time may be useful to obtain a more efficient ATR system. For example, a minimum of *n*-looks or sensor duration greater than a predefined *minimum* number of seconds may *usually* be required before an ATR system can provide a reasonable label assessment. The rejection threshold associated with these looks should be large enough to promote additional looks to acquire new data when the identification confidence is low. If a target can be labeled with high confidence after 1 or 2 looks, the ATR system may operate more efficiently if this label is declared, and the ATR system is now available to assess the next ROI. Subsequent label updates may be obtained with a smaller rejection window, when a limited amount of new sensor information, with diminishing returns for the improvement of classification accuracy, is obtained by additional sensor looks. Specific constraints would then need to be developed across discrete or continuous time periods for each $\theta^{ij}(t)_{decision}$ where new constraints may force a minimum rejection window size. The decision space would now include optimization across all feasible $F \times S \times ML \times \Theta \times T$, where T is the associated feasible finite time-domain.

3.5 Variations to Initial MVP ATR System Optimization

An objective function inclusive of temporal system performance has been proposed, along with desired operational constraints followed by an initial formulation of potential ATR system design constraints. The selection of this objective function is flexible and may be replaced to optimize one of the warfighter operational constraints, such as minimizing the Critical error. As stated by Brown (2004), when performing optimization for military applications, "expect any constraint to become an objective, and vice versa." This may be particularly true in the case of using an ATR system for Combat ID in a politically sensitive situation, where minimizing collateral damage may be more important than maximizing bombs-on-target. This change in the objective function would require some measure of *TP* or *TPR* to be included as a constraint, to ensure an acceptable number of targets are declared. In addition, constraints may be modified, added or deleted depending on situation specific objectives. For example, if research is desired to design an optimal ATR system with respect to the time required to make an initial detection of an ROI containing a man-made object of interest, the formulation may focus on different internal ATR system thresholds. These thresholds may be used to determine if sufficient evidence is obtained from an initial surveillance look of an area to warrant an increase in allocated sensor time for the area or to cue additional ISR assets as part of the optimization across a netcentric system of multiple fused assets.

3.6 Limitations and Concerns for ATR System Optimization

One limitation of the optimization framework presented within this chapter, is that it focuses on the optimization of one objective function, while simply meeting the requirements of the levied constraints. The maximization of *TPR* was presented as a useful objective function for obtaining more "bombs-on-target," but it was noted that in some circumstances other objective functions, such as minimizing the Critical Error may be preferred. Analysis across two or more highly desired objectives may be undertaken to understand the trade-offs between competing objectives. Thus, a better understanding of the relationship between *TPR* and Critical Error for an ATR system may be sought. Assessment of a pareto-optimal boundary across these two performance estimates may be insightful to further evaluate a system if desired Critical error constrain values are not known with certainty. If constraint levels can not be determined, goal programming or multi-objective decision making may also be helpful to understand such tradeoffs. These analyses may be used to compare two competing systems to see if one system dominates the other, across different regions of the measures of performance. Alternatively, these analyses may assist decision makers, who have a broader knowledge of the requirements of an ATR system, by providing insight and helping them to determine specific values for the operational constraints of an ATR system.

A potential modification, to the mathematical formulation presented, may seek to use constraints more similar to the objective function. Since the proposed objective function is an estimated rate across time, having some operational constraints inclusive of time may also be of value. Yet, this is a first step beyond traditional static time ROC

analysis, and as such it is desirable to limit the modifications of currently accepted performance measures. Standard reported measures of an ATR system performance include P_{TP} and P_{FP} (Alsing, 2000; Bassham, 2002; Ross *et al.,* 1999; Ross and Mossing 1999). These measures of performance are currently accepted within the ATR community and should be reported along with the newly developed optimal *TP Rate.* Further steps away from the current measures of performance should only be made after an initial acceptance of the proposed methodology within this chapter is received as a means to gain further insight of ATR performance.

In addition, since the vertical analysis of the error constraints is highly dependent on the prior probabilities or prevalence of class types, analysis should be performed to evaluate competing systems of the preferred system parameters across a range of priors. Sensitivity analysis may be useful to perform this task. This may be accomplished by assessing an ATR system across a range of priors to determine its performance limitations given different class prevalence. These estimates may be evaluated by analyzing the performance associated with each desired class prior, averaging performance across chosen priors, or weighting the performance across a range of foreseeable prior probabilities using a parametric distribution.

Other concerns, for the use of this mathematical programming assessment of ATR include the determination of a preferred system with a desired level of confidence. It should be noted, that the objective function as well as the operational constraint values are all estimated measures of system performance and may be modeled as random variables. These random variables are typically estimated using different data sets. If a

stochastic process is used to obtain a fusion model, as may be required for training certain fusion algorithms, then additional variability may be associated with each measure of performance. While measures of performance associated with a dichotomous decision may be modeled as a binomial random variable, no standard parametric distributions may be applicable to an objective function that incorporates time. As noted by Catlin *et al.* (1999) ATR test data is expensive. Thus, limited data sets may be available to obtain accurate estimates and confidence bounds for the performance of these systems. Research by Ross *et al.* (1997) suggests assessments may vary considerably across data sets which are associated with different extended operating conditions (EOC). Thus, confidence intervals are desired, yet may be difficult to obtain without evaluation of the ATR systems across numerous expensive data sets.

3.7 Summary of MVP Optimization

Overall, the use of mathematical optimization to determine a best ATR system is presented. A best ATR system is defined by a preferred fusion rule, sensor ensemble and the associated thresholds. The preferred ATR system is obtained without use of explicit misclassification costs. An objective function that incorporates the time associated with making declarations is incorporated. Constraints are outlined to account for the warfighter preferences and are flexible, where new ones may be added or current ones may be modified or deleted. A flexible objective function may be changed to fit different operational goals. The explicit use of "no-declaration" labels is highlighted as a top decision priority by the ATR system. The results obtained from assessing different ATR

116

systems using this formulation should offer some insights as to the operational utility of a proposed ATR system without assessment by modeling and simulation (M&S) methods. The use of combat models to evaluate competing systems should still be accomplished, from which the impact of the ATR's performance may be evaluated for measures of effectiveness at the mission or campaign level (Feuchter, 2000; Bassham, 2002). The optimization method presented within this chapter may help limit the number of ATR systems to be compared, across specific missions or scenarios using M&S, where the ATR parameters are just one of many systems to include as input for a combat model. The overall utility of this optimization framework may be viewed as a new means to accomplish two different objectives. First, ATR system performance can be tuned toward a known operating condition. Second, comparisons across competing systems can be made at a design level. The comparison of ATR systems using various fusion strategies may then be performed through a range of test data.

The following two chapters present a variety of ATR system experiments using the MVP optimization presented within this chapter. Chapter 4 presents experiments using generated data representative of two true output labels plus the option for rejection. Chapter 5 presents a comprehensive experiment, with three desired ATR system output labels. This experiment includes individual fusion algorithm optimization with subsequent comparison of the fusion systems, using collected radar data of ground targets. Both chapters use the warfighter operational constraints and threshold constraints. The thresholds are held constant through time and no examples of budgetary monetary or physical design constraints are illustrated. For one of the initial experiments,

117

investigation was performed using the proposed mathematical framework to assess the impact of different data correlations of known parametric design. For this experiment, only a single fusion algorithm was used for a set number of sensors. Thus, only the decision thresholds were included as decision variables and no categorical variables are presented in the first application of this optimization formulation.

IV. Mathematical Optimization for the Fusion of Generated Data

This chapter summarizes two primary research efforts undertaken to demonstrate the utility of the mathematical optimization framework and gain insight for the fusion of data with synthetically generated features with various degrees of correlation. This chapter contains three primary sections, the development of the mathematical optimization and constraints for a two-class problem with "Non-declarations," application to generated Gaussian data for multiple sensors and multiple looks, and to generated temporal signatures representing data patterns observed from imaging two-satellite classes. More details of the Gaussian data fusion experiments can be found in a SPIE conference proceeding (Laine and Bauer, 2004a). Specifics for the second experiment involving the fusion of generated temporal signatures via an Elman RNN can be found in three references. Feature selection using an RNN is documented in an *IEEE International Joint Conference on Neural Networks* (IJCNN) proceeding, (Laine and Bauer, 2003), application of the optimization framework is documented in an *Artificial Neural Networks in Engineering* (ANNIE) proceeding (Laine and Bauer, 2004b), while a more thorough discussion for use of an RNN and fusion is found in an invited journal article submitted to *Military Operations Research* (Laine and Bauer, 2005).

4.1 Introduction to 2-Class Data Fusion Experiments

Many classification problems can be modeled at the top-level using 2-classes, where either the desired class is identified or not. For example, an ATR system may

declare an unknown object as "Target" or "Non-target," where "Target" includes enemy assets and "Non-Target" may include clutter, neutral or friendly forces. Yet, before a target is declared and engaged, the USAF requires a minimum level of confidence (DAF, 1998, 2000). Consequently, an ATR system forcing two decision labels is inadequate. A minimum of three output classes, including "Target," "Non-target" and "Non-declaration" is required to account for those cases when the confidence is not met. Intelligence fusion is identified as a guiding principle to obtain increased confidence for combat identification (Dept. of AF, 2000).

A sample 2-class confusion matrix with a "Non-declaration" option is presented as Figure 4.1, with a row for each true class and a column for each model label. As previously mentioned, for most applications, engineers perform "horizontal" confusion matrix analysis, independent of class membership prior probabilities. In contrast, warfighters are predominately concerned with ATR output labels (Sadowski, 2004). "Vertical" analysis of the confusion matrix yields error estimates from the number of class declarations. These estimated values may be obtained from the confusion matrix frequency counts associated with the tested prior probabilities of classes. Equivalently, the error rates may be calculated as conditional probabilities using Bayes rule with other prior probabilities of class membership (denoted P_T and P_F).

True Classes	"Target"	"Friend"	"No declaration"	Horizontal Totals
Target	Target labeled "Enemy"	Target labeled "Friend"	Target labeled "Unknown"	Target evaluated
Friend	Friend labeled "Target"	Friend labeled "Friend"	Friend labeled "Unknown"	Friend evaluated
Vertical Totals	"Target" declared	"Friend" declared	"Unknown" declared	

Legend	Contribution	Analysis
	Correct ID	Horizontal
	Critical Error	Vertical
	Non-Critical Error	Vertical
	Non-Declaration	Horizontal
	Totals	H or V Analysis

Figure 4.1 Confusion Matrix with Rejection and Error Contributions

With only two classes, the Critical and Non-critical errors will be defined as follows:

- Probability of a Critical Error: the probability a "Target" declaration is actually a Friend (i.e. those cases which may result in friendly-fire),

$$\hat{P}(E_{CR}) = \frac{\text{number of Friends declared as "Target"}}{\text{total numer of "Target" declarations}} = \frac{P_F \hat{P}_{FP}}{P_F \hat{P}_{FP} + P_T \hat{P}_{TP}} \; , \qquad (4\text{-}2)$$

- Probability of a Non-Critical Error: the probability a "Friend" declaration is actually an enemy Target (i.e. lost opportunities to engage the enemy),

$$\hat{P}(E_{NC}) = \frac{\text{number of Targets declared as "Friend"}}{\text{total number of "Friend" declarations}} = \frac{P_T \hat{P}_{FN}}{P_F \hat{P}_{TN} + P_T \hat{P}_{FN}} \; , \qquad (4\text{-}3)$$

and the probabilities of False Negatives and True Negatives are $\hat{P}_{FN} = \hat{P}_{FN}(\boldsymbol{\theta}) = 1 - \hat{P}_{TP}(\boldsymbol{\theta})$

and $\hat{P}_{TN} = \hat{P}_{TN}(\boldsymbol{\theta}) = 1 - \hat{P}_{FP}(\boldsymbol{\theta})$. Assuming all objects belong to one of the true classes, the

probability of declaration, $\hat{P}_{Dec}$, can be used as a performance measure of the "Non-

declaration" labels. The probability of rejecting a sample is related as: $\hat{P}_{REJ} = 1 - \hat{P}_{Dec}$. The

probability of a declaration is then:

- Probability of a Declaration: the probability of either class being declared "ND"

$$\hat{P}_{Dec} = \frac{\text{\# of objects declared as "ND"}}{\text{total objects evaluated}} = P_T \hat{P}_{UT} + P_F \hat{P}_{UF} , \qquad (4\text{-}4)$$

where $\hat{P}_{UT} = \hat{P}("ND"|T)$ and $\hat{P}_{UF} = \hat{P}("ND"|F)$. Table 4.1 summarizes the probability

estimates associated with horizontal analysis of each row, and the vertical analysis

metrics in terms of the confusion matrix cells, *CM*(row, col), from Figure 4.1. With all

probabilities estimated from test data, the "hat" has been dropped, and $P = \hat{P}$ will be

assumed for the remainder of this chapter.

Table 4.1 Typical Performance Measures Associated with the Confusion Matrix Cells,
CM(row,col) from Figure 4.1

Classifier "Labels"

		"Target" | declaration	*"Friend" | declaration*	*"Non-declaration"*
True Classes	Target	$P_{TP} = \dfrac{CM(1,1)}{CM(1,1)+CM(1,2)}$	$P_{FN} = \dfrac{CM(1,2)}{CM(1,1)+CM(1,2)}$ $P_{FN} = 1 - P_{TP}$	$P_{UT} = \dfrac{CM(1,3)}{CM(1,4)}$
	Friend	$P_{FP} = \dfrac{CM(2,1)}{CM(2,1)+CM(2,2)}$	$P_{TN} = \dfrac{CM(2,2)}{CM(2,1)+CM(2,2)}$ $P_{TN} = 1 - P_{FP}$	$P_{UF} = \dfrac{CM(2,3)}{CM(2,4)}$
Other metrics		$E_{CR} = \dfrac{P_F P_{FP}}{P_F P_{FP} + P_T P_{TP}}$	$E_{NC} = \dfrac{P_T P_{FN}}{P_F P_{TN} + P_T P_{FN}}$	$P_{REJ} = P_T P_{UT} + P_F P_{UF}$ $P_{Dec} = 1 - P_{REJ}$

Optimization may be performed across two thresholds to obtain a desirable

objective, such as a maximum true positive declaration rate, $TPR(\theta)$, or in the case

without temporal assessment, $P_{TP}(\boldsymbol{\theta})$, subject to meeting other constraints identified by

the decision maker. With limited categorical variables in this chapter, the optimization

framework can be presented across just the continuous valued thresholds. A basic

framework to determine optimal ROC and declaration thresholds can be obtained by

solving the following mathematical program:

$$\max_{\boldsymbol{\theta} \in \Theta} \; TPR(\boldsymbol{\theta}) = \frac{P_{TP}(\boldsymbol{\theta})}{\mu_{timeDec}(\boldsymbol{\theta})} \qquad \text{maximize } P_{TP} \text{ per mean time to declare} \qquad (4\text{-}1)$$

or, $\max \; P_{TP}(\boldsymbol{\theta})$ to maximize the probability of true positive declarations without time

s.t. $E_{CR}(\boldsymbol{\theta}) < \Pi_1$ limit potential friendly fire

         $E_{NC}(\boldsymbol{\theta}) < \Pi_2$ limit lost opportunities to engage the enemy

         $P_{Dec}(\boldsymbol{\theta}) > \Pi_3$ limit Non-declarations

Each Π_i is set at a tolerable limit < 1. The expected number of looks is

$\mu_{timeDec}(\boldsymbol{\theta}) = E(L(\boldsymbol{\theta}))$, and $\boldsymbol{\Theta} = \{\boldsymbol{\theta} : \boldsymbol{\theta} = (\theta_{low}, \theta_{up})^T \in \Re^2 \ni 0 \le \theta_{low} \le \theta_{up} \le 1\}$, is as shown in

Figure 3.3. Given a data set, the associated function is estimated by varying the

thresholds, $\boldsymbol{\theta}$, across all the desired ranges to determine the associated performance

values. The performance measures are then analyzed to determine which settings yield

feasible design points and the optimal point. To aid in visual analysis, connecting the

estimated values of P_{TP}, P_{FP} & P_{Dec} will generate a 3-D ROC surface, as introduced in

Chapter 3. For these preliminary two-class investigations, $\boldsymbol{\theta}$ includes the width of the

rejection window along with a ROC threshold to facilitate conservative to aggressive

settings. As presented in Chapter 3, two-class ATR outputs, *ppT* & *ppF*, will be used as

estimated posterior probabilities for Target and Friend classes, with:

123

$$ppT + ppF = 1. \tag{4-5}$$

Since $ppT + ppF$ sum to one, decisions may be made based on just ppT:

$$label = \left\{ \text{ "}T\text{" if } ppT > \theta_{up}, \text{ "}F\text{" if } ppT < \theta_{low}, \text{ "}ND\text{" if } \theta_{low} \leq ppT \leq \theta_{up} \right\} \tag{4-6}$$

where θ_{up} and θ_{low} are upper and lower thresholds, functions of the ROC, θ_{ROC}, and rejection threshold, θ_{REJ}:

$$\theta_{low} = \theta_{ROC} \qquad \text{and} \qquad \theta_{up} = \theta_{ROC} + \theta_{REJ} \tag{4-7}$$

A 3-D ROC surface may then be generated to help visualize the different declaration trade-offs as presented in Chapter 3. Optimization of the thresholds is then performed to obtain the maximum *TPR* or maximum P_{TP}, subject to other constraints identified by a decision maker. The framework used within this chapter is a subset of that presented in Chapter 3, with optimization focused primarily on these two thresholds. Limited categorical decision variables are incorporated to determine a preferred fusion method or ensemble of sensors. One categorical variable assessment includes use of three different methods of generating posterior probabilities from temporal looks and is presented in Section 4.3.4. Each of the three posterior probability assessments may be representative of a fusion rule. The next categorical variable under investigation is the determination of a preferred set of sensor data. In Section 4.4, each fusion rule is defined by the set of input features used by an RNN model with optimization performed across the two continuous thresholds to compare the two ATR systems.

4.2 Gaussian Data Generation for Classifier and Fusion Testing

The generation of data with known correlation is desired to determine the effects various correlation levels may have on different sensor fusion techniques. A desirable fusion technique will yield optimal target classification in terms of maximum true positive target identification (ID) and minimum false positive target ID, regardless of the correlation levels between input data. One particular research topic of interest is how correlated data affects the classification results for fusion algorithms that may or may not assume independent data is being fused. This may provide insight for the design of fusion systems forced to operate in an environment with various degrees of correlated input data. One approach to assessing the impact of correlated data is to design an experiment with generated data with known levels of correlation. The primary or first-order levels of correlation to control are the correlation across any two features stationary in time and the autocorrelation within a feature observed across the first time lag. A first step toward the exploration of the effects of correlation across features in a synthetic classifier fusion-testing environment was performed by Storm (2003) in which three classifier fusion techniques were explored. Further investigations using a synthetic classifier fusion-testing environment were performed by Clemans (2004) and Leap (2004). In the research performed by Clemans (2004), effects of correlation across features were analyzed across three sensor/classifier algorithms using an optimization framework to compare different fusion methods. The research performed by Leap (2004) assessed the impact of sample size, across feature correlation, and the within or temporal feature correlation. Each research effort used multivariate Gaussian data generated using a process similar to that described in Section 4.2.2 to follow.

Multivariate Gaussian data has been used as a synthetic classifier fusion-testing environment for the assessment of ATR systems. Such assessment may be desired across multiple looks of each potential Target or Non-Target. In addition to being able to easily model known correlation levels with generated multidimensional Gaussian data, use of a Gaussian distribution is well supported to represent a "final" ATR score which may be derived from one or more sensors to include radar or spectral data. First, during a feature extraction process, signal processing typically includes a linear transformation with subsequent linear operations to refine features. Specifically, for real time ATR, feature extraction must be performed quickly while vast amounts of radar or spectral data are being collected and processed. Thus, linear operators are prevalent for ATR feature generation as discussed by Cooke *et al.* (2000), Meyer (2003), Nasr (2003), Schroeder (2002) and Suvorova & Schroeder (2002). Further, if data of high dimensionality is mapped to a much lower dimension through a linear transformation such as principal component analysis (PCA) or singular value decomposition (SVD), the resulting low dimensional data will tend to be normally distributed with probability approaching one (Diaconis and Freedman, 1984; Hall and Li, 1993). If an appropriate non-Gaussian distribution of the data is known based on governing physical properties or observations, a Gaussian representation may still be appropriate since a Power Transform (PT) (Fukunaga, 1990; Bhatnagar *et al.,* 1998) can be used to convert many distributions close to normal using $z = x^v$, with $0 < v < 1$. For example, an application of a PT to high range radar (HRR) data has been shown to result in Gaussian distributed variables (Bhatnagar *et al.,* 1998).

Use of multivariate Gaussian data may also be justified from an information theoretic point of view. For a measured mean and covariance of sample data set, the Gaussian distribution provides for a parametric modeling with the maximum entropy (Duda *et al.*, 2001), where entropy is originally defined within (Shannon, 1948). Thus, use of a Gaussian distribution should be a conservative estimate of the information associated with a given generated data feature. Finally, by using a multivariate Gaussian representation of sensor data, designed correlation structures both across sensors and within a sensor through time can be quickly generated to test fusion algorithms for numerous designed levels. Overall, experiments performed using generated multidimensional Gaussian data appears reasonable.

4.2.1 Generation of Univariate Gaussian Data with Autocorrelation

A univariate stochastic process can be represented as:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} \ldots \phi_p \tilde{z}_{t-p} + \varepsilon_t \ . \tag{4-8}$$

This describes an autoregressive (AR) process of order p where the model coefficients ϕ_i can be estimated from the data (Box and Jenkins, 1976: Ch 3), ε_t is the associated error of the AR process and is modeled as white noise at time t, and $\tilde{z}_t$ is the deviation from the expected value μ such that $\tilde{z}_t = z_t - \mu$. The random variable z_t is an observation from random series z_k represented by a univariate normal or Gaussian distribution with population or class mean μ_k, standard deviation σ_k and probability density function (pdf),

$$f(z) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left[-\frac{1}{2}\left(\frac{z - \mu_k}{\sigma_k}\right)\right], \qquad (4\text{-}9)$$

where $f(z)$ is the probability density function of a sample observation from class k. While z is approximated by a Gaussian distribution, the observations are not independent, with each z_t represented by a linear combination of previous observations and a white noise component. The white noise series ε_t is also assumed to be Gaussian, with $\mu_\varepsilon = 0$ and variance σ_ε^2, and is independent across time and is denoted i.i.d. $\sim$norm$(0, \sigma_\varepsilon^2)$.

To generate a series of autocorrelated observations with a desired mean and variance, a first-order autoregressive process AR(1) can be used, where eq. 4-8 can be written as a recursive relation such that each observation is a function of the white noise:

$$\tilde{z}_t = \phi_1\tilde{z}_{t-1} + \varepsilon_t = \varepsilon_t + \phi_1\varepsilon_{t-1} + \phi_1^2\varepsilon_{t-2} + \dots, \qquad (4\text{-}10)$$

where $-1 < \phi_1 < 1$ for the process to be stationary with a constant mean and variance, and the influence of prior observations will decay across time. The autocorrelation between two consecutive observations can be estimated as

$$\rho(k) = \phi_1\rho(k-1) \qquad \text{for } k > 0. \qquad (4\text{-}11)$$

With $\rho(0) = 1$, eq. 4-11 can be used recursively to obtain the autocorrelation at any desired time lag k and is calculated as

$$\rho(k) = \phi_1^k \qquad \text{for } k \geq 0. \qquad (4\text{-}12)$$

Eq. 4-12 produces an exponential decay toward zero when ϕ_1 is positive and oscillating decay when ϕ_1 is negative. In addition, the maximum likelihood estimate (MLE) of $\phi_1 = \rho(1)$ and the variance of the AR(1) process is (Box and Jenkins, 1976, p. 58),

128

$$\sigma_z^2 = \frac{\sigma_\varepsilon^2}{1 - \rho(1)\phi_1} = \frac{\sigma_\varepsilon^2}{1 - \rho(1)^2} \ . \tag{4-13}$$

Thus, a stationary Gaussian univariate process z_t with $t = T$ new observations, starting

value z_0, mean μ_z, variance σ_z^2, and lag 1 autocorrelation $\rho(1) = \rho$, can be generated as:

1. Generate $E_1, E_2,\ldots, E_T$ from a standard normal distribution, i.i.d. ~norm(0,1)
2. Let $\varepsilon_t = E_t \sqrt{\sigma_z^2(1 - \rho^2)}$, (Solving eq. 4-13 for σ_ε^2 yields $\sigma_\varepsilon^2 = \sigma_z^2(1 - \rho^2)$)
3. Starting with $t = 1$, let $z_t = \mu_z + \rho(z_{t-1} - \mu_z) + \varepsilon_t$
4. Repeat steps 2 and 3 until $t = T$

Note, if $\sigma_z^2 = 1.0$, then a white noise series with variance of $\sigma_\varepsilon^2 = \sqrt{1 - \rho^2}$ will generate a

stationary series with a constant variance of 1.0, and if $\mu = 0$ the series can be post-

processed using $z = \mu_k + \sigma_k z'$ where z' represents standardized data, μ_k is the desired

mean and σ_k is the desired standard deviation for population k.

4.2.2 Generation of Multivariate Gaussian Data with given Correlation

The multivariate vector autoregressive, VAR(p), model is an extension of the

univariate AR(p) model, where a pth order VAR(p) model is defined as (Lütkepohl,

1993:9),

$$\mathbf{z}_t = \boldsymbol{\mu} + \mathbf{A}_1 \mathbf{z}_{t-1} + \ldots + \mathbf{A}_p \mathbf{z}_{t-p} + \boldsymbol{\varepsilon}_t \ , \tag{4-14}$$

where $\mathbf{z}_t$ is an n-dimensional random variate, where each z_i at a given time t represents a

feature observed in time and any observation $\mathbf{z} = (z_1, z_2,\ldots, z_n)^T$ has expected values

$\boldsymbol{\mu} = (\mu_1, \mu_2,\ldots, \mu_n)^T$ and standard deviations $\boldsymbol{\sigma} = (\sigma_1, \sigma_2,\ldots, \sigma_n)^T$ where $(\cdot)^T$ denotes the

transpose and $\mathbf{z}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ are column vectors. Each $\mathbf{A}_i$ is a fixed n x n matrix of

coefficients, $\boldsymbol{\Sigma}$ is the n x n covariance matrix, and $\boldsymbol{\varepsilon}_t$ is a n-dimensional column vector of

stationary white noise, representing the part of $\mathbf{z}_t$ not linearly dependent on past observations. Observation $\mathbf{z}$ is now modeled by a multivariate Gaussian distribution with pdf,

$$f(\mathbf{z}) = \frac{1}{2\pi^{n/2}|\mathbf{\Sigma}_k|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{z}-\mathbf{\mu}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{z}-\mathbf{\mu}_k)\right], \tag{4-15}$$

where $f(\mathbf{z})$ is the probability $\mathbf{z}$ was a sample taken from population class k. After dropping the class indicator for a given population, ρ_{ij} is defined to be the correlation across features z_i and z_j, $\mathbf{R}$ is defined to be the matrix of correlation coefficients, and the covariance matrix $\mathbf{\Sigma}$ can be expressed as $\mathbf{\Sigma} = \mathbf{\sigma R \sigma}^T$ as shown below.

$$\mathbf{R} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{12} & 1 & \cdots & \rho_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{1n} & \rho_{2n} & \cdots & 1 \end{pmatrix} \quad \mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1n}\sigma_1\sigma_n \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2n}\sigma_2\sigma_n \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{1n}\sigma_1\sigma_n & \rho_{2n}\sigma_2\sigma_n & \cdots & \sigma_n^2 \end{pmatrix} \tag{4-16}$$

For a complete discussion of VAR(p) and vector autoregressive moving average VARMA(p) processes a good source is (Lütkepohl, 1993).

This section will now describe use of a VAR(1) model to generate multivariate data with a desired mean, correlation, and covariance structure. A first order VAR(1) model derived from eq. 4-14 is defined as:

$$\mathbf{z}_t = \mathbf{\mu} + \mathbf{A}_1\mathbf{z}_{t-1} + \mathbf{\varepsilon}_t \tag{4-17}$$

The autocovariance matrix $\mathbf{\Gamma}(t)$ is a symmetric $n \times n$ matrix of correlations across features i and j measured between t time lags. If $t = 0$, then $\mathbf{\Gamma}(0) = \mathbf{\Sigma}$. If the features are standardized with $\mathbf{\sigma} = \mathbf{1}$, then $\mathbf{\Gamma}(0) = \mathbf{\Sigma} = \mathbf{R}$. The *Yule-Walker* equations (Lütkepohl,

1993:21) can be used to compute $\mathbf{\Gamma}(t)$ recursively if $\mathbf{A}_1$ and $\mathbf{\Sigma}$ are known, and are

presented as equations 4-18 and 4-19,

$$\mathbf{\Gamma}(0) = \mathbf{A}_1\mathbf{\Gamma}(-1) + \mathbf{\Sigma}_\varepsilon = \mathbf{A}_1\mathbf{\Gamma}(1)' + \mathbf{\Sigma}_\varepsilon \qquad \text{for } t = 0 \qquad (4\text{-}18)$$

$$\text{and} \quad \mathbf{\Gamma}(t) = \mathbf{A}_1\mathbf{\Gamma}(t-1) \qquad \text{for } t > 0. \qquad (4\text{-}19)$$

The multivariate Gaussian data observations may be standardized to be unitless with $\mu_i = 0$ and $\sigma_i = 1 \; \forall \, i = 1, 2 \ldots n$ of the multivariate features via,

$$z'_i = \frac{z_i - \mu_i}{\sigma_i} \quad \text{or in matrix notation} \quad \mathbf{z}' = (\mathbf{z} - \mathbf{\mu})^T \mathbf{D}^{-1/2}, \qquad (4\text{-}20)$$

where $\mathbf{D}$ is the $n \times n$ matrix of feature variances,

$$\mathbf{D} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix}$$

and may then be transformed back to the desired units using,

$$z_i = \mu_i + \sigma_i z'_i \text{ or in matrix notion } \mathbf{z} = \mathbf{\mu} + \mathbf{z}'\mathbf{D}^{1/2}. \qquad (4\text{-}21)$$

It is then sufficient to generate a standardized VAR(1) series with the desired level of

autocorrelation and across feature correlation structure for each desired population or

target class. This standardized data may then be transformed to obtain the desired feature

means and covariance structure using (4-21). Using standardized data, equation (4-17) is

reduced to $\mathbf{z}_t = \mathbf{A}_1\mathbf{z}_{t-1} + \mathbf{\varepsilon}_t$ and $\mathbf{\Gamma}(0) = \mathbf{\Sigma} = \mathbf{R}$. The lag 1 autocovariance matrix $\mathbf{\Gamma}(1) = $

$\mathbf{R}(1)$ is a matrix of correlation values across 1 time step and includes each feature's

autocorrelation on the main diagonal. To generate data with desired correlation $\mathbf{R}$ and

lag 1 autocorrelation and crosscorrelation $\mathbf{R}(1)$, starting with eq. 4-18 $\mathbf{A}_1$ is obtained as

131

$\mathbf{R}(1)\mathbf{R}(0)^{-1}$. The corresponding covariance of the VAR series $\mathbf{z_t}$ is $\mathbf{\Sigma}_z = \mathbf{A}_1 \mathbf{\Sigma}_z \mathbf{A}_1^T + \mathbf{\Sigma}_\varepsilon$.

For the VAR process to be feasible, $\mathbf{\Sigma}_\varepsilon = \mathbf{\Sigma}_z - \mathbf{A}_1 \mathbf{\Sigma}_z \mathbf{A}_1^T$ must result in a positive

semidefinite matrix for the white noise process to have a feasible constant variance

structure (Duda *et al.,* 2001: 618).

With a defined positive definite covariance matrix $\mathbf{\Sigma}_\varepsilon$, Cholesky decomposition

can be used to generate random vectors from a Gaussian distribution with mean $\mathbf{\mu}$ and

covariance $\mathbf{\Sigma}$. A single observation can be generated starting with a vector of *n* i.i.d.

~norm(0,1) RVs, such that $\mathbf{E_t} = (E_1, E_2,\ldots, E_n)^T$ with associated $\mathbf{\mu}_E = \mathbf{0}$ and $\mathbf{\Sigma}_E = \mathbf{I}$, the

identity matrix. Since covariance and correlation matrices are symmetric and positive

definite they can be factored as (Strang, 1988: 195),

$$\mathbf{\Sigma} = \mathbf{LDL}^T = (\mathbf{LD}^{1/2})(\mathbf{LD}^{1/2})^T = \mathbf{CC}^T. \tag{4-22}$$

Matrix $\mathbf{C}$ is known as the Cholesky decomposition or matrix "square root" of $\mathbf{\Sigma}$, $\mathbf{L}$ is a

lower triangular matrix and $\mathbf{D}$ is a diagonal matrix. Starting with vector $\mathbf{E}$ as described

above, a random vector $\mathbf{z}$ with mean $\mathbf{\mu}$ and covariance structure $\mathbf{\Sigma}$, can be generated as

(Law and Kelton, 2000: 480),

$$\mathbf{z} = \mathbf{\mu} + \mathbf{CE}, \tag{4-23}$$

with $\mathbf{C}$ being the lower triangular Cholesky decomposition of the desired white noise

covariance $\mathbf{\Sigma}$, where $\mathbf{C}\mathbf{\Sigma}_\varepsilon \mathbf{C}^T = \mathbf{CIC}^T = \mathbf{CC}^T = \mathbf{\Sigma}$.

The steps to generate 1 observation with a desired within feature correlation

across 1 time period for *M* observations of *n*-dimensional multivariate data are

summarized below. The process may be repeated for $k = 1\ldots K$ times to represent any

number of classes with different population means, covariance and correlation structures:

1. Set the desired population mean $\boldsymbol{\mu}_k$, variances $\boldsymbol{\sigma}_k$ and correlations: $\mathbf{R}(0)_k$ and $\mathbf{R}(1)_k$.
2. Generate M standardized random starting observations using $\tilde{\mathbf{z}}_m = \mathbf{CE}$, for $m = 1...M$ with $\mathbf{CC}^T = \mathbf{R}(0)_k$ and $\mathbf{E} = (E_1, E_2,..., E_n)^T$ where E_i is i.i.d. ~norm$(0,1)$.
3. Using equation 4-17 let $\mathbf{A}_1 = \mathbf{R}(1)\mathbf{R}(0)_k^{-1}$
4. For each of M observations generate $\mathbf{E}_t = (E_1, E_2,..., E_n)^T$ where E_i is i.i.d. ~norm$(0,1)$
5. Let $\boldsymbol{\varepsilon}_t = \mathbf{C}_k\mathbf{E}_t$ to induce the desired correlation structure in the white noise $\boldsymbol{\varepsilon}_t$, where $\mathbf{C}_k\mathbf{C}_k^T = \boldsymbol{\Sigma}_\varepsilon = \mathbf{R}(0)_k - \mathbf{A}_1\mathbf{R}(0)_k\mathbf{A}_1^T$
6. For each observation let $\tilde{\mathbf{z}}_{t+1} = \mathbf{A}_1\tilde{\mathbf{z}}_t + \boldsymbol{\varepsilon}_t$ to obtain a new observation across 1 time step with standardized unit variance while maintaining the desired correlation structure.
7. For each observation transform the standardized data using $\mathbf{z}_t = \boldsymbol{\mu}_k + \tilde{\mathbf{z}}_t\mathbf{D}_k^{1/2}$ to obtain the desired class mean and covariance.

While the preceding steps can be used to generate data with given covariance and correlation, two areas of caution should be considered. First, given a covariance structure $\boldsymbol{\Sigma}$, not all lag 1 correlation structures $\mathbf{R}(1)$ are feasible. Arbitrary assignment of desired $\mathbf{R}(1)$ values may not be feasible, but if $\mathbf{R}(1) = \rho\mathbf{R}(0)$ where ρ is a scalar constant of a desired positive correlation $(0 < \rho < 1)$, a feasible solution is guaranteed. Solving eq. 4-18, $\mathbf{A}_1 = \mathbf{R}(1)\mathbf{R}(0)^{-1} = \rho\mathbf{R}(0)\mathbf{R}(0)^{-1} = \rho\mathbf{I}$, thus $\mathbf{A}_1$ is a diagonal matrix of ρ, and the associated covariance of the VAR(1) process is $\mathbf{A}_1\mathbf{R}(0)\mathbf{A}_1 + \boldsymbol{\Sigma}_\varepsilon = \rho^2\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_\varepsilon$. The VAR(1) process will then have stationary covariance $\boldsymbol{\Sigma}$ if white noise is generated as $\boldsymbol{\varepsilon}_t = \mathbf{B}\boldsymbol{\Sigma}$, where $\mathbf{B}$ is a diagonal matrix of $\sqrt{1-\rho^2}$, with covariance $\boldsymbol{\Sigma}_\varepsilon = \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}$. The VAR(1) process covariance is $\mathbf{A}_1\boldsymbol{\Sigma}\mathbf{A}_1 + \mathbf{B}\boldsymbol{\Sigma}\mathbf{B} = \rho^2\boldsymbol{\Sigma} + (1-\rho^2)\boldsymbol{\Sigma} = \boldsymbol{\Sigma}$. In addition, the white noise covariance matrix $\boldsymbol{\Sigma}_\varepsilon$ will be positive definite and can be factored using Cholesky decomposition, since it is a scalar multiple of the positive definite matrix $\boldsymbol{\Sigma}$,

with positive eigenvalues λ (Strang, 1988: 245). Thus, Σ_ε will be positive definite with positive eigenvalues $\alpha\lambda$, since $\Sigma_\varepsilon \mathbf{x} = \alpha\Sigma\mathbf{x} = \alpha\lambda\mathbf{x}$ for any fixed vector $\mathbf{x}$ and arbitrary constant α.

If a limited amount of data observations are to be generated, the correlation values should be estimated to determine if the data are adequate for the research experiment to be accomplished. For a multivariate distribution, the estimate of correlation between two features, $\hat{\rho}$, has a standard error approximated by $\sigma_{\hat{\rho}} \cong \dfrac{(1 - \hat{\rho}^2)}{\sqrt{n}}$ where n is the number of samples generated (Schmeizer, 1990: 311). Thus, to obtain two-place accuracy for $\hat{\rho}$, 10,000 data points may be required to generate data with a desired correlation level. If a small sample of data is generated, unacceptable levels of random variability are possible. These small data generation sets may require multiple sets be created, to obtain one set with correlation levels within a desired tolerance or to test fusion algorithms on multiple test sets. For example if a single time step is used to create 100 additional correlated "looks" with a desired correlation of 0.1, $\sigma_{\hat{\rho}}$ would be 9.9%, while a desired correlation level of 0.9 would have a standard error of 1.9%. If only 25 observations are generated the standard error increases to 19.8% and 3.8% respectively.

Overall, the generation of multivariate Gaussian data can be performed quickly, but the levels of desired correlation obtained from an initial random vector may vary significantly depending on sample size, and even vary as a function of the desired levels of correlation across variables and correlation through time within each variable. Finally, if information is available that suggests use of a parametric distribution other than a

Gaussian should be used to model a specific sensor's features, the literature offers several techniques to generate different distributions with desired correlation structures. Most techniques involve the generation of normally distributed random variables for error terms, which are then combined with previously generated data points through a linear transformation. Some of the available techniques are found within (Song and Hsiao, 1993), (Nelson and Yamnitsky, 1998), (Deler *et al.*, 2001), and (Cario and Nelson, 1996, 1998). Use of other generated parametric distributions with desired levels of correlation, would also yield samples with observed correlation significantly affected by the sample size and desired correlation levels.

4.3 Generated Gaussian Two Class Fusion Experiments

These experiments will demonstrate the utility of the mathematical programming methodology introduced in Chapter 3 to optimize rejection and ROC thresholds given decision maker preferences and operational constraints. The maximum P_{TP} or *TPR* will be used to assess the effects of correlation in a predetermined fusion process. By performing this research insight may be gained for fusion in an ATR system where "Non-declaration" is a valid output label, and when the source of data being fused from different sensors may be correlated at various levels. This initial fusion research using generated Gaussian data was presented at three conferences, including the SPIE sponsored Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2004 where the initial application of the optimization framework was presented (Laine and Bauer, 2004a). Further experiments fusing additional looks of Gaussian data collected through time were the presented at the 72nd Military Operations

Research Society (MORS) Symposium in 2004 and at the ATR Systems and Technology

Symposium in 2004. Overall, this research uses the optimization framework of Chapter 3

to explore the fusion of Gaussian scores across various correlation levels and reports

some interesting properties as presented in the next four sections.

4.3.1 Multivariate Gaussian Data Properties

It has been proven (Johnson & Wichern, 1998), if two populations are known

multivariate Gaussian populations with equal covariance Σ, then the optimum error rate

(or minimum Total Probability of Misclassification, *TPM*) given equal misclassification

costs and prior probabilities may be calculated as follows:

$$TPM = \Phi\left(-\frac{\Delta}{2}\right),$$ (4-24)

where, $\Phi(\cdot)$ is the cdf of a standard normal distribution and,

$$\Delta^2 = \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^T \Sigma^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)$$ (4-25)

is the Mahalanobis distance squared between $\boldsymbol{\mu}_1$ & $\boldsymbol{\mu}_2$. Since it has been hypothesized

multi-look ATR information may include significant levels of correlation, it is of interest

to examine the extrema associated with the Mahalanobis distance as a function of ρ. For

bivariate Gaussian data with $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$, differentiating the Mahalanobis distance

with respect to ρ yields:

$$\frac{d\Delta}{d\rho} = \frac{-2(\mu_1\mu_2 + \mu_1\mu_2\rho^2 - \mu_1^2\rho - \mu_2^2\rho)}{(\rho^2 - 1)^2}, \text{ with extrema at } \rho = \frac{\mu_1}{\mu_2} \& \frac{\mu_2}{\mu_1}$$ (4-26)

But, since $\rho \le 1$ for feasible solutions (Duda *et al.*, 2001) only a single solution is obtained. Without loss of generality (*wlog*) let $\mu_1 < \mu_2$. Evaluating the Mahalanobis distance for $\rho = \mu_1 / \mu_2$ yields $\Delta = \mu_2$. Thus, under the assumption of bivariate normality, the *TPM* of two correlated variables is always better than or equal to the univariate *TPM* associated with the better of the two scores. This also leads to an initially non-intuitive property, where the *TPM* associated with two correlated scores may be lower than two independent scores, when the individual variable means are not equal. Yet, as expected, the maximum *TPM* occurs as ρ approaches 1.0 for two scores of equal means. Overall, while the decreased *TPM* associated with high correlation is theoretically feasible in some cases of fusion, it is unknown if real world applications may realize or capitalize on such correlation values.

4.3.2 Fusion of 2 ATR Target Scores Modeled by Gaussian Data

This experiment will identify targets as two labels: targets specified for attack (class 1 "Targets") and non-targets or friends (class 2 "Friends"). As a demonstration of the *TPM* phenomena above, consider a two population experiment with two ATR systems modeled by two Gaussian distributions. The variance is held stationary as ρ varies between 0 and 1. The data for two classes is generated as bivariate Gaussian with

$\mathbf{\mu}_1 = (0,0)^T$, $\mathbf{\mu}_1 = (1.8, 2.2)^T$, and $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. This Gaussian data has a mean

separation of two standard units, resulting in a system with ~84% correct classification accuracy for any single variable. To show the value of fusing a 2nd ATR system look with increased performance, possibly through a decrease in range, the data is generated to

represent two classes with a separation 10% worse than average for the 1st look and 10% better than average for the 2nd look. The associated *TPM* for the 1st look is 0.184 and 0.136 for the 2nd look. Figure 4.1 shows the quadratic nature of the bivariate Gaussian *TPM* as a function of ρ, where the largest *TPM* = 0.136 is obtained when $\rho = 0.818 = 1.8/2.2$.
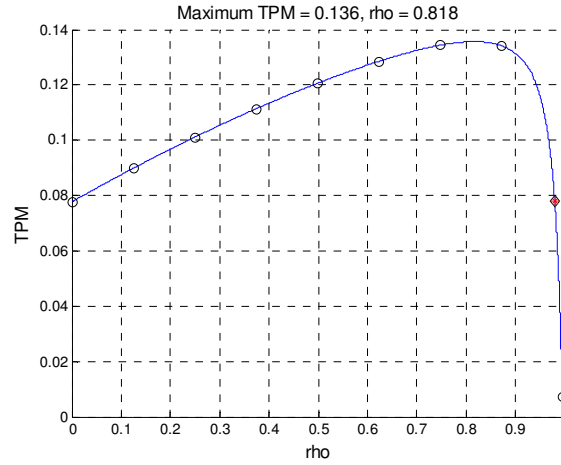
Figure 4.2 Total Probability of Misclassification (*TPM*) as a Function of Correlation

(ρ = rho) for given μ_1, μ_2 and $\Sigma_1 = \Sigma_2$

To show this phenomenon geometrically, the Fisher Discriminant line representing the optimal class boundary (Duda *et al.*, 2001) is plotted in Figure 4.3 for the values of ρ identified by circles in Figure 4.2 (ρ varies between 0.0 and 0.992 across increments of 0.124). It is of interest to note, that while some research (Dudgeon, 1998) identifies independence of fused data as generally the limiting case of performance; theoretically, an increase in performance may be obtained in some cases of very high correlation due to the quadratic nature of the Mahalanobis distance, as demonstrated when $\rho > 0.98$, since *TPM*$_{rho=0.98} \cong$ *TPM*$_{rho=0.0} \cong 0.078$. This agrees with finding by

138

Willett, *et al.* (2000), who note correlation levels may hinder or help a classification effort depending on the location of class means for multivariate Gaussian populations.

Bayes optimal boundary for $mu_1=(0,0)^t$ $mu_2=(1.8,2.2)^t$ | rho
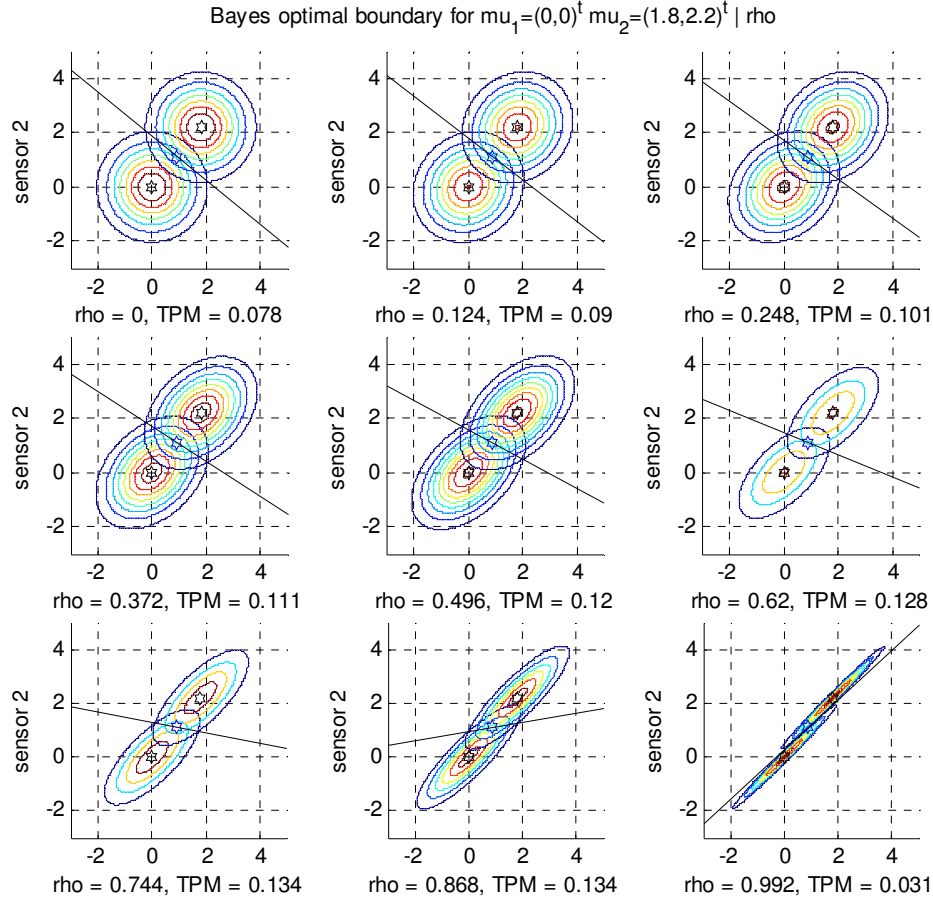
Figure 4.3 Fisher Discriminant Lines for Optimal Class Boundaries with the Minimum Total Probability of Misclassification (*TPM*) as a Function of Correlation (ρ = rho) for Specified Multivariate Gaussian Populations

4.3.3 Results Obtained using Optimization Framework

For the 2-D Gaussian data generated, the fusion strategy will maximize the probability of True Positive Target Declarations, $P_{TP}(\theta)$, subject to decision maker constraints as outlined in eq 4-1. Since this investigation seeks to discover differences

across data correlation, only a single fusion algorithm is used. The thresholds were optimized for test data generated across the desired correlation structures. The first of two ATR scores is generated as the posterior probability of class membership obtained from a single value of 1-D Gaussian data with known distribution parameters. If the 1st ATR score is not declared as a "Target" or "Friend" a 2nd score is then obtained. The posterior probabilities of the second score are then evaluated using the score obtained from both the 1st and 2nd look. By performing the fusion in this manner, a maximum of information is preserved and used for the final decision.

Thirty projected 2-D ROC curves are presented in each subplot of Figure 4.4. Each ROC curve was generated using 30 uniformly spaced ROC thresholds, for each of 30 different rejection thresholds. The test data included 20K multivariate Gaussian data points with 9 levels of correlation. In all subplots, the benefit of allowing a 2nd look is illustrated by comparing the single lower ROC curve associated with the 1st look and no reject option, with improvement observed after allowing any 2nd look for the "non declared" observations. In general, the ROC improvements are observed as the dark region in the upper left-hand area of each plot, representing the projection of 29 ROC curves onto the subplot. While improvements are clearly seen after allowing "Non-declarations" ($\theta_{REJ} > 0$), further visual analysis is difficult. For example, differences between ROC curves when the declaration threshold, θ_{REJ} , is above 0.0 blend together, and the identification of a preferred ROC and declaration threshold associated with a visual 'knee' in the ROC curve is difficult to identify. In addition, for the case of $\rho =$ 0.992, all 30 ROC curves appear to project onto either of two curves representing the

ROC curve with no 2nd look ($\theta_{REJ} = 0$) and the alternative case where any second look

($\theta_{REJ} > 0$) yields an almost perfect ROC curve.
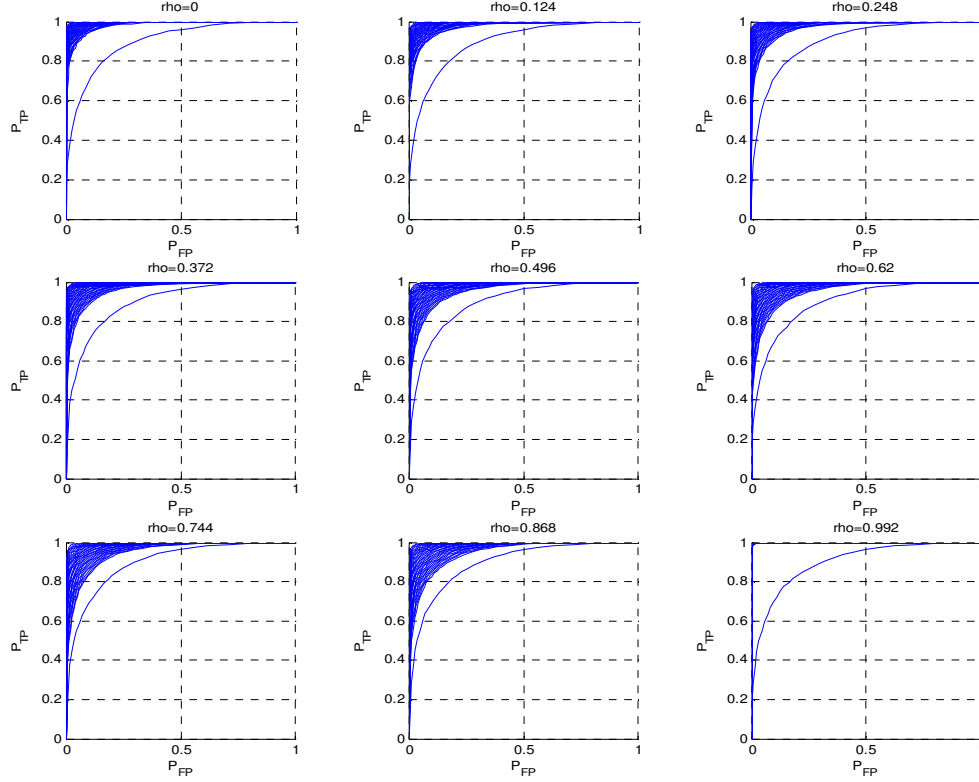
Figure 4.4 Thirty Projected ROC Curves Generated using 30 Uniformly Spaced ROC Thresholds for each of 30 Uniformly Spaced Rejection Thresholds for 20K Multivariate Gaussian Data Observations with 9 Levels of Correlation (ρ = rho)

To determine the feasible and optimal thresholds: $\boldsymbol{\theta} = (\theta_{low}, \theta_{up})^T$, with respect to the decision maker preferences presented in eq. 4-1, $P_{TP}(\boldsymbol{\theta})$ is maximized with the constraints shown in eq. 4-27.

$$\max_{\theta \in \Theta} P_{TP}(\theta) \quad \text{Maximize probability of true positive declarations} \quad (4\text{-}27)$$

s.t. $\quad E_{CR}(\theta) < 0.02 \quad\quad$ limit potential friendly fire

$\quad\quad\quad E_{NC}(\theta) < 0.05 \quad\quad$ limit lost opportunities to engage the enemy

$\quad\quad\quad P_{Dec}(\theta) > 0.70 \quad\quad$ limit the number of re-looks & Non-declarations

Plotting $P_{TP}(\theta)$ and $P_{FP}(\theta)$ from Figure 4.4 along with $P_{Dec}(\theta)$ leads to the ROC surfaces

in Figure 4.5. Feasible points, meeting all decision maker constraints, are then identified

by the dark areas. The optimal thresholds maximizing $P_{TP}(\theta \mid \rho)$ are identified in Tables

4.2 and 4.3 for two ratios of prior probabilities. In addition to generating the 3-D ROC

surfaces in Figure 4.5, similar plots were examined across a range of prior probabilities

where $P_T{:}P_F = 1{:}4$ through 4:1. As expected, if limiting feasible points to include the

entire range of priors, additional constraints are imposed and fewer viable operating

thresholds are obtained. These points tended to emerge on the classical "knee" in the

ROC curve.

Table 4.2 Performance Measures of the 3-D ROC Surfaces Obtained from 20K
Generated Data Observations for $P_T{:}P_F = 4{:}1$

ρ	% feas	max P_{TP}	P_{FP}	E_{CR}	E_{NC}	P_{Dec}	θ_{REJ}	θ_{low}	θ_{up}
0.000	4.60%	99.79%	11.34%	1.65%	1.60%	70.87%	0.585	0.014	0.599
0.124	1.83%	99.55%	10.80%	1.69%	3.08%	71.03%	0.630	0.025	0.655
0.248	0.17%	99.29%	11.31%	1.84%	4.63%	70.20%	0.630	0.037	0.667
0.372	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.496	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.620	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.744	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.868	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.992	87.1%	100.00%	0.11%	0.02%	0.00%	86.60%	0.90	0.010	0.910
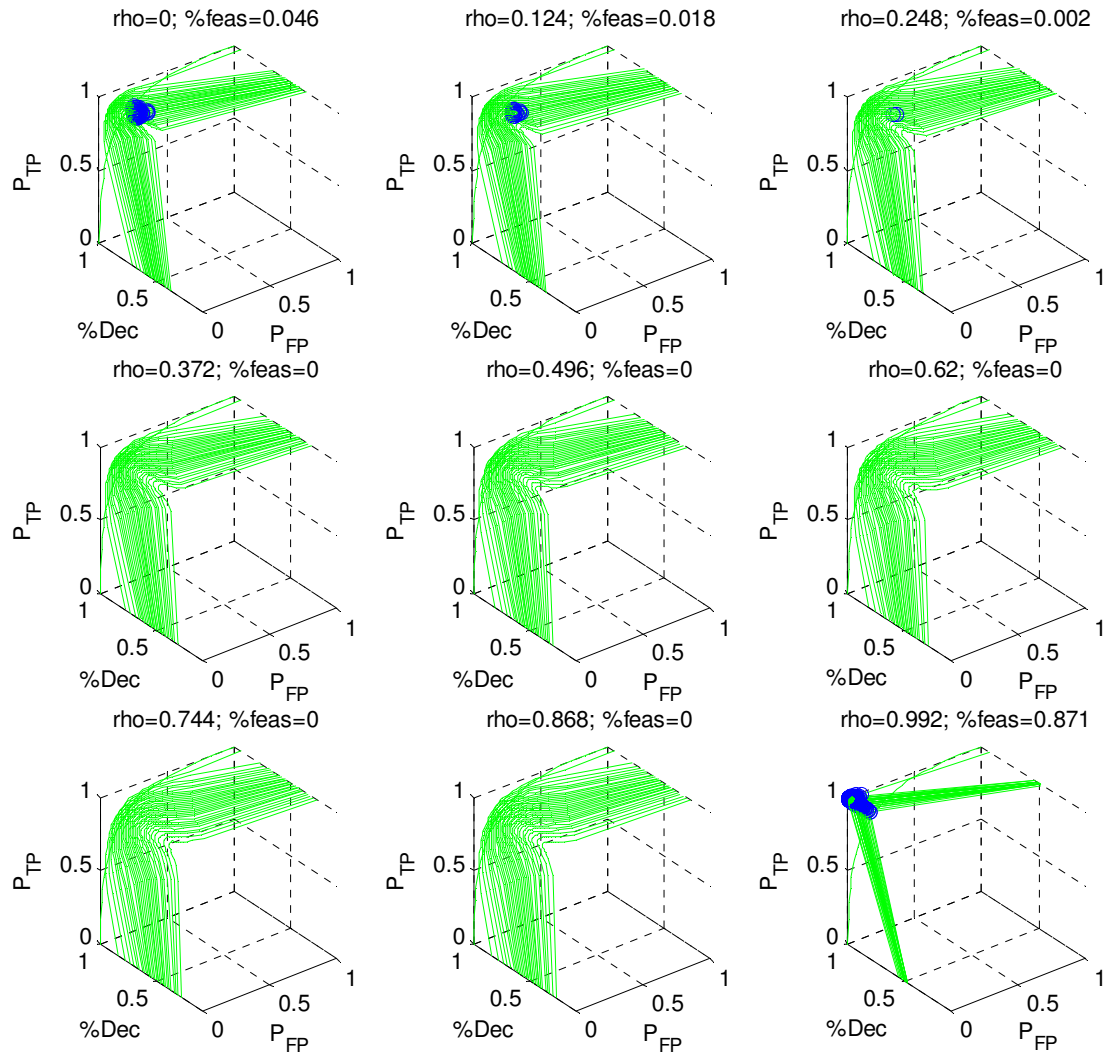
Figure 4.5 ROC Surfaces with Feasible Points (%feas) Identified by Dark Areas for

20K Data Observations across 9 Levels of Correlation (ρ = rho) for Specified

Multivariate Gaussian Populations with Prior Probabilities, P_T:P_F = 4:1

Table 4.3 Performance Measures of the 3-D ROC Surfaces Obtained from 20K

Generated Data Observations for $T{:}F = 1{:}4$

ρ	% feas	max P_{TP}	P_{FP}	E_{CR}	E_{NC}	P_{Dec}	θ_{REJ}	θ_{low}	θ_{up}
0.000	0.50%	87.41%	0.24%	1.89%	1.79%	71.39%	0.585	0.401	0.986
0.124	0.67%	83.52%	0.22%	1.79%	2.33%	71.79%	0.495	0.488	0.983
0.248	0.17%	77.69%	0.22%	1.99%	3.04%	70.74%	0.450	0.532	0.982
0.372	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.496	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.620	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.744	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.868	0.00%	none	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.992	65.67%	100.00%	0.11%	0.38%	0.00%	86.60%	0.90	0.010	0.910

With a target sparse environment ($T{:}F = 1{:}4$) conservative threshold settings below the

"knee" in the ROC surface were obtained. For a target rich environment ($T{:}F = 4{:}1$)

more aggressive threshold settings were feasible with points slightly above the "knee" in

the ROC surface, as can be seen in the top 2 subplots of Figure 4.5 with lowest

correlation.

4.4.4 Two Sensor Multilook Fusion Experiment with Gaussian Data

After the initial use of the optimization framework for the limited two-look

example, a subsequent natural extension involved applying the optimization framework

to a scenario in which two ATR systems were fused across multiple looks. Since the

ATR system performance improved considerably by fusion of the two ATR scores, the

multi-look experiment in this section forces two ATR looks at each time period.

Gaussian data was generated across ATR correlations and across time for up to 10 looks

using the procedures outlined in Section 4.2.2. The data was generated to represent two

sensors with equal performance, where $\mathbf{\mu}_1 = (0,0)^T$, $\mathbf{\mu}_2 = (2.0, 2.0)^T$, the covariance is

equal for both ATR systems, where $\Gamma(0) = \Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & \rho_x \\ \rho_x & 1 \end{pmatrix}$, and for looks across

one time lag, $\Gamma(1) = \rho_t \Gamma(0) = \begin{pmatrix} \rho_t & \rho_t \rho_x \\ \rho_t \rho_x & \rho_t \end{pmatrix}$ with $\rho_t \ \& \ \rho_x \in \{0.0, 0.24, 0.48, 0.72, 0.96\}$.

The Theoretic Total Probability of Misclassification (*TPM*) modeled by each 1-D

Gaussian ATR system is ~15%, and the 2-D *TPM* associated with the fusion of the two

systems each taking 1-look varies from 7.8% to 15.6% as the correlation (ρ_x) varies

between the ATR systems from 0 to 0.96. The associated *TPM* using all 10-looks for

each of the two ATR scores is presented in Table 4.4. From this table, low

misclassification levels, < 2%, are observed for all correlation less than $\rho_x = 0.48$ across

systems. Highly desirable misclassification, < 0.5%, is highlighted in bold, while the

least desirable *TPM*, > 10%, is indicated by the gray background.

Table 4.4 Theoretic Probability of Total Misclassification as a Function of Sensor
Correlation and Autocorrelation with 10 Looks

Across Correlation	Equal feature means **TPM for 10 looks** *Temporal Correlation*				
	0	**0.24**	**0.48**	**0.72**	**0.96**
0	0.0%	0.0%	0.0%	0.0%	0.1%
0.24	0.0%	0.1%	0.2%	0.3%	0.5%
0.48	0.2%	0.5%	0.9%	1.4%	2.0%
0.72	1.3%	2.3%	3.4%	4.5%	5.6%
0.96	6.2%	8.4%	**10.3%**	**12.0%**	**13.6%**

With data generated using known Gaussian parameters, the fused ATR scores

were computed directly from the two Gaussian scores associated with each ATR system.

For a single look the posterior probability of being a hostile target was computed by

normalizing the two probability estimates associated with either of the two classes using a 2-D multivariate Gaussian pdf. After the first look had occurred, posterior probabilities were generated in three different manners to assess the value of information through time. The first method always used a 2-D Gaussian approximation, where only the current ATR system scores were fused to determine a combined score. If a fused system label was "Non-declaration" another two looks would be taken. The second method generated a fused posterior probability score by using the two most current looks from each system. This required using a 4-D multivariate Gaussian distribution to represent the associated probability of Hostile vs. Friendly class membership. The final method of generating a fused posterior ATR score used all available ATR system scores including the current look. Thus, a $2 \times n - looks$ Gaussian distribution was used obtain the final ATR score, with the potential to reach the low levels of *TPM* reported in Table 4.7 if all 10 looks were used. The True Positive declaration rate was then maximized subject to the constraints identified in eq. 4-28.

$$\underset{\theta \in \Theta}{\arg \max} \, TPR(\theta) \quad \text{Maximize true positive declaration rate} \tag{4-28}$$

s.t. $\quad E_{CR}(\theta) < 0.02 \qquad$ limit potential friendly fire

$\qquad\quad E_{NC}(\theta) < 0.05 \qquad$ limit lost opportunities to engage the enemy

$\qquad\quad P_{Dec}(\theta) > 0.70 \qquad$ limit the number of re-looks & Non-declarations

To perform the fusion across the two ATR systems, the following sequential fusion strategy was implemented using the three different posterior probability scores.

1. Vary $\theta_{low} = \theta_{ROC}$ and $\theta_{up} = \theta_{ROC} + \theta_{REJ}$ uniformly across the feasible range, where a constant θ_{REJ} yields a single ROC curve.

2. Attempt to classify 4,000 generated potential targets using the 1st fused ATR score, with the posterior probability of "Hostile" target, derived from the 2-D Gaussian data.

- If $ppH < \theta_{low}$ declare as "Friend".

- If $ppH > \theta_{up}$ declare as "Hostile".

- If $\theta_{low} < ppH < \theta_{up}$ declare as "Non-declaration" and obtain another look from each ATR system to generate a fused ATR score.

- Continue until current target is declared "Hostile" or "Friend" or until maximum (10th) ATR score is used.

- Using the same values of θ_{low} and θ_{up} attempt to classify all objects.

3. Identify feasible points across all θ_{low} and θ_{up}.

4. Determine the optimal thresholds associated with the maximum $TPR(\theta)$.

The next figure shows the collection of ROC curves generated using all available ATR scores to generate a final ATR system score for the lowest correlation levels on the left and the highest correlation levels of the right. Each individual ROC curve is generated from a different value of θ_{REJ} as θ_{ROC} varies. The black region in the left plot shows feasible P_{TP} and P_{FP} values associated with feasible thresholds and a star shows where the maximum $TPR(\theta)$ is achieved.
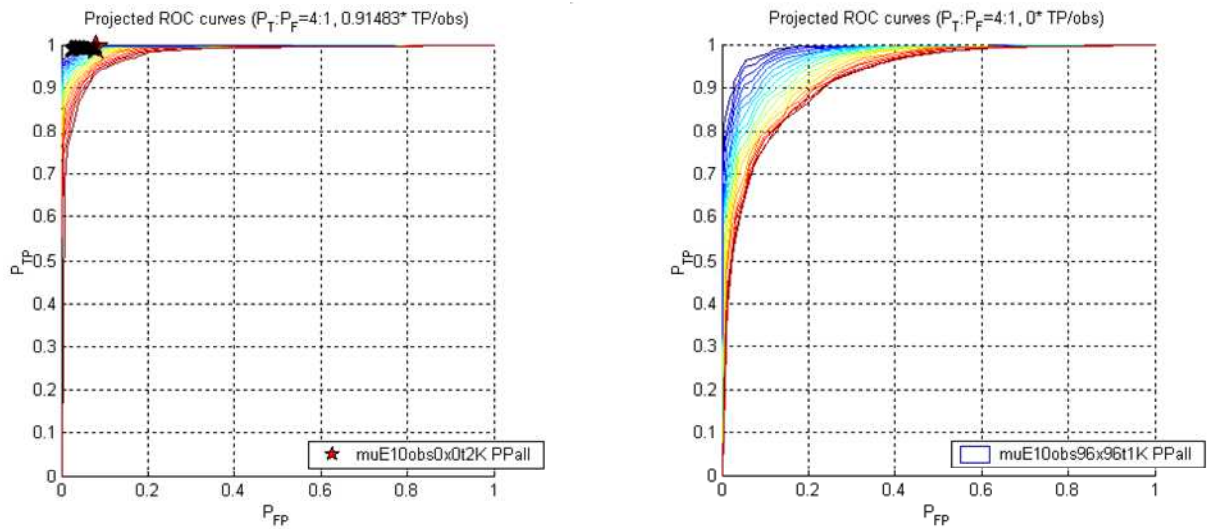
Figure 4.6 ROC Curves for Lowest (0,0) vs. Highest Correlation Levels (0.96, 0.96)

147

Figure 4.7 shows the associated ROC surfaces by plotting the ROC curves from Figure 4.6 along with the associated probability of declaration. The black cluster of points shows where feasible thresholds are obtained in the left plot with the lowest level of correlation. The right plot, with the highest levels of correlation shows the general dispersion of the ROC curves generated across different rejection thresholds with no feasible points.
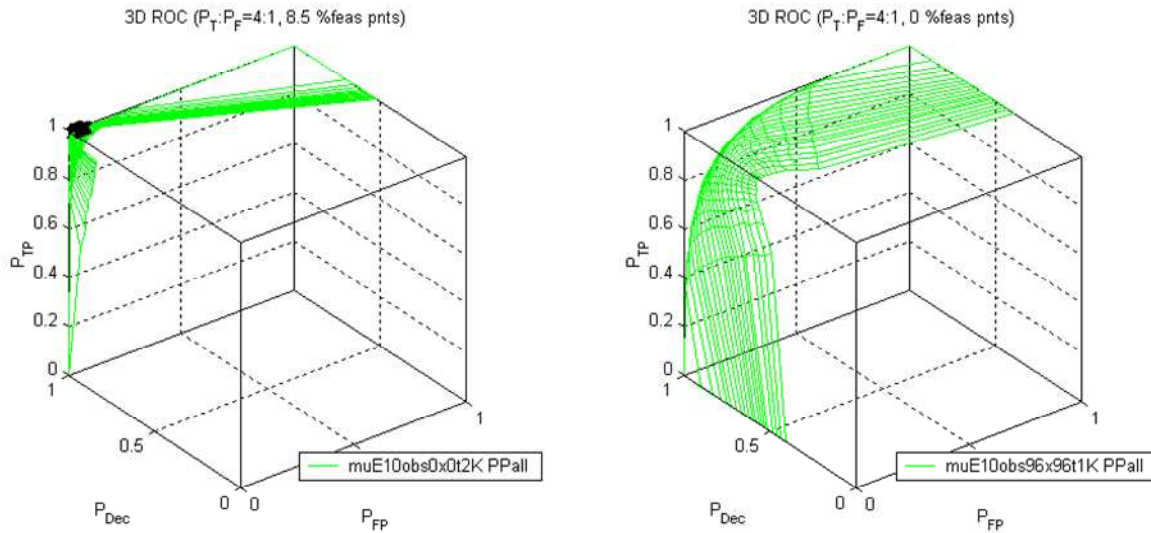
Figure 4.7 ROC Surfaces for Lowest (0,0) vs. Highest Correlation Levels (0.96, 0.96), across ATR Systems and through Multiple Looks

A summary for all three different methods to generate the final fused ATR system score is included in Table 4-5. Both the maximum $TPR(\theta)$ obtained and the percentage of feasible thresholds evaluated is included for each correlation structure. Significant performance degradation is indicated for high levels of correlation across all three techniques. The infeasible correlation structures are indicated by the '---' max TPR and an associated light gray 0% for % Feasible. From these tables, significant feasibility improvement is observed when using the information from more of the available looks to

assess the current posterior probability, as indicated by an increased number of feasible correlation structures and an increase in the percentage of feasible evaluated thresholds. If all three methods were feasible, limited differences in the maximum obtainable $TPR(\theta)$ were observed. Thus, the primary advantage for incorporating all available ATR scores to generate the current class estimate is an increase in feasibility, subsequently providing a feasible ID system with a positive max $TPR(\theta)$.

Table 4.5 Maximum *TPR* and Percentage of Feasible Thresholds by Correlation using Posterior Probabilities Generated with 1-look, 2-looks or All *n*-looks

max TPR   **% Feasible**

Posterior Probability derived from 1 looks

Across Correlation	max TPR — Temporal Correlation						% Feasible — Temporal Correlation				
	0	0.24	0.48	0.72	0.96		0	0.24	0.48	0.72	0.96
0	0.77	0.68	---	---	---	0	7.8%	2.5%	0.0%	0.0%	0.0%
0.24	0.65	0.55	---	---	---	0.24	2.5%	0.5%	0.0%	0.0%	0.0%
0.48	0.53	---	---	---	---	0.48	1.0%	0.0%	0.0%	0.0%	0.0%
0.72	---	---	---	---	---	0.72	0.0%	0.0%	0.0%	0.0%	0.0%
0.96	---	---	---	---	---	0.96	0.0%	0.0%	0.0%	0.0%	0.0%

Posterior Probability derived from 2-looks

Across Correlation	max TPR — Temporal Correlation						% Feasible — Temporal Correlation				
	0	0.24	0.48	0.72	0.96		0	0.24	0.48	0.72	0.96
0	0.78	0.70	0.57	---	---	0	10.0%	4.3%	0.5%	0.0%	0.0%
0.24	0.66	0.61	0.52	---	---	0.24	4.8%	2.3%	0.3%	0.0%	0.0%
0.48	0.58	0.48	---	---	---	0.48	1.8%	0.5%	0.0%	0.0%	0.0%
0.72	0.50	---	---	---	---	0.72	0.5%	0.0%	0.0%	0.0%	0.0%
0.96	---	---	---	---	---	0.96	0.0%	0.0%	0.0%	0.0%	0.0%

Posterior Probability derived from n-looks

Across Correlation	max TPR — Temporal Correlation						% Feasible — Temporal Correlation				
	0	0.24	0.48	0.72	0.96		0	0.24	0.48	0.72	0.96
0	0.78	0.72	0.61	0.53	0.29	0	11.5%	6.5%	2.8%	2.8%	3.0%
0.24	0.67	0.62	0.55	0.33	0.15	0.24	5.8%	3.8%	1.8%	0.8%	0.8%
0.48	0.57	0.52	0.40	---	0.15	0.48	2.0%	1.8%	0.3%	0.0%	0.5%
0.72	0.51	---	---	---	---	0.72	1.0%	0.0%	0.0%	0.0%	0.0%
0.96	0.46	---	---	---	---	0.96	0.5%	0.0%	0.0%	0.0%	0.0%

To help answer the question of, "What is preferred, high TP obtained by fusing independent data or a lower TP obtained in less time by fusing more correlated data?" Table 4.6 was produced to show relative *TPR equivalence.* This provides a rough

assessment of an ATR system's operational utility, by providing the relative number of looks required to make a positive Hostile ID when compared to the best system. Since the best $TPR(\theta)$ is achieved by data independent across sensors and through time, the upper left hand value with $\rho_x = \rho_t = 0$ is used to scale all other $TPR(\theta)$ scores associated with different correlation structures. These values were generated using a Hostile:Friend ratio of 1:1 and 4:1 and all available n-looks of ATR scores. From this table, two observations are made. First, when the ratio of H:F is increased, an increase in feasible correlation structures occurs. Next, if highly correlated feasible data can be obtained more quickly than less correlated data, it may be preferred. For example, if the data associated with $\rho_x = \rho_t = 0.48$ can be obtained in half the time as the data with

$\rho_x = \rho_t = 0$, an effective $TPR(\theta)$ would be higher for the correlated data since the time required would be less than the associated time for the independent data. Thus, analysis of the maximum $TPR(\theta)$ or associated looks per true positive hostile ID may be useful analysis and provide insight of preferred data collection strategies by fused ATR systems.

Table 4.6 Temporal Equivalence Indicated by the Number of Looks Required in the Same Time Period used to Collect One Look of Independent Data

Across	H:F=1:1	"Temporal Equivalence" (# of looks required vs. best TPR) Temporal Correlation				H:F=4:1	"Temporal Equivalence" (# of looks required vs. best TPR) Temporal Correlation			
Correlation	0	0.24	0.48	0.72	0.96	0	0.24	0.48	0.72	0.96
0	1.00	1.09	1.29	1.47	2.66	1.00	1.02	1.19	1.23	1.72
0.24	1.17	1.26	1.43	2.34	5.18	1.09	1.12	1.25	1.58	2.21
0.48	1.37	1.50	1.94	---	5.19	1.20	1.30	1.43	2.10	3.10
0.72	1.54	---	---	---	---	1.34	1.48	1.87	---	---
0.96	1.68	---	---	---	---	1.44	1.61	2.13	---	---

4.3.5 Summary of Gaussian Data Experiment

In these Gaussian data experiments the mathematical programming framework from Chapter 3 was used to optimize thresholds and a 3-D ROC plot was used to help visualize the effects of tuning rejection and ROC thresholds to maximize a decision maker's preferred objective while constrained by other requirements. The 3-D ROC surface was generated by adding the probability of declaration. This methodology may be useful for the comparison of classification algorithms across operating conditions with different potential prior probabilities of class membership, where the percentage of feasible operating thresholds tested can help measure a system's robustness. The mathematical optimization framework can also be easily modified, as was done with the objective function being modified to initially determine the maximum $P_{TP}(\boldsymbol{\theta})$ across correlation levels and then to determine the True Positive rate, $TPR(\boldsymbol{\theta})$, across multiple looks. In addition, some properties were shown for the classification of bivariate Gaussian data, and a justification for modeling ATR scores by Gaussian data was presented.

4.4 Investigation of RNN Fusion using an Optimization Framework

The following experiment applies the mathematical programming framework to compare two RNN models used to fuse sequential data obtained from generated data. Each of the generated input features is representative of the output data from a different sensor. The only differences in the fusion models were the number of input features used for classification. The initial feature saliency research is documented within Laine and Bauer (2003), and was included as an illustrative example of fusion via one-big-net at the

71st Military Operations Research Society (MORS) Symposium, where the paper was selected as the best in the Air Power and Combat ID Analysis Working Group. Reviews have recently been received from a subsequent invited submission to *MOR* (Laine & Bauer, 2005). The demonstrated application of the mathematical programming framework was then documented in Laine and Bauer (2004b), and will be summarized in the following sections.

4.4.1 Overview of Data Generation, Feature Selection and RNN Fusion Model

For this experiment, an ATR system is simulated and allowed to obtain up to 10 looks of each object known to be a satellite of class "Target" or "Friend." The objective is to identify as many enemies as possible with a limited sensing resource, constrained by allowable false IDs. The generated data was inspired from data collected for 2 geosynchronous satellite types observed through time and processed by a Johnson filter. The real data included the magnitude, corrected for distance, in red and blue frequency bands, with temporal trends associated with the rotation of the earth, reflection from the sun and other atmospheric effects. Three features were generated from a known parabolic "red" signature corrupted with 3 levels of noise. Similarly, 3 features were generated from a decreasing logarithmic "blue" signature. Since the data were generated as continuous functions of time with noise added, autocorrelation was statistically significant, as was crosscorrelation between variables derived from the same "color". Two "noise" features were constructed with no difference between classes. No feature provided linear separation of classes. An example of the underlying "truth" signal functions and data with the lowest level of noise follows in Figure 4.8.
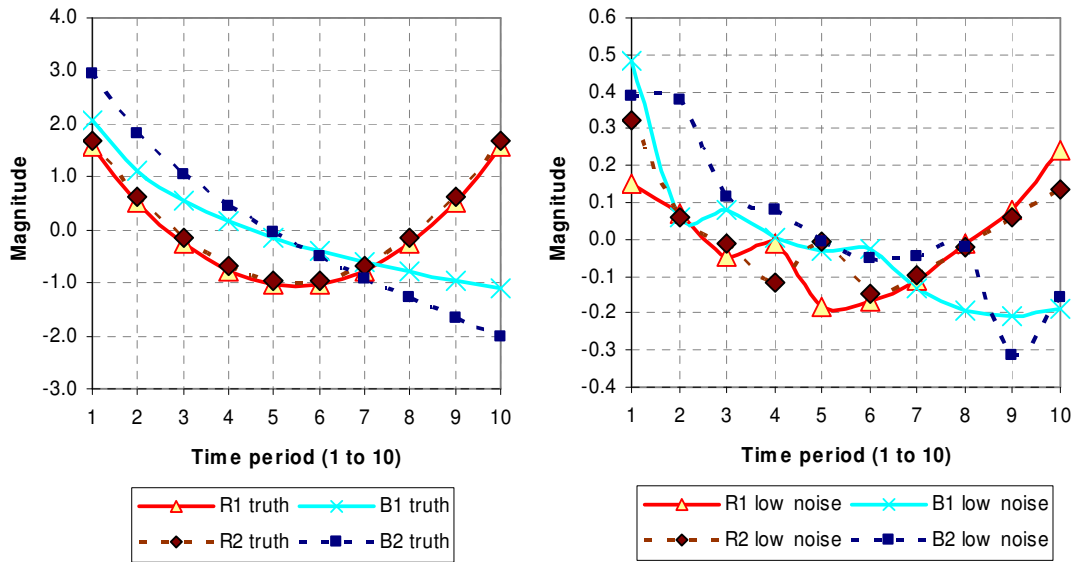
Figure 4.8 Truth and Low Noise Data for "red" (Parabolic Pattern) and "blue" (Nonlinear Decreasing) Features for Target 1 (R1 & B1) and Target 2 (R2 & B2)

Eight total input features, representative of the data obtained by eight different sensors, were generated for 10 time units each. Data sets were comprised of 10 sequences of each class, resulting in 200 total observations in each data set. Twenty data sets were generated for use as Training, training-Test, and Validation sets. Training data with all available observations was used to calculate error and update network weights, the training-Test set was used to assess the trained RNN to stop training before over-fitting occurred, and the validation set was held as an independent test set to assess the RNNs ability to generalize.

Since a strong temporal component may be hypothesized for an ATR system processing multiple looks in close spatial-temporal proximity, a neural fusion model was sought with the capability to fuse input features with a single architecture with any number of re-looks. An Elman RNN (Elman, 1990) was selected since it includes

internal feedback and the ability to model temporal patterns without restrictions on the

structure of input data correlation or number of temporal samples obtained (Kolen &

Kremer, 2001). The RNN's input features consisted of either all 8 input features or a

parsimonious subset of 3 features, as determined by Laine and Bauer (2003). The

reduced features were determined with both a Signal-to-Noise weight based saliency

measure (Bauer *et al*., 2000) and an output error based saliency measure (Moody, 1998).

The reduced features included "red" & "blue" features with low noise plus "blue" with

medium noise.

The experiment was performed using *Matlab 6.1* with the *Neural Network*

Toolbox. RNNs were initialized with 8 hidden nodes and 2 output nodes with hyperbolic

tangent and sigmoid transfer functions respectively. The desired outputs were set to 0.9

and 0.1 for correct and incorrect classes. All networks were trained using gradient

descent with momentum and an adaptive learning rate for a maximum of 2500 epochs.

Most training stopped early after the training-Test set MSE failed to improve after 500

epochs. The RNN weights associated with the minimum training-Test set MSE were

retained to be used as the trained fusion model.

The fusion strategy attempts to maximize the probability of true positive target

declarations per time, $TPR(\theta)$, subject to constraints as outlined in eq. 4-1. The tth ATR

score for a potential target was generated as the posterior probability of class membership

obtained from the outputs of a trained RNN using the current input exemplar and the

previous *t-1* input features. For example, if the 2nd observation is not declared as a

"Target" or "Non-Target" a 3rd observation is obtained for RNN input data and updated

posteriors are obtained. Class updates continue until a declaration is made or the 10th

observation is left "undeclared." Thirty Training and Test set ROC curves are presented in each subplot of Figure 4.9. Each curve was generated as a function of 30 uniformly spaced ROC thresholds, repeated for 30 different rejection thresholds. The left plot in Figure 4.9 is created from Training data, while the right plot, without feasible points, is generated from Test data, with $P_T:P_F = 1:1$. An "O" is plotted for each feasible threshold vector. A "star" is plotted for the optimal $TPR(\theta) = 0.61$ for the Training data on the left. The output data from 20 trained RNNs were combined to provide 400 data sequences with an equal number of class samples to generate each ROC curve.
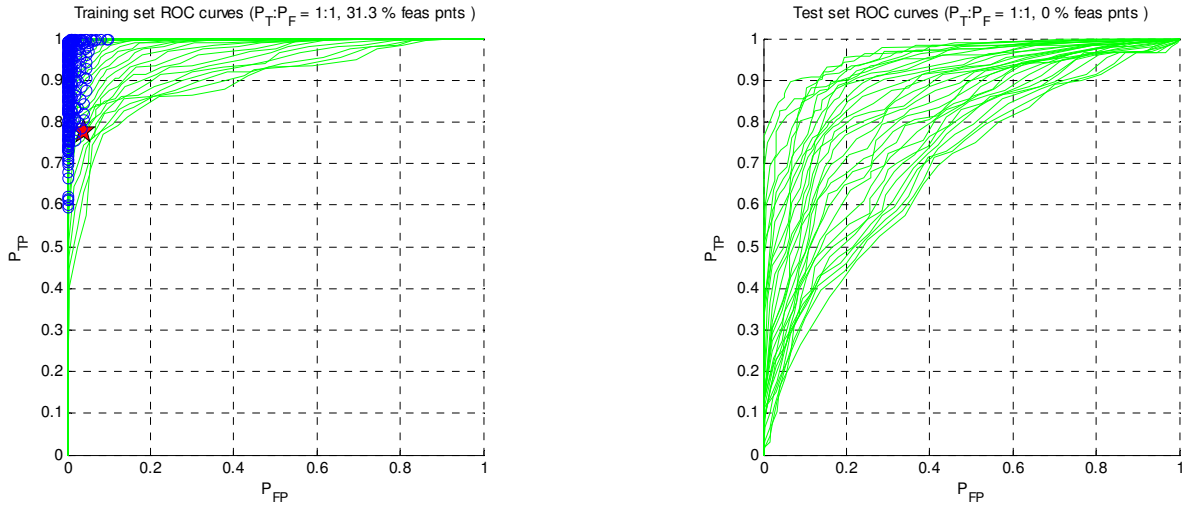
Figure 4.9 An RNN with 8 Input Features Assessed to Generate One ROC Curve for 30 Uniform ROC Thresholds for Each of 30 Uniform Rejection Thresholds

4.4.2 RNN Fusion Experiment Results

The benefit of additional looks is illustrated by comparing the single lowest ROC curve associated with use of only 1 observation and no rejection option, with improvement observed after allowing rejection. In general, the curves generated with larger rejection regions converge toward the upper left-hand plot area, indicative of

improved ROC performance. To determine feasible and optimal thresholds, $TPR(\theta)$ was

maximized with the constraints from eq. 4-1 set at $\Pi_1 = 0.05$, $\Pi_2 = 0.20$ & $\Pi_3 = 0.70$.

Plotting $P_{TP}(\theta)$, $P_{FP}(\theta)$ and $P_{Rej}(\theta)$ leads to the 3D ROC surfaces in Figure 4-10 for

Training and Test sets of the RNN using all 8 input features.
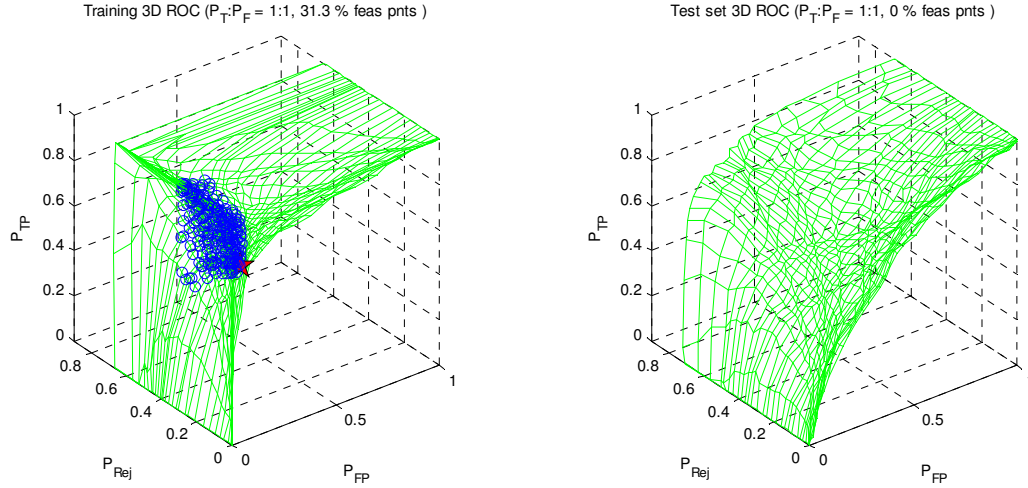
Figure 4.10 Thirty ROC Curves Connected across $\hat{P}_{Rej}$ Values. With Equal Priors,

the Feasible Points Appear Concentrated around a "knee" in the Training Set ROC

Surface, with Feasible Points Located on the Vertical Surface below the "knee"

Feasible points, meeting all decision maker constraints, are identified by the dark

circles in both Figures 4.10 and 4.11. The associated optimal thresholds and performance

parameters are identified in Table 4.7, along with performance and threshold values for

other data sets with various ratios of Targets to Friends. Excursions in P_T:P_F were

performed to obtain additional feasible points. In general, at the optimal thresholds E_{CR}

was found to be a binding constraint across low P_T:P_F priors and either set of input

features. Therefore, increasing target density provided a means to obtain feasible points

and compare classifiers. A pattern of decreased feasibility across thresholds from

Training, to Test, to Validation data sets highlighted similar RNN behavior observed with a forced 2-class decision using the same data with a winner-take-all decision rule (Laine and Bauer, 2003).
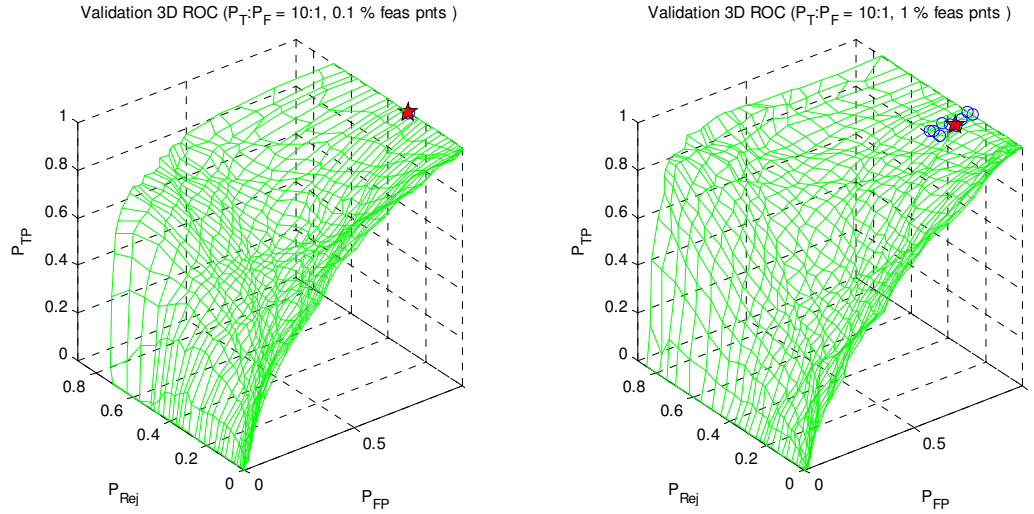
Figure 4.11 ROC Surfaces Generated for Validation Data using All 8 Features on the Left and 3 Features on the Right for P_T:P_F = 10:1. The max $TPR(\theta)$ is Shown by a "star." Aggressive Feasible ROC Thresholds Classify Most Objects as "Targets"

In addition to generating the 3D ROC surfaces in Figure 4.10, similar plots were examined for the RNN using 3 input features with various priors. At both P_T:P_F = 1:1 & 4:1 about 4% of the Training data thresholds resulted in feasible points as compared to about 20-30% of the 8 feature model. Evaluation of the Test data yielded no feasible points. A new prior ratio of P_T:P_F = 10:1 was assessed for both the complete and reduced feature Test data sets and is included in Table 4.7. Analysis of the independent Validation set showed significant performance degradation in both RNN fusion models. With P_T:P_F = 4:1, no feasible operating points were obtained for either feature set,

157

indicative of poor model generalization outside of the Training and Test data sets.

Validation data was then optimized in target rich environments with $P_T:P_F = 10:1$, as seen

in Figure 4.11. A final ratio of $P_T:P_F = 20:1$ identified optimal thresholds that

aggressively labeled almost all declared objects as hostile "Targets," as may be seen in

the bottom rows of Table 4.7.

Table 4.7 Optimal Thresholds for the Maximum *TPR* and Associated Performance
Values for Training (TR), Test (TE) and Validation (VA) Data

feats	data	$P_T:P_F$	P_{TP}	P_{FP}	E_{CR}	E_{NC}	P_{Dec}	θ_l	θ_u	ID/time	max TPR	% Feas
8	TR	1:1	78%	4%	4.9%	19%	100%	0.52	0.70	0.79	0.61	29.30%
3	TR	1:1	78%	4%	4.9%	18%	92%	0.35	0.83	0.27	0.21	4.80%
8	TE	1:1	----	----	----	----	----	----	----	----	----	0%
3	TE	1:1	----	----	----	----	----	----	----	----	----	0%
8	TR	4:1	96%	18%	4.5%	16%	100%	0.21	0.51	0.54	0.51	20.30%
3	TR	4:1	97%	27%	4.8%	16%	84%	0.17	0.59	0.25	0.25	3.70%
8	TE	4:1	97%	31%	4.8%	20%	75%	0.07	0.73	0.20	0.19	0.50%
3	TE	4:1	----	----	----	----	----	----	----	----	----	0%
8	TE	10:1	100%	59%	4.1%	15%	83%	0.05	0.56	0.25	0.25	1.40%
3	TE	10:1	100%	100%	4.9%	0%	75%	0.00	0.51	0.24	0.24	0.30%
8	VA	10:1	100%	100%	4.7%	0%	71%	0.00	0.54	0.22	0.22	0.10%
3	VA	10:1	100%	91%	5.0%	0%	75%	0.03	0.54	0.23	0.23	0.90%
8	VA	20:1	100%	100%	4.8%	0%	100%	0.00	0.00	1.00	1.00	2.00%
3	VA	20:1	100%	97%	4.6%	0%	100%	0.07	0.07	1.00	1.00	6.90%

Noticeable differences in Training and Test sets indicate all 8 features may be

preferred based on the max $TPR(\mathbf{\theta})$ or if evaluating robustness by the percentage of

feasible operating thresholds. Yet, both feature sets failed to generalize well to an

external Validation set with no feasible thresholds when $P_T:P_F = 4:1$. Further

performance evaluation of the Validation data for $P_T:P_F = 10:1$ and $P_T:P_F = 20:1$ resulted

in minimal differences between the two input feature sets. These results are comparable

with those obtained by Laine and Bauer (2003), where forced declaration was analyzed

after each look and no statistical difference could be declared for any data set. Similarly,

steady degradation was observed between Training, Test, and Validation sets, and the

apparent differences between the complete and reduced feature set diminished. Thus, for

classification of data independent of RNN training, use of the reduced feature set appears reasonable. In addition, a reduced feature set may be more efficient for ATR in terms of requiring less data to be collected and processed in near real-time. Overall, additional analysis should be performed for the optimization framework to help identify a preferred model when neither yields feasible thresholds compliant with the initial levied constraints across desired priors.

4.4.3 RNN Experiment Conclusion

In this RNN fusion experiment, a mathematical programming framework was used to optimize rejection and ROC thresholds to maximize a preferred objective while constrained by the operational warfighter constraints. To visualize some key performance relations a 3-D ROC surface was presented. An objective function to maximize $TPR(\theta)$ was selected. One advantage of the optimization framework is development of acceptable constraints vs. quantifying difficult misclassification costs, leading to feasible regions across the multiple projected ROC curves in addition to optimal points. The percentage of feasible operating thresholds may help measure a system's robustness and only gives credit to acceptable portions of a ROC curve. This may be useful for comparison of systems desired to perform across untested extended operating conditions (EOC) with potential deviations from the training data (Ross *et al.*, 1997).

4.5 Initial Findings & Contributions for Two-Class Generated Data

In general, these generated 2-class experiments show the optimization framework from Chapter 3 has the potential to be a helpful diagnostic tool in ATR when P_{TP} and P_{FP} are not sufficient to compare competing classifiers. Such is the case for USAF applications where a minimum level of confidence, as reflected by the operational constraints, is required before making an actionable decision to engage enemy targets. The optimization of thresholds may be performed based on a preferred objective function subject to other constraints. Visualization of the ROC surface, generated from the same thresholds, may aid in a better understanding of the tradeoffs between true positives, false positives, and declarations; along with providing an image representative of traditional ROC performance variables. Further, the values of P_{TP} and P_{FP} which satisfy the decision maker's constraints are highlighted on traditional ROC curves and show feasible operating regions. To gain more insight of classification systems, sensitivity analysis of the constraints and the operating environment identified by the ratio of P_T:P_F or through use of EOC test data may be performed. Subsequent adaptation of ATR systems across operational settings, possibly through the tuning of the rejection and ROC thresholds, may contribute toward ATR system utility in which systems may adapt to the operating environment. This type of adaptation is currently being supported by research performed at AFRL/SN as documented by Wise *et al.* (2004).

The mathematical programming framework was illustrated with simple ATR examples using simulated data. While limited examples with up to 10-looks of generated data were presented, some general observations were made. For all experiments, an

increase in the ratio of Targets to Friends facilitated the feasible use of aggressive thresholds. Under these conditions many of the Friendly targets were classified as "no-declaration," to obtain a desired Critical error performance. This was accomplished by allowing up to 30% of all targets to be undeclared, with a desired Declaration constraint of 70% or greater. This mathematically supports "Blue-Force tracking" and cooperative ID. These two systems may effectively change the prior ratio of "unidentified" potential targets, since those targets being assessed by an ATR system typically do not provide a response to electronic interrogation as used in cooperative ID, and are not yet tracked as a positive friendly force.

Overall, this mathematical optimization may be a significant aid for the evaluation and comparison of competing ATR systems, which are required to fuse data to reach desired levels of correct class declarations. The proposed methodology goes beyond the traditional ATR system evaluation methods and determines the preferred ATR operating thresholds and other system parameters without use of explicit costs and may optimize *TPR,* a proposed measure of performance, to account for the time involved to collect and analyze sensor data. This measure can then be used to help determine the relative value of obtaining correlated data quickly or of obtaining less correlated data across a longer time period. In summary, the optimization methodology incorporates a flexible framework to establish a decision maker's primary objective, subject to constraints, and does so across both the warfighter's "vertical" view of declared targets and the engineer's "horizontal" view of actual types of objects classified.

V. MVP Optimization Application to DCS Radar Data

The goal of this chapter is present an illustrative example of the utility of the mixed variable optimization formulation to assess and compare different fusion systems. The specific task at hand is to determine which fusion system would be preferred by the warfighter, for a given a specific collection of radar data. This chapter is organized in the following manner. Section 5.1 presents an overview of the fusion experiment and introduction to the DCS radar data collection used by the fusion systems within this chapter. Section 5.2 gives the specific details of the optimization formulation as applied to the for DCS data set. Specific information on the generation of sensor level data features from the collected DCS radar imagery is then provided in Section 5.3. The two competing fusion methodologies are then described in Section 5.4 for the Majority Vote Boolean (MVB) Fusion Method and in Section 5.5 for the Probabilistic Neural Network (PNN) Fusion Method. Section 5.6 provides an initial comparison of fusion systems, followed by sensitivity analysis in Section 5.7. Section 5.8 then introduces a temporal comparison across correlation levels for a limited number of cases. Finally, potential future experimental excursions are briefly identified in Section 5.9, and a summary of findings is included as Section 5.10.

5.1 Overview of DCS Radar Data Fusion Experiment

The primary objective of this chapter is to perform an experiment using collected radar imagery data to demonstrate the potential utility of the MVP optimization framework. This optimization framework will facilitate gaining insight of fusion

preferences for an ATR system with two input sensors used to make unforced decisions through time. In general, warfighter perspectives are incorporated by maximizing the objective function, *TPR*(*x*), subject to meeting the desired Critical Error, Non-critical Error and Declaration constraints. The decision variables identified by *x* include the categorical fusion rules under investigation and the continuously valued threshold variables, θ. For this experiment, the ATR system is designed to provide the warfighter one of four output labels. The desired output labels include "Target of the Day," "Other Hostile," "Friend/Neutral" and "No-declaration." Figure 5.1 provides a general description of the task-at-hand, where two sensors will be used to make the label assessments, and additional data may be collected through time.
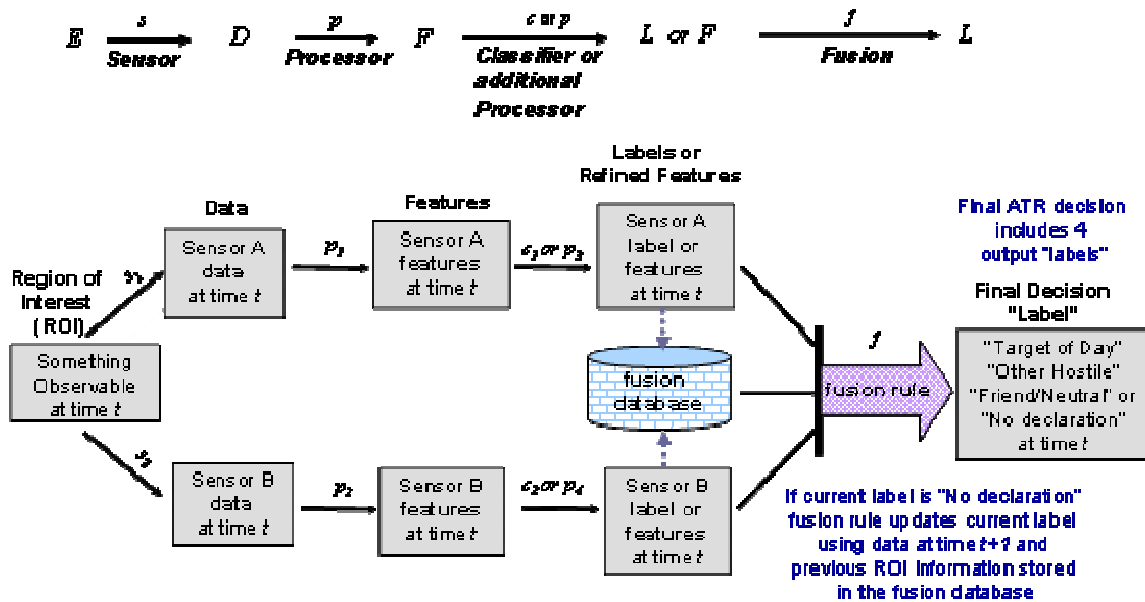
Figure 5.1 Overview of the ATR Process with Four Desired ATR Output Labels

The DCS radar data collection includes 2–dimensional X-band radar imagery collected on 15 different ground targets located in the same general area. The imagery was collected using both HH and VV radar polarizations, which will be used to represent

163

sensors A and B. The targets are all different ground vehicles with representation of potential Friend, Enemy and Neutral targets. Table 5.1 provides a description of the 15 different vehicles. A subset of five Hostile enemy vehicles and five Friend/Neutral vehicles was selected to generate a balanced data set for this experiment. The selected Hostile targets include the SCUD, SMERCH, SA-6 Radar, SA-6 TEL and T-72, as indicated by first five rows of Table 5.1. The next five rows with a grey background show the selected Friend/Neutral targets include the Zil-131 (medium Budget truck), HMMWV, M113, Zil-131 (small Budget truck) and M-35 (large Budget truck). In addition, the SCUD will be designated the desired Target of the Day (TOD). The SMERCH is the same relative size as a SCUD; yet, differs in that it has a Multiple Launch Rocket System (MLRS) as opposed to a single large missile on the SCUD. Since the SMERCH is built on the same chassis as the SCUD, it is a potential TOD confuser. Five unused targets are shown in the last five rows of the table.

Table 5.1 Description of 15 Targets Imaged by DCS Radar

DCS Radar Collection

Location	Type	Target Description	tracks	wheels	gun
1	SCUD	Single Large Missile	N	8	N
2	SMERCH	MLRS & Scud Confuser	N	8	N
5	SA-6 Radar	Similar to SA-6 TEL	Y	0	N
10	T-72	Main Battle Tank	Y	0	Y
13	SA-6 TEL	3 Medium SAMs	Y	0	N
6	*Zil-131*	*Medium Budget Truck*	*N*	*4*	*N*
7	*HMMWV*	*Jeep like SUV*	*N*	*4*	*N*
11	*M113*	*Armored Personnel Carrier*	*Y*	*0*	*Y*
12	*Zil-131*	*Small Budget Truck*	*N*	*4*	*N*
15	*M35*	*Large Budget Truck*	*N*	*4*	*N*
3	SA-8 TZM	SA-8 Reload vehicle	N	6	N
4	BMP-1	tank w/small turret	Y	0	Y
8	BTR-70	8-wheeled transport	N	8	N
9	SA-13	turret SAMs	Y	0	N
14	SA-8 TEL	integrated radar & exposed SAMS	N	6	N

Additional information on the targets is provided in Section 5.3 as sensor features are developed. The basic experiment will seek to determine which of two fusion methods is preferred. The two competing fusion schemes include a Majority Vote Boolean (MVB) fusion method and use of a Probabilistic Neural Network (PNN) for fusion. These two methods provide for different levels of fusion to occur. The MVB Boolean fusion combines labels generated by each sensor, while the PNN neural fusion combines continuous valued probability estimates from sensors A and B associated with the three desired labels. The two sensors being fused include the DCS radar collected with HH-polarization and processed by an HRR algorithm developed by Çetin (2001) and the VV-polarized data processed with an HRR algorithm described within (Williams *et al.,* 1999) and obtained from the authors associated with AFRL/SN. A set of training data will be used to estimate aspect angle templates to provide initial sensor estimates of the posterior probabilities associated with each target type and to train the PNNs used for fusion.

5.2 MVP Formulation for DCS Data Experiment

As developed in Chapter 3, optimization of a mixed variable mathematical programming formulation will be the used to determine the preferred fusion method. The objective function will seek to maximize *TPR* subject to constraints. The applicable constraints include the warfighter operational constraints, the fusion rule constraint, minimum look constraint, and threshold constraints. Decision variables are identified by *x* and include categorical fusion rule and minimum look variables, and the variable thresholds, $\boldsymbol{\theta}$. The formulation is as follows:

Objective function:

$$\max_{x \in X} T\hat{P}R(x) = \frac{\hat{P}_{TP}(x)}{\hat{E}(time_{TP}(x))} \quad \text{maximize the rate of hostile target detection} \qquad (5\text{-}1)$$

Subject to:

Initial warfighter operational constraints:

$$\hat{E}_{CR}(x) < \Pi_1 = 0.02 \quad \text{limit incorrect fire decisions (vertical analysis)}$$

$$\hat{E}_{NC}(x) < \Pi_2 = 0.05 \quad \text{limit lower impact incorrect decisions (vertical analysis)}$$

$$\hat{P}_{Dec}(x) > \Pi_3 = 0.70 \quad \text{limit Non-declarations (horizontal analysis)}$$

Fusion Rule constraint:

$$\sum_{i=1}^{2} F_i = 1 \qquad \text{indicate MVB or PNN fusion}$$

$$\text{where } F_i = \begin{cases} 1 & \text{if } i \text{ th MVB or PNN fusion used} \\ 0 & \text{otherwise} \end{cases}$$

Sensor Selection constraints:

For this experiment, fusion of both Sensors A and B will be used at each time t

Minimum Look constraint:

$ML \geq \min Looks = \{1,2,3,4,5\}$ to require a minimum number of looks prior

to making a final label declaration

Threshold constraints:

$\boldsymbol{\theta}^{MVB} = ((\boldsymbol{\theta}^{SA})^T, (\boldsymbol{\theta}^{SB})^T)^T$, where SA and SB refer to Sensors A and B,

$\boldsymbol{\theta}^{SA} = (\theta^{SA}{}_{low}, \theta^{SA}{}_{up})^T$ and $\boldsymbol{\theta}^{SB} = (\theta^{SB}{}_{low}, \theta^{SB}{}_{up})^T$ for MVB fusion, and

$\boldsymbol{\theta}^{PNN} = (\theta^{PNN}{}_{low}, \theta^{PNN}{}_{up})^T$ for PNN fusion, and

$\Theta = \{\mathbf{\theta}^{ij} : 0 \le \theta^{ij}_{\ low} \le \theta^{ij}_{\ up} \le 1\}$ and $ij \in \{SA, SB, PNN\}$ is the set of all feasible

thresholds used by PNN fusion or used by a single sensor for MVB fusion.

The Critical error and Non-critical error warfighter operational constraints are

computed by vertical analysis of a confusion matrix consisting of the true classes and the

classifier output labels. Figure 6.2 shows the associated confusion matrix for this

experiment. The true classes include the Hostile Target of the Day (TOD), Other

Hostiles (OH), and a consolidated Friend/Neutral class (FN). The classifier labels

include "TOD," "OH," "FN" and "ND," for the rejection option of "Non-declaration."

As indicated by the legend, Critical errors occur for an incorrect Hostile (TOD or OH) vs.

"Friend/Neutral" label or vise versa, while Non-critical errors occur only within the two

Hostile classes by incorrectly labeling a TOD as "OH" or OH as "TOD".

		Classifier "Labels"			
True Classes	"TOD"	"Other Hostile"	"Friend / Neutral"	"No declaration"	Horizontal Totals
TOD	TOD labeled "TOD"	TOD labeled incorrect "Hostile"	TOD labeled "FN"	TOD labeled "Unknown"	TOD evaluated
Other Hostile	Hostile labeled "TOD"	Hostile labeled "Hostile"	Hostile labeled "FN"	Hostile labeled "Unknown"	Other Hostile evaluated
Friend/ Neutral	F or N labeled "TOD"	F or N labeled "Hostile"	F or N labeled "FN"	F or N labeled "Unknown"	F or N evaluated
Vertical Totals	"TOD" declared	"Other Hostile" declared	"F or N" declared	"Unknown" declared	

Legend	Assessment	Analysis
	Correct ID	Horizontal
	Critical Error	Vertical
	Non-Critical Error	Vertical
	Non-Declaration	Horizontal
	Totals	H or V Analysis

Figure 5.2 Confusion Matrix Associated with Four Desired Output Labels

Probability estimates can be obtained by horizontal analysis of the confusion matrix, where the probability of a specific label given a true class is calculated using the frequency of occurrence. The probabilities associated with "*label j*" declarations are given as:

$$\hat{P}("label_j" \mid true\ class_i) = \frac{\#("label_j" \& true\ class_i)}{Total\ \#true\ class_i\ evaluated}. \qquad (5\text{-}2)$$

In some situations, like the assessment of true positive declarations, the probabilities associated with the labels excluding the "Non-declarations" may be desired, where:

$$\hat{P}("label_j" \mid true\ class_i\ \&\ declaration) = \frac{\#("label_j" \& true\ class_i)}{Total\ \#true\ class_i\ declared}. \qquad (5\text{-}3)$$

For assessment of the probability of declaration, $\hat{P}_{Dec}$, each label is a disjoint event, with

$$\hat{P}_{Dec} = \hat{P}("TOD" \cup "OH" \cup "FN") = \hat{P}(\overline{"ND"}) = 1 - \hat{P}("ND"). \qquad (5\text{-}4)$$

Because all probabilities and other measures of performance will be estimated using different data sets, the "hat" will not be in the remainder of this chapter, but is implied for estimated values, such that $P = \hat{P}$.

5.2.1 Critical Error Calculation

From Figure 5.2, it can be seen that four possible events may be labeled as critical error. These disjoint events include classification as a "Hostile" ("TOD" or "OH") given a true FN or classification as "FN" given one of the two true Hostile classes, TOD or Other Hostile (OH). The probability of Critical Error is defined as probability associated with the union of the four output label and true class intersections, given a declaration is made, as shown in eq. 5-5.

$$P\left(E_{CR}\right) = P\left(\left(\begin{array}{c} P("TOD" \cap FN) \cup P("OH" \cap FN) \\ \cup P("FN" \cap TOD) \cup P("FN" \cap OH) \end{array}\right) \middle| declaration\right) \qquad (5\text{-}5)$$

If the probabilities associated with each confusion matrix element are estimated, and the

prior probabilities of each of the true classes are known, then vertical analysis of the

appropriate confusion matrix elements may be performed to calculate the Critical Error.

First, let $P(TOD)$, $P(OH)$ and $P(FN)$ be the prior probabilities associated with each true

class, where $P(TOD) + P(OH) + P(FN) = 1$. Similarly, $P("TOD")$, $P("OH")$,

$P("FN")$ and $P("ND")$ are the unconditional probabilities of the ATR system output

labels and sum to 1. Because the four class/label combinations are disjoint events, eq. 5-

5 may be rewritten as,

$$\begin{aligned} P\left(E_{CR}\right) = {} & P(FN \cap "TOD" \mid declaration) + P(FN \cap "OH" \mid declaration) \\ & + P(OH \cap "FN" \mid declaration) + P(TOD \cap "FN" \mid declaration) \end{aligned} \qquad (5\text{-}6)$$

Using Bayes rule then provides the following equation,

$$\begin{aligned} P\left(E_{CR}\right) = {} & P(FN \mid "TOD" \cap declaration) P("TOD" \mid declaration) \\ & + P(FN \mid "OH" \cap declaration) P("OH" \mid declaration) \\ & + P(OH \mid "FN" \cap declaration) P("FN" \mid declaration) \\ & + P(TOD \mid "FN" \cap declaration) P("FN" \mid declaration) \end{aligned} \qquad (5\text{-}7)$$

Eq 5-7 then reduces to,

$$\begin{aligned} P\left(E_{CR}\right) = {} & P(FN \mid "TOD") P("TOD" \mid declaration) \\ & + P(FN \mid "OH") P("OH" \mid declaration) \\ & + P(OH \mid "FN") P("FN" \mid declaration) \\ & + P(TOD \mid "FN") P("FN" \mid declaration) \end{aligned} \qquad (5\text{-}8)$$

The appropriate conditional probabilities of each "*label i*" are given as follows, where the

equalities from eq. 5-4 may be used to obtain the following relationship,

$$P("label_i"\,|\,declaration) = \frac{P("label_i"\cap declaration)}{P(declaration)} = \frac{P("label_i")}{1-P("ND")}, \quad (5\text{-}9)$$

where, $P("TOD"\,|\,dec) + P("OH"\,|\,dec) + P("FN"\,|\,dec) = 1$, with *dec* shorthand for

declaration. Substituting eq. 5-9 into eq. 5-8 yields:

$$P(E_{CR}) = \frac{\begin{pmatrix} P(FN\,|\,"TOD")P("TOD") + P(FN\,|\,"OH")\,P("OH") \\ + P(OH\,|\,"FN")P("FN") + P(TOD\,|\,"FN")P("FN") \end{pmatrix}}{1-P("ND")}. \quad (5\text{-}10)$$

Using Bayes Rule for each of the conditional probabilities of eq. 5-10 produces the

following relations:

$$P(FN\,|\,"TOD") = \frac{P("TOD"\,|\,FN)P(FN)}{P("TOD"\,|\,TOD)P(TOD) + P("TOD"\,|\,OH)P(OH) + P("TOD"\,|\,FN)P(FN)}$$

$$P(FN\,|\,"OH") = \frac{P("OH"\,|\,FN)P(FN)}{P("OH"\,|\,TOD)P(TOD) + P("OH"\,|\,OH)P(OH) + P("OH"\,|\,FN)P(FN)} \quad (5\text{-}11)$$

$$P(OH\,|\,"FN") = \frac{P("FN"\,|\,OH)P(OH)}{P("FN"\,|\,TOD)P(TOD) + P("FN"\,|\,OH)P(OH) + P("FN"\,|\,FN)P(FN)}$$

$$P(TOD\,|\,"FN") = \frac{P("FN"\,|\,TOD)P(TOD)}{P("FN"\,|\,TOD)P(TOD) + P("FN"\,|\,OH)P(OH) + P("FN"\,|\,FN)P(FN)}$$

Using the Law of Total Probability, unconditional probabilities for each "label" can be

calculated as,

$$\begin{aligned} P("TOD") &= P("TOD"\,|\,TOD)P(TOD) + P("TOD"\,|\,OH)P(OH) + P("TOD"\,|\,FN)P(FN) \\ P("OH") &= P("OH"\,|\,TOD)P(TOD) + P("OH"\,|\,OH)P(OH) + P("OH"\,|\,FN)P(FN) \\ P("FN") &= P("FN"\,|\,TOD)P(TOD) + P("FN"\,|\,OH)P(OH) + P("FN"\,|\,FN)P(FN) \\ P("ND") &= P("ND"\,|\,TOD)P(TOD) + P("ND"\,|\,OH)P(OH) + P("ND"\,|\,FN)P(FN) \end{aligned} \quad (5\text{-}12)$$

Substituting equalities from eq. 5-11 and 5-12 into eq. 5-13 shows $P(E_{CR})$ may be

calculated as:

$$P(E_{CR}) = \frac{\begin{pmatrix} P("TOD"\,|\,FN)P(FN) + P("OH"\,|\,FN)P(FN) \\ + P("FN"\,|\,OH)P(OH) + P("FN"\,|\,TOD)P(TOD) \end{pmatrix}}{1-P("ND")}, \quad (5\text{-}13)$$

$$P(E_{CR}) = \frac{\left(\begin{array}{c} P("TOD"|FN)P(FN) + P("OH"|FN)P(FN) \\ + P("FN"|OH)P(OH) + P("FN"|TOD)P(TOD) \end{array} \right)}{1 - \left(P("ND"|TOD)P(TOD) + P("ND"|OH)P(OH) + P("ND"|FN)P(FN) \right)} \quad (5\text{-}14)$$

Thus, E_{CR} may be calculated directly from the estimated probabilities generated by the standard horizontal analysis of the confusion matrix frequency counts, along with the desired prior probability for each class.

5.2.2 Non-critical Error Calculation

The non-critical errors (E_{NC}) may be calculated in a similar manner to E_{CR} and includes declarations leading to non-optimal sortie performance or weapon selection. The E_{NC} events include declarations of true Hostile targets incorrectly as the "TOD" or "OH." The two desired Hostile target labels include:

1. *"TOD" = Target of the Day (SCUD)*

2. *"OH" = Other Hostiles (SMERCH, SA-6 Radar, SA-6 TEL, T-72)*

Non-critical errors occur for incorrect Hostile label declarations, such as labeling a SCUD as "OH" or a SMERCH as "TOD" as shown by the following definition:

$$P(E_{NC}) = P\left(\left(P("TOD" \cap OH) \cup P("OH" \cap TOD) \right) | declaration \right). \quad (5\text{-}15)$$

Thus, using the Law of Total Probability and proceeding as was done for eq. 5-6, and 5-7,

$$P(E_{NC}) = P(OH|"TOD")P("TOD"| declaration) \\ + P(TOD|"OH")P("OH"| declaration) \quad (5\text{-}16)$$

Using eq. 5-9 for the calculation of $P("label_i"| declaration)$,

$$P(E_{NC}) = \frac{P(OH|"TOD")P("TOD") + P(TOD|"OH")P("OH")}{1 - P("ND")}. \quad (5\text{-}17)$$

171

Substituting the relationships from eq. 5-12 and use of equations similar to 5-11 shows $P(E_{NC})$ may be calculated as:

$$P(E_{NC}) = \frac{P("OH"|TOD)P(TOD) + P("TOD"|OH)P(OH)}{1 - \left(P("ND"|TOD)P(TOD) + P("ND"|OH)P(OH) + P("ND"|FN)P(FN)\right)}. \quad (5-18)$$

Therefore, an estimate of the non-critical error may be obtained from the probabilities calculated from an initial horizontal analysis of an ATR system's confusion matrix for any desired prior probabilities.

5.2.3 "Non-declaration" Calculation

The estimated percentage of final fusion system rejections or Non-declarations (P_{ND}) provides a measure of the total objects being assessed that are left labeled as "Non-declaration" or "ND" at the end of all potential sensing opportunities. For the fusion experiments in this chapter, a final label of "ND" only occurs after attempting to classify the target vehicle using sensor looks during all five time periods available. Since the "Non-declaration" measure is calculated using horizontal analysis of the confusion matrix elements, an estimate for P_{ND} may be calculated using of the Law of Total Probability directly from eq. 5-4 as,

$$P_{ND} = P("ND"|TOD)P(TOD) + P("ND"|OH)P(OH) + P("ND"|FN)P(FN). \quad (5-19)$$

With the Probability of Declaration, $P_{Dec} = 1 - P_{ND}$, and $P_{ND} = P_{REJ}$.

5.3 Sensor Level Features Derived from the DCS Radar Data

The DCS radar data were collected May 2004 at Eglin AFB and was obtained through a data request submitted to the Sensor Data Management System (SDMS)

website (https://www.sdms.afrl.af.mil/request/data_request.php) by an on-line data request. Data were collected by a General Dynamics DCS X-band synthetic aperture (SAR) radar operating in spotlight mode. A medium sized Convair 580 with twin engines and turbo-propellers was used as a host platform for the DCS radar system. The DCS radar sensor bandwidth was 640 MHz with a peak transmit power of 4 kW. The HH and VV polarized DCS radar 2-D imagery was collected with a resolution of 1.0 ft, for both magnitude and phase information. Spotlight scenes with all 15 targets were collected. All targets were imaged in an open area without concealment, and all vehicles were aligned in similar headings, but remained stationary for the data collection. From these full spotlight scenes, individual target region of interest (ROI) chips were extracted. The individual target chips are 256 x 256 pixels and centered on each target. Separation between targets provided individual chips to only contain radar returns associated with the individual target of interest or the benign background. All data used in this experiment was processed using the individual target ROIs.

5.3.1 Selection of Training and Test Data

The DCS radar data collection contained spot radar images collected across 360 degrees of aspect angle for depression angles of 6, 8 and 10 degrees with multiple flight passes collecting data through approximately 90 degrees. Some additional flight passes collected data at varying depression angles of 4-11 degrees and at a depression angle of 12 degrees. The data collected at depression angles of 6 and 8 degrees was selected to be used as a Training data set. Data collected at a depression angle of 10 degrees, with similar flight passes as those selected for the Training data, was selected to form a Test

data set to represent an extended operating condition (EOC), outside the range of depression angles used for any training purposes.

The Training data were collected at an elevation of approximately 3000 and 4000 feet, while the EOC Test data were collected at an elevation of approximately 5000 feet. The Training data included 724 observations of each target type for both HH and VV polarizations. For each flight pass of the aircraft, approximately 4 degrees of aspect angle separates consecutively collected radar images. A total of 32 flight passes with 22 or 23 looks per flight is included in the Training data set, where each flight pass covers approximately 90 degrees. The Test data includes 446 observations of each target type by each polarization, with a desired collection at a depression angle of 10 degrees. A total of 20 flight passes, with 22 or 23 looks per flight, is used to generate all Test sets and provides for testing across the full aspect range of 360 degrees. The specific flight passes used to generate the Training and Test data sets are included in Table 5.2 and Table 5.3.

Table 5.2 Data Selected for Training with a Desired Depression Angle of 6 or 8 Degrees

Number	Flight	Pass	Identifier	# chips	looks per vehicle	Desired Dep
1	1	10	FP0110	690	46	6
2	1	11	FP0111	660	44	6
3	1	12	FP0112	660	44	6
4	1	13	FP0113	660	44	6
5	1	15	FP0115	690	46	8
6	1	16	FP0116	690	46	8
7	1	17	FP0117	690	46	8
8	1	18	FP0118	690	46	8
9	1	34	FP0134	690	46	8
10	2	12	FP0212	660	44	6
11	2	13	FP0213	660	44	6
12	2	14	FP0214	690	46	6
13	2	16	FP0216	690	46	8
14	2	17	FP0217	690	46	8
15	2	18	FP0218	690	46	8
16	2	19	FP0219	690	46	8
17	2	32	FP0232	660	44	6
18	2	33	FP0233	660	44	6
19	2	34	FP0234	690	46	6
20	2	35	FP0235	660	44	6
21	2	36	FP0236	660	44	6
22	2	37	FP0237	660	44	6
23	2	38	FP0238	690	46	6
24	2	39	FP0239	660	44	6
25	3	6	FP0306	660	44	6
26	3	7	FP0307	690	46	6
27	3	8	FP0308	690	46	6
28	3	9	FP0309	690	46	6
29	3	11	FP0311	690	46	8
30	3	12	FP0312	690	46	8
31	3	13	FP0313	690	46	8
32	3	14	FP0314	690	46	8

# looks per vehicle	1448
HH looks per vehicle	724
VV looks per vehicle	724
mean aspect sampling	0.50 degrees
Total number of chips processed	21720

Table 5.3 Data Selected for Test with a Desired Depression Angle of 10 Degrees

Number	Flight	Pass	Identifier	# chips	looks per vehicle	Desired Dep
1	1	20	FP0120	660	44	10
2	1	22	FP0122	660	44	10
3	1	23	FP0123	690	46	10
4	1	25	FP0125	690	46	10
5	2	21	FP0221	660	44	10
6	2	23	FP0223	660	44	10
7	2	24	FP0224	660	44	10
8	2	26	FP0226	660	44	10
9	3	16	FP0316	660	44	10
10	3	18	FP0318	660	44	10
11	3	19	FP0319	660	44	10
12	3	21	FP0321	660	44	10
13	3	28	FP0328	690	46	10
14	3	29	FP0329	660	44	10
15	3	31	FP0331	660	44	10
16	3	32	FP0332	660	44	10
17	3	33	FP0333	660	44	10
18	3	34	FP0334	690	46	10
19	3	35	FP0335	690	46	10
20	3	36	FP0336	690	46	10

# looks per vehicle	892
HH looks per vehicle	446
VV looks per vehicle	446
mean aspect sampling	0.81 degrees
Total number of chips processed	13380

Overall, a total of 35,100 complex SAR chips with 256 x 256 pixels were processed.

Each chip is approximately 520 KB. Thus, a subset of the original DCS radar dataset

including over 18 GB of radar data were processed as described in the following section.

5.3.2 Generation of HRR Features

Once a subset of the original DCS radar data collection was identified for use as

Training and Test data, the next step was to process the data into reasonable sensor output

features. High Range Resolution (HRR) profiles offer enhanced target-to-clutter and

noise signatures for moving targets through Doppler filtering and the use of clutter

176

cancellation (Williams *et al.,* 2000). To model the tracking and subsequent identification of moving ground targets, each of the images was processed into a HRR profile using two different algorithms, followed by template matching using two different angular resolutions. In attempt to generate sensors with different characteristics, the two polarizations of radar data were processed using different algorithms for generating the HRR range profiles. Çetin's point based reconstruction (PBR) algorithm (Çetin, 2001) was selected for processing the HH-polarized data and an algorithm developed by AFRL with use by the MSTAR program (Williams *et al.,* 1998, 1999, 2000) was selected to process the VV polarized data. An overview of the steps required to process a single chip using each method is presented in Figure 6.3 on the following page. Specific details for both processing algorithms can be viewed in the *Matlab* code used to process all chips and is included in Appendix B as files DCS_proc1.m, DCS_proc2.m, and DCS_proc3.m. Initial versions of these *Matlab* files were obtained from Albrecht (2004) and were originally used to process MSTAR chips of size 128×128. The Matlab files were modified to be used as function calls to process the complex 256×256 DCS radar chips. DCS_proc1.m generates an 1×322 HRR range profile using the AFRL procedure for each chip, along with an 1×200 input profile for use by Çetin's PBR algorithm. DCS_proc2.m generates normalized 1×200 profiles across all aspect angles and targets, and DCS_proc3.m generates the PBR 1×200 profiles from the normalized profiles.

Figures 5.4-5.13 provide samples of HRR profiles obtained by both the AFRL and PBR HRR algorithms for each of the 10 target vehicles used in this experiment. All ten figures contain samples of both HH and VV polarization for each HRR algorithm and were collected by the DCS radar during one spotlight image of the entire scene.

177
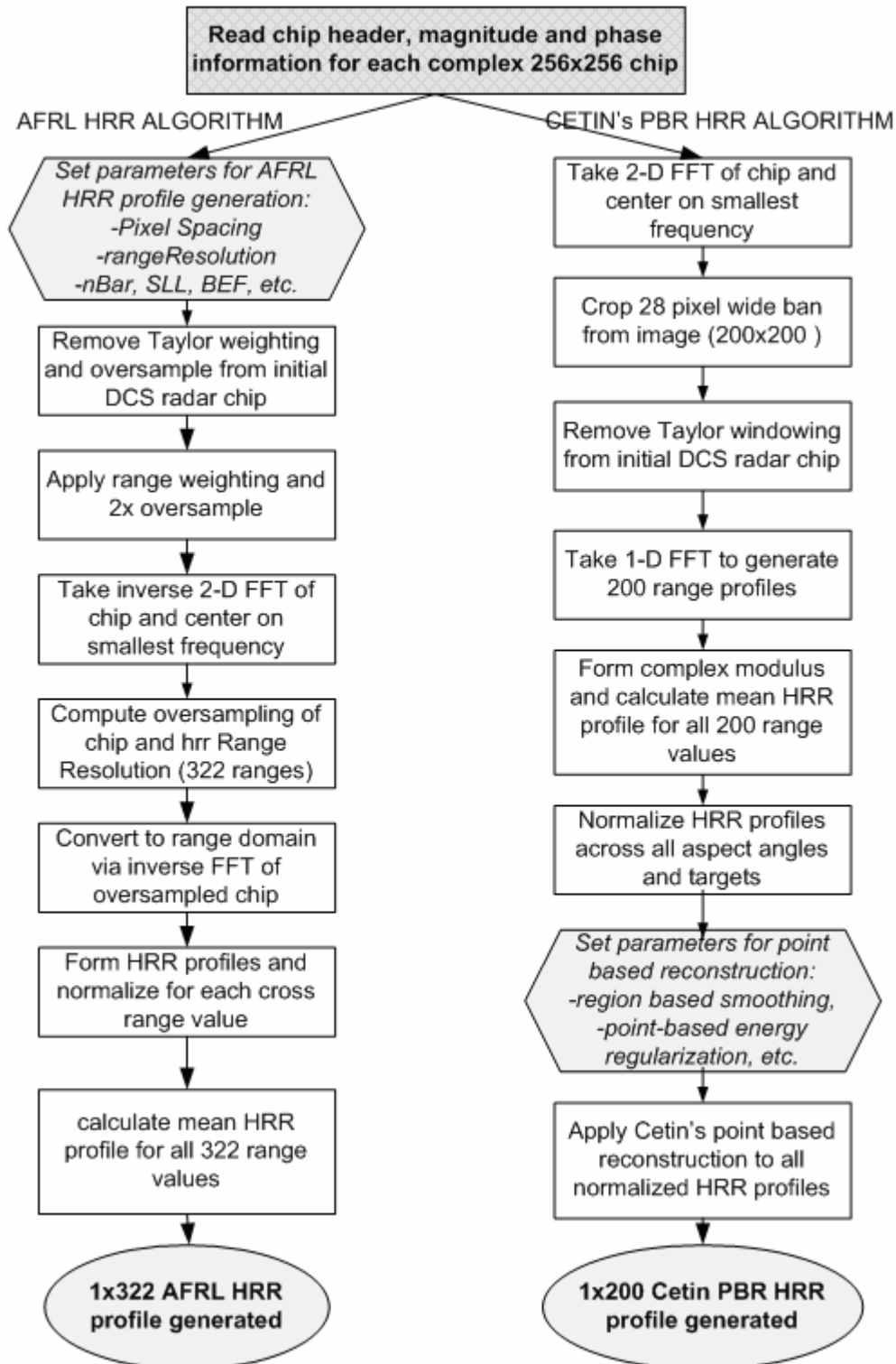
Figure 5.3 DCS Data HRR Processing by AFRL and Çetin's PBR Algorithms
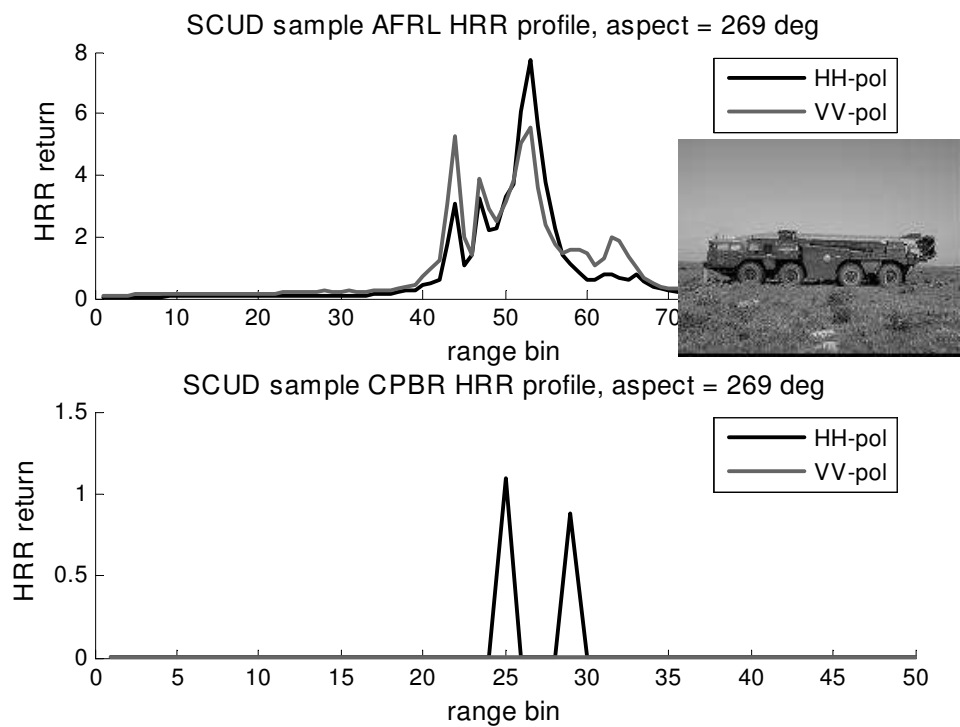
Figure 5.4 Sample SCUD HRR Profile (label: Hostile - TOD)
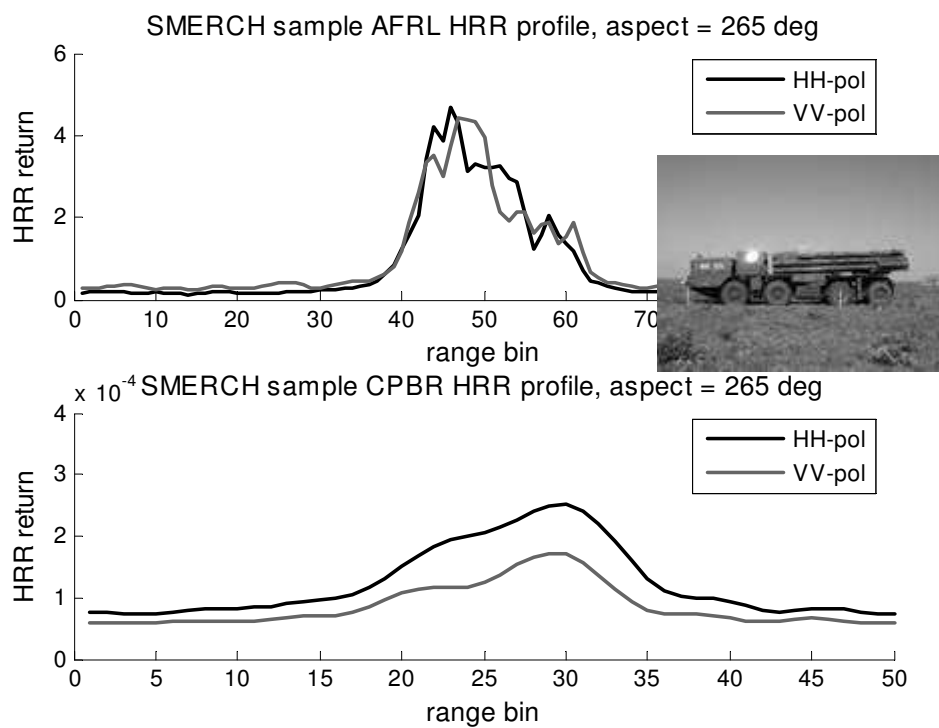
Figure 5.5 Sample SMERCH HRR Profile (label: Other Hostile)
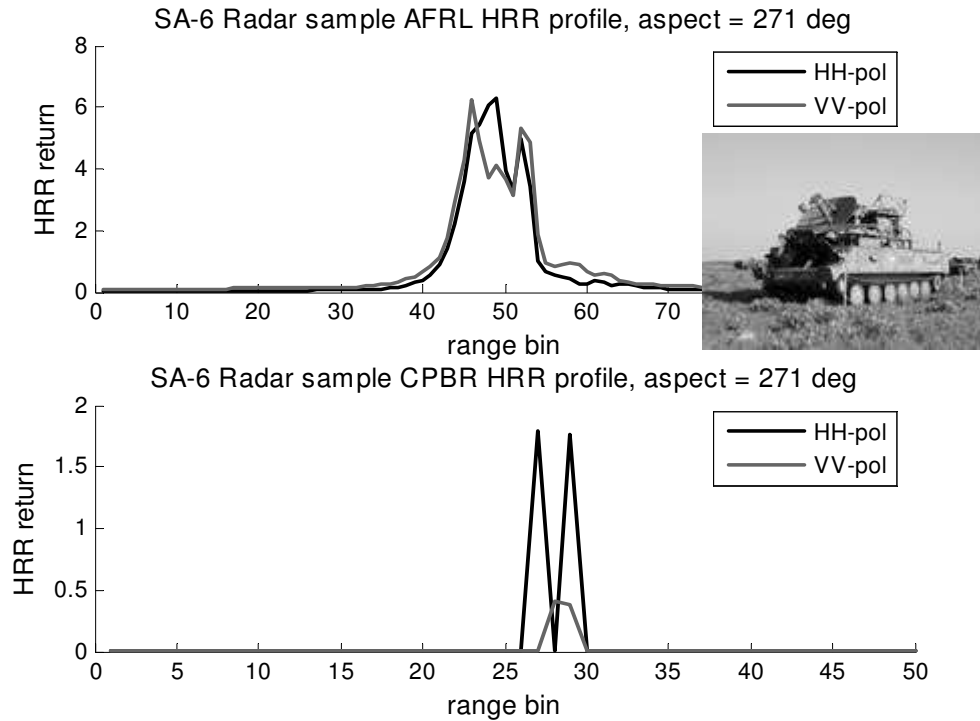
Figure 5.6 Sample SA-6 Radar HRR Profile (label: Other Hostile)
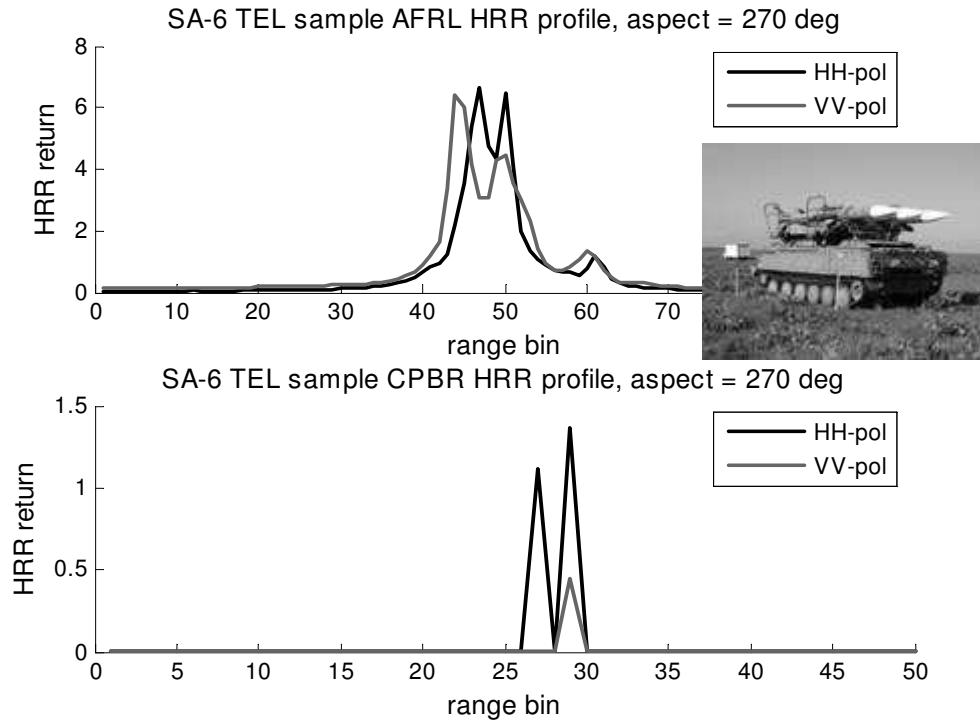
Figure 5.7 Sample SA-6 TEL HRR Profile (label: Other Hostile)
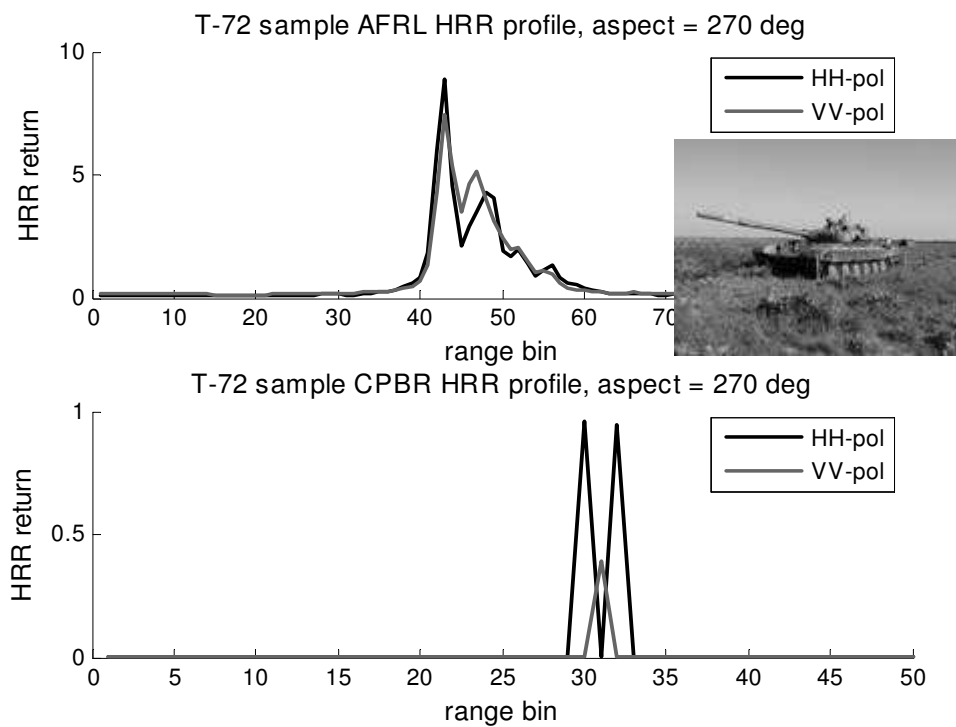
Figure 5.8 Sample T-72 HRR Profile (label: Other Hostile)
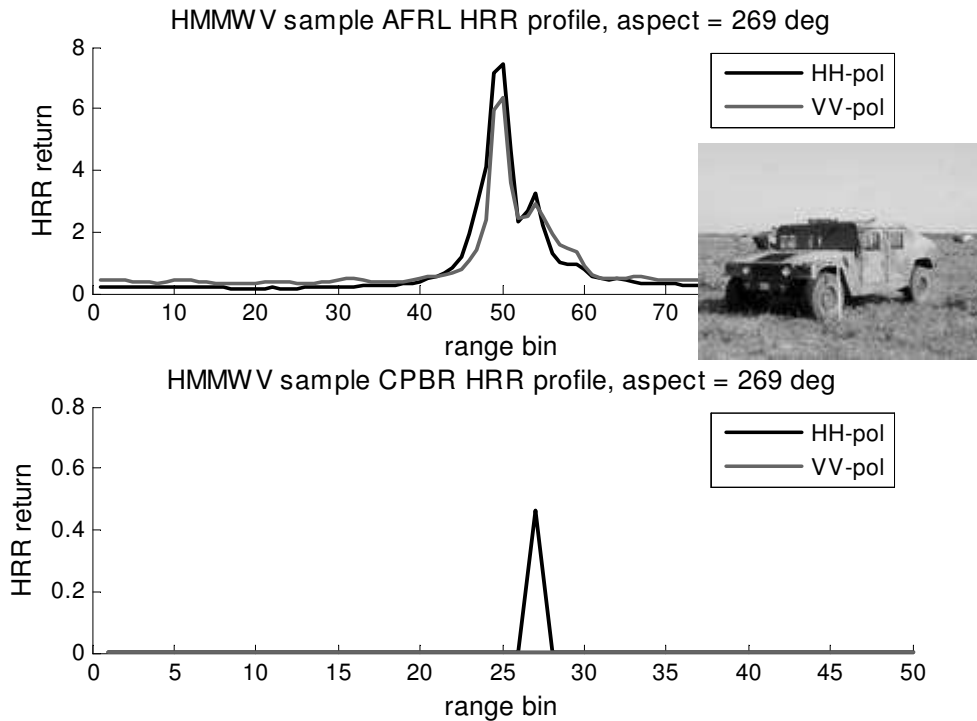
Figure 5.9 Sample HMMWV HRR Profile (label: Friend)
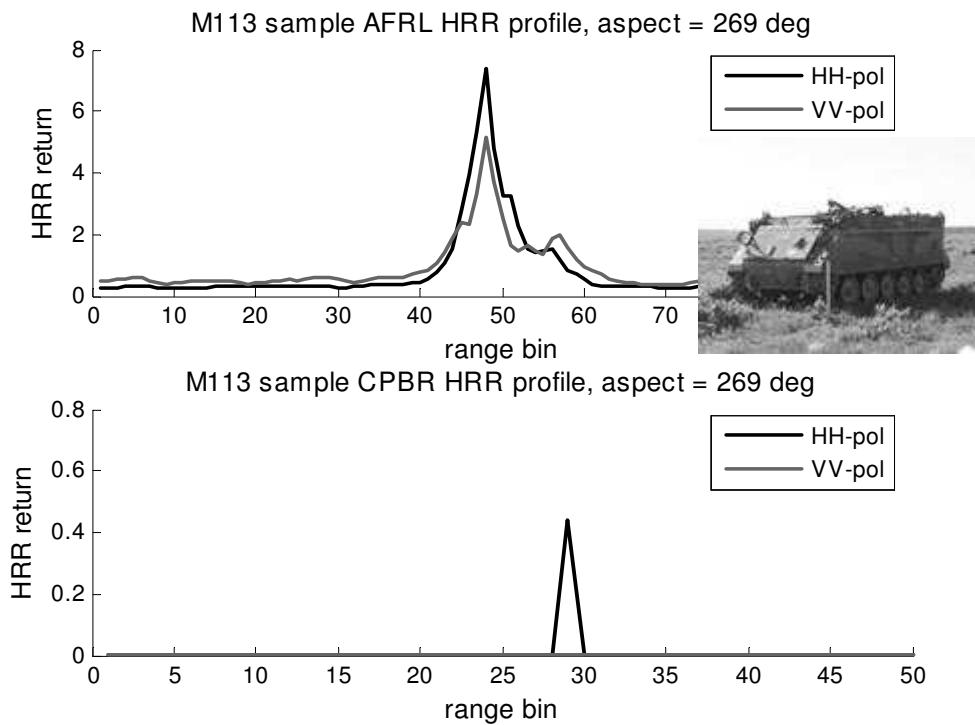
Figure 5.10 Sample M113 HRR Profile (label: Friend)
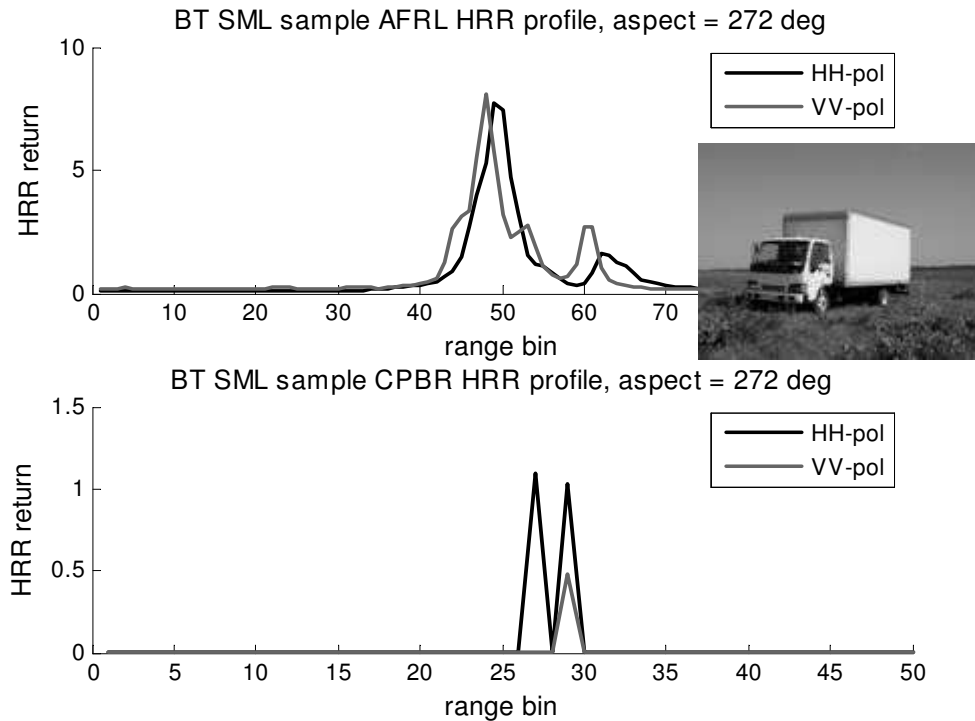
Figure 5.11 Sample Small Truck HRR Profile (label: Neutral)
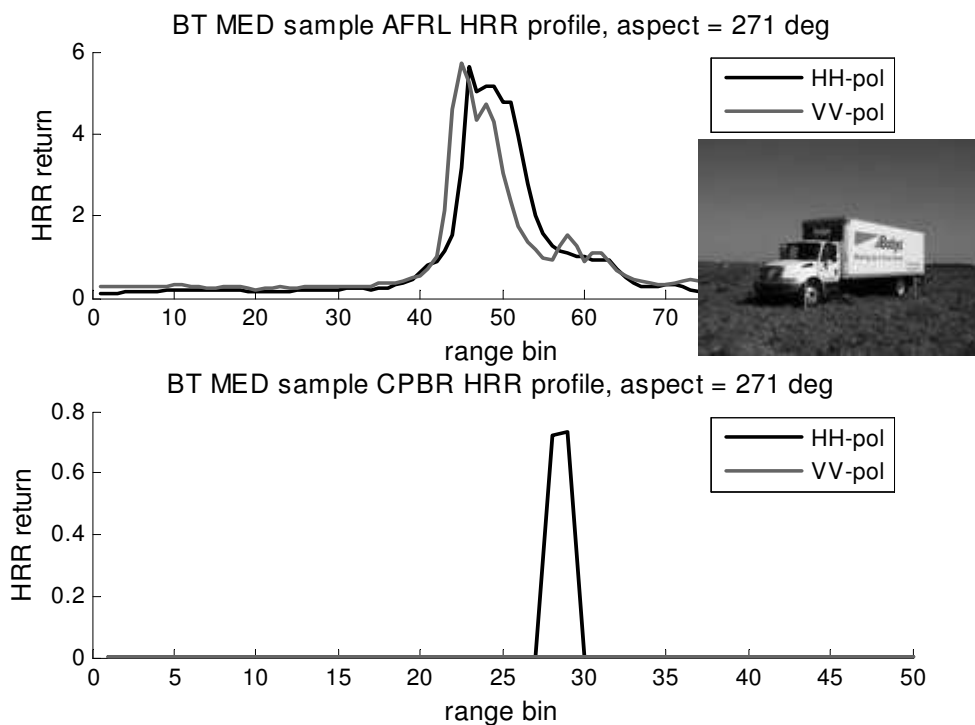
Figure 5.12 Sample Med Truck HRR Profile (label: Neutral)
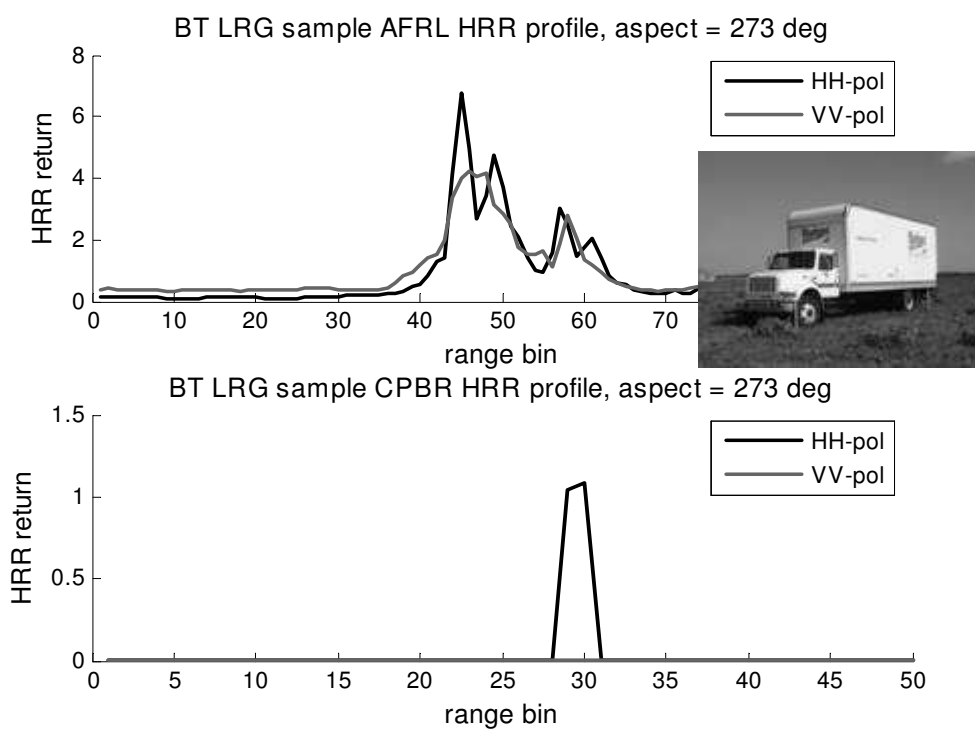
Figure 5.13 Sample Large Truck HRR Profile (label: Neutral)

From the single aspect sample of HRR range profiles, significant differences can be seen in the two processing algorithms. The PBR algorithm developed by Çetin (2001) generates significantly sharper peaks with greater differences from the background returns. This may be a significant aid in performing target recognition when data quality is degraded or reduced (Çetin and William, 2001; Çetin *et al.,* 2003). Yet, with the DCS data collected on a relatively benign background without camouflage, concealment or deception, classification advantages of the PBR algorithm may not be realized and may need to be tested under more stressing EOC to show advantage over the AFRL algorithm. In addition, Çetin's algorithm includes adjustable parameters, which were set at initial values and not changed. These values appear to generate reasonable profiles for HH polarized data, but do not appear to generate VV profiles with a consistent order of magnitude or the desired "peaked" profiles. Overall, the primary reason for use of the PBR HRR algorithm appears valid, from which a second sensor that will yield different classification from the more extensively used AFRL algorithm is obtained. This can be seen in future analysis of the sensor output by vehicle, where the AFRL and PBR HRR algorithms operating on the HH and VV data appear to represent two sensors with different output characteristics.

Once HRR range profiles for each target chip had been processed, features representative of processed sensor data were developed. Standard methods for classification using HRR signatures include generation of feature vectors from the entire HRR range profile or selection of peak amplitudes within desired range bins (Mitchell and Westerkamp, 1998, Williams *et al.,* 2000). For this research, features were generated by taking the peak amplitude for each HRR return from 10 uniformly spaced range bins

after the range bins were filtered to include only those range bins with radar returns significantly different from ground noise. The features associated with Sensor A, HH polarized SAR data processed by Çetin's PBR HRR algorithm, included the peak magnitudes in bins 70-75, 76-81, 82-87, 88-93, 94-99, 100-105, 106-111, 112-117, 118-123 and 124-129 from a total of 200 available bins. The features associated with Sensor B, VV polarized SAR data processed by AFRL's HRR algorithm, included peak magnitudes in bins 115-126, 127-138, 139-150, 151-162, 163-174, 175-186, 187-198, 199-210, 211-222 and 223-234 from a total of 322 bins available.

With the 256×256 complex radar images reduced to a feature vector of ten values, templates were estimated to represent specific ranges of aspect angle for each target vehicle. Figure 5.14 shows the aspect angle convention used by the DCS radar collection. As identified for other template based classification (Duda *et al.* 2001, Meyer, 2003), the Mahalanobis distance, was used to assess each HRR feature vector.

$$\Delta^2 = \left(\boldsymbol{\mu}_{ij} - \mathbf{x} \right)^T \Sigma_{ij}^{-1} \left(\boldsymbol{\mu}_{ij} - \mathbf{x} \right) \tag{5-20}$$

is the Mahalanobis distance squared between feature vector $\mathbf{x}$ and $\boldsymbol{\mu}_{ij}$. Where, $\boldsymbol{\mu}_{ij}$ is the mean of target vehicle i estimated for angular range j and Σ_{ij} is the estimated covariance for target vehicle i and angular range j. Training data were used to estimate the Gaussian parameters, $\boldsymbol{\mu}_{ij}$ and Σ_{ij} across all aspect templates, with $j = 360^o / \# angular\ templates$. In addition, agreement has been found between the use of a multivariate Gaussian approximation of features obtained using radar sensor templates and features generated from simulation of higher fidelity radar returns (Haspert *et al.*, 2004). Further, since the Gaussian distribution has the maximum entropy associated with an observed mean and

185

variance, it should provide for a conservative estimate (Duda *et al.,* 2001: 631). Thus, use of the Mahalanobis distance appears reasonable.
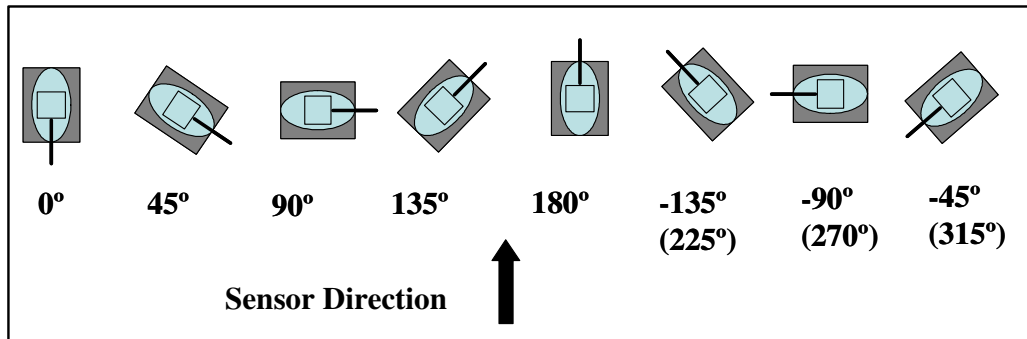
Figure 5.14 Aspect Angle Conventions for Collected Radar Data

With 724 samples of each target processed across 360 degrees, angular templates of 10 and 15 degrees were selected by use for the two sensors. The number of Training and Test images per angular template varies slightly for each target vehicle. The non-uniformity across angular templates is attributable to both the variability in the DCS collection during the numerous flight passes and differences in aspect angles between each vehicle up to 8 degrees. The aspect differences for each vehicle are shown in Figures 5.4 – 5.13, where each vehicle was imaged at the same time by the DCS radar in spot mode. From Figure 5.15, the $j = 36$ angular bins associated with 10 degree templates show the number of samples for one particular vehicle. The number of training images range from about 15 to 25, from which $\boldsymbol{\mu}_{ij}$ and Σ_{ij} are estimated. With sequential looks from one flight pass likely to contribute 2 or 3 data points for each of these templates, it should be noted that covariance may be underestimated since the data samples are not independent. Figure 5.15 also shows the number of Test data samples by angular bin, where higher variability is shown in the number of samples for each angular template.

186
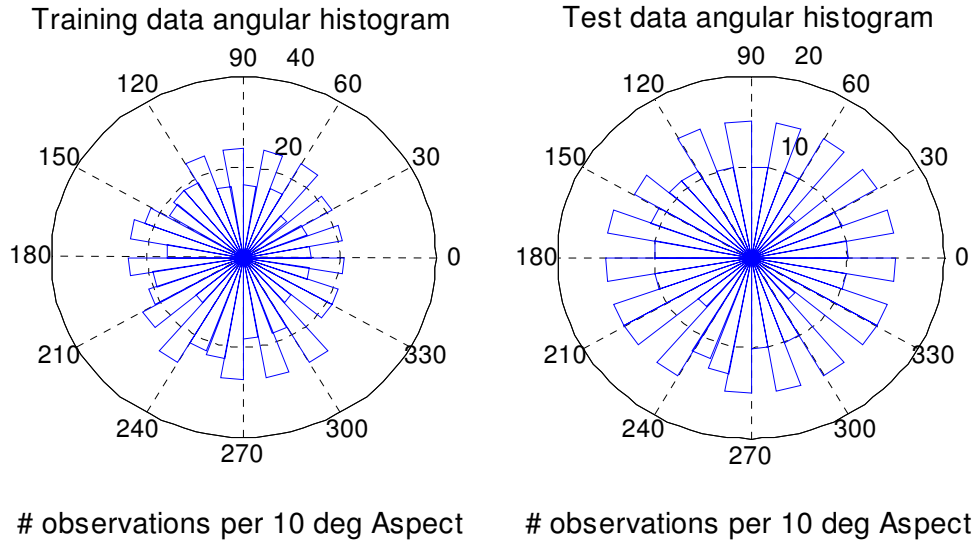
Figure 5.15 Sample Angular Histograms of Training & Test Data for 10° Templates

With at least 10 samples required to estimate Σ_{ij}, angular templates less than 10 degrees were potentially insufficient across certain angles. Angular templates of 15 degrees were used to generate templates for the second sensor. The associated $j = 24$ angular ranges are shown in Figure 5.16 and show the number samples available for training and test.
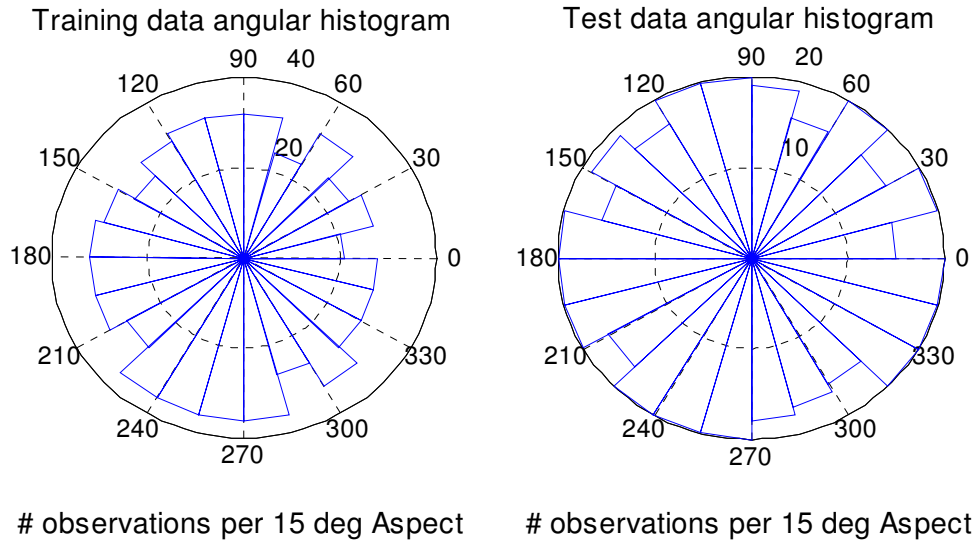
Figure 5.16 Sample Angular Histograms of Training & Test Data for 15° Templates

To gain initial confidence in the processing of the DCS radar data, the Total

Probability of Misclassification (TPM) (Johnson and Wichern, 1998) was calculated for

the Training data. The mean TPM across 24 or 36 Gaussian templates between any two

classes is presented in Table 5.4a for sensor A and Table 5.4b for Sensor B. As desired

by a fusion system, the two sensors appear to provide slightly different classification

results based on this initial pair-wise assessment. From these tables, Sensor A or B may

perform better for a specific classification task between two vehicles. This offers some

initial evidence that fusion of these two sensors may be significantly beneficial.

Table 5.4a Pair-wise TPM for Sensor A for Each of the 10 Target Types using Training
Data, HH Polarization, Çetin's PBR HRR Algorithm and 10 Degree Aspect Templates

	SCUD	SMERCH	SA-6 radar	Med Truck	HMMWV	T-72	M113	Sm Truck	SA-6 TEL	Lg Truck
SCUD	--	5.9%	2.3%	4.0%	0.4%	1.4%	0.6%	2.4%	2.1%	3.0%
SMERCH	5.9%	--	2.8%	3.2%	0.4%	1.1%	0.4%	2.0%	2.6%	2.3%
SA-6 radar	2.3%	2.8%	--	2.9%	1.0%	3.7%	1.2%	3.4%	8.9%	2.2%
Med Truck	4.0%	3.2%	2.9%	--	1.2%	1.7%	0.8%	2.2%	2.5%	4.7%
HMMWV	0.4%	0.4%	1.0%	1.2%	--	1.6%	4.2%	1.8%	1.1%	1.2%
T-72	1.4%	1.1%	3.7%	1.7%	1.6%	--	1.9%	2.2%	4.3%	1.6%
M113	0.6%	0.4%	1.2%	0.8%	4.2%	1.9%	--	1.5%	1.4%	1.3%
Sm Truck	2.4%	2.0%	3.4%	2.2%	1.8%	2.2%	1.5%	--	3.6%	2.0%
SA-6 TEL	2.1%	2.6%	8.9%	2.5%	1.1%	4.3%	1.4%	3.6%	--	1.5%
Lrg Truck	3.0%	2.3%	2.2%	4.7%	1.2%	1.6%	1.3%	2.0%	1.5%	--
TPM sum	22.1%	20.6%	28.4%	23.1%	12.8%	19.5%	13.3%	20.9%	27.8%	19.7%
mean TPM	2.5%	2.3%	3.2%	2.6%	1.4%	2.2%	1.5%	2.3%	3.1%	2.2%

Table 5.4b Pair-wise TPM for Sensor B for Each of the 10 Target Types using Training
Data, VV Polarization, AFRL's HRR Algorithm and 15 Degree Aspect Templates

	SCUD	SMERCH	SA-6 radar	Med Truck	HMMWV	T-72	M113	Sm Truck	SA-6 TEL	Lg Truck
SCUD	--	2.1%	1.1%	3.2%	0.0%	0.2%	0.1%	1.2%	1.1%	1.9%
SMERCH	2.1%	--	2.2%	0.7%	0.0%	0.3%	0.1%	0.5%	1.4%	0.4%
SA-6 radar	1.1%	2.2%	--	2.0%	1.0%	3.1%	0.8%	2.4%	10.0%	2.8%
Med Truck	3.2%	0.7%	2.0%	--	0.4%	2.3%	1.0%	3.0%	1.9%	5.2%
HMMWV	0.0%	0.0%	1.0%	0.4%	--	1.6%	6.2%	3.0%	0.9%	0.9%
T-72	0.2%	0.3%	3.1%	2.3%	1.6%	--	1.8%	1.9%	3.7%	1.0%
M113	0.1%	0.1%	0.8%	1.0%	6.2%	1.8%	--	2.2%	1.2%	0.6%
Sm Truck	1.2%	0.5%	2.4%	3.0%	3.0%	1.9%	2.2%	--	2.5%	3.2%
SA-6 TEL	1.1%	1.4%	10.0%	1.9%	0.9%	3.7%	1.2%	2.5%	--	1.4%
Lrg Truck	1.9%	0.4%	2.8%	5.2%	0.9%	1.0%	0.6%	3.2%	1.4%	--
TPM sum	10.8%	7.6%	25.3%	19.6%	14.0%	15.9%	13.9%	19.9%	24.0%	17.3%
mean TPM	1.2%	0.8%	2.8%	2.2%	1.6%	1.8%	1.5%	2.2%	2.7%	1.9%

The TPM values presented represent an apparent error rate obtained using the same data used to estimate the Gaussian parameters. These values are useful to determine where a majority of misclassifications may occur. For example the largest pair-wise TPM occurs for Sensor B between the SA-6 radar and SA-6 TEL. This may have little operational impact, if neutralizing either of the two vehicles would effectively neutralize the system. Yet, another fairly large TPM occurs for Sensor A between the SCUD and SMERCH, which could have an operational impact and contribute to non-critical error when the desired Target of the Day is the SCUD.

From visual analysis of these TPM tables, along with others generated using different angular template ranges and polarizations of the data, a few insights were gained. First, for each HRR algorithm, TPM tended to increase for templates of increased angular range. Yet, the performance trends for Test data are unknown, with respect to template size. Larger templates may generalize better, especially if performance degradation is more significant for smaller templates between Training and Test data. Next, overall vehicle size appears to be a good indicator of potential discrimination between targets. For example, the HMMWV is smallest of the 15 targets with low TPM between most other targets. As the largest vehicles, the SCUD and SMERCH also show low TPM between other vehicles, while the PBR HRR algorithm TPM between these two vehicles is relatively high at 6%.

While the TPM associated with the true aspect angles provides a general idea of sensor discrimination, perfect aspect angle information is unlikely to be available. By assuming a moving target indicator (MTI) sensor is used to acquire the track of a potential target prior to or concurrent with the HRR data collection, the associated target

track can be used to estimate the vehicle's aspect angle within +/- 15 degrees (Williams *et al.*, 2000). Assuming this estimated aspect angle information is available, template based matching was performed by computing the Mahalanobis distance for each vehicle's feature vector, **x**, to each of the angular templates for all ten vehicles. Template matching was performed using the ground-truth of the imaged vehicle to determine the first angular template to be searched. To account for the +/- 15 degrees of aspect angle estimation by an MTI system, the Mahalanobis distances from the angular templates occurring before and after the true aspect angle were also evaluated. A total of 30 templates were used to compute 30 Mahalanobis distances for each imaged vehicle's HRR feature vector. The minimum Mahalanobis distance for each of the ten target types was then used to compute a bounded score associated with each vehicle type. An one-dimensional 'z-score' $\in [0,1]$ associated with each target type's distance measure, Δ_i, was computed as:

$$p_i = p(\Delta_i) = \quad = \frac{1}{\sqrt{2\pi}} e^{-1/2(\Delta_i^2)}. \tag{5-21}$$

While some assumptions of obtaining *i.i.d.* samples for Gaussian parameters were violated in generating these probability scores, such as independence between observations, it is used as a mapping to obtain reasonable scores between 0 and 1.

Posterior probability estimates for each of the three desired output labels were then obtained by normalizing probability scores by the sum of all classes for each observation.

$$\text{TOD posterior probability} = ppTOD = \frac{p_{SCUD}}{\sum\limits_{i=1}^{10} p_i} = \frac{p_1}{\sum\limits_{i=1}^{10} p_i} \tag{5-22}$$

OH posterior probability =

$$ppOH = \frac{p_{SMERCH} + p_{SA-6\,Radar} + p_{SA-6TEL} + p_{T-72}}{\sum\limits_{i=1}^{10} p_i} = \frac{p_2 + p_3 + p_4 + p_5}{\sum\limits_{i=1}^{10} p_i} \qquad (5\text{-}23)$$

FN posterior probability =

$$ppFN = \frac{p_{HMMWV} + p_{M113} + p_{SmTruck} + p_{MedTruck} + p_{LgTruck}}{\sum\limits_{i=1}^{10} p_i} = \frac{p_6 + p_7 + p_8 + p_9 + p_{10}}{\sum\limits_{i=1}^{10} p_i} \qquad (5\text{-}24)$$

Thus, for any target being assessed, the final posterior probabilities sum to one,

$$ppTOD + ppOH + ppFN = 1. \qquad (5\text{-}25)$$

A plot of the SCUD posterior probabilities associated with a Hostile declaration of

"TOD" or "OH" is shown in Figure 5.17 for each sensor and both data sets, where

$$\text{Hostile (H) posterior probability} = ppH = ppTOD + ppOH. \qquad (5\text{-}26)$$

Viewing the plots in Figure 5.17, in most cases the sensors do a good job of correctly

estimating the posterior probability of Hostile as close to 1. Similar figures representing

all 10 target types are included in Appendix A. From visual analysis of these plots, it

appears as if Sensors A and B, provide different target information associated with the

radar aspect angle. For example, the posterior probability estimates of being a Hostile

enemy tend to be correct, except for some side views of the SCUD associated with aspect

angles centered about 270 degrees and to a lesser extent, at the opposing 90 degree aspect

angle. This reduction in sensor performance, when the vehicles are being classified by

broadside views, using HRR signatures agrees with research by Williams *et al.,* (2000)

and should be expected since the HRR processing algorithms generate a mean range

profile across the width of the vehicle. In contrast, better HRR features may be obtained

191

by imaging vehicles from the front or rear, from which estimates of relative vehicle length may aid in the discrimination effort. While informative, one deficiency of these angular posterior probability plots is that the number of incorrect Hostile posterior probabilities close to 0 is not ready visible. Since the figures plot a single dot at any probability/angle combination and do not indicate the frequency of occurrence, Tables 5.5 and 5.6 are presented to summarize sensor performance.
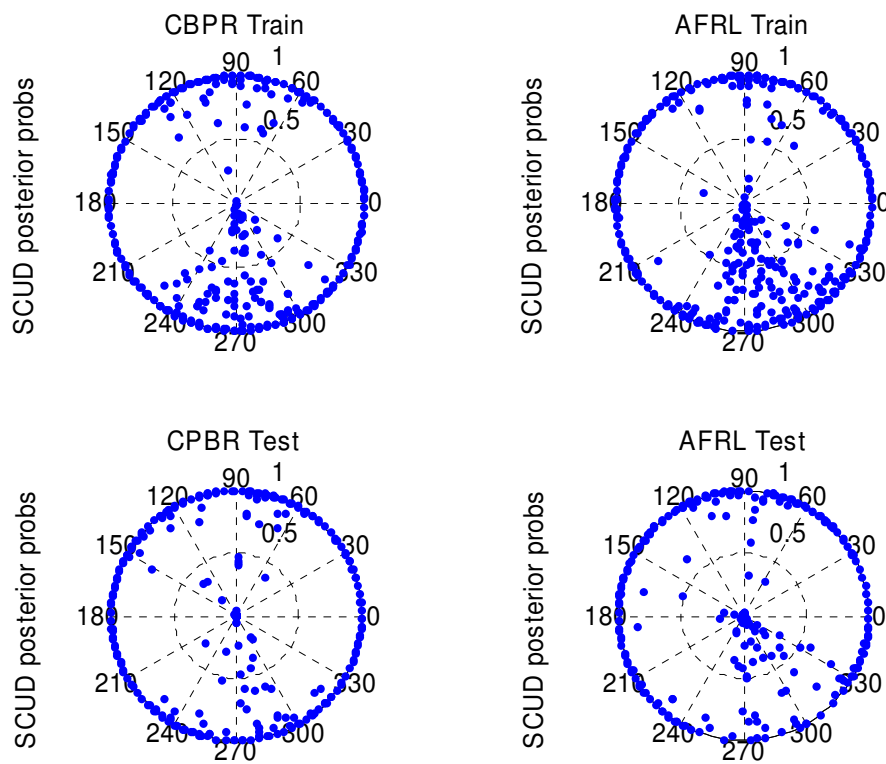
Figure 5.17 Sensor A & B Posterior Probability of Hostile by Aspect Angle for All SCUD DCS Radar Imagery Included in the Training and Test Data Sets

To gain further insight of each sensor's single look performance, assessments were made at the dichotomous Hostile vs. Friend/Neutral level by using set thresholds

and the posterior probability of Hostile (*ppH)* as a single value to determine class

membership from. For the following table, the mean True Class and False Class

estimates can be modeled as binomial random variables. This facilitates calculation of

confidence intervals on these variables, where an approximate 90% (1-α) confidence

interval, with 724 training samples yields an associated CI of approximately +/- 2-3%,

while the CI associated with the test data with 446 samples is slightly higher at +/- 3-4%.

    As shown earlier in Figure 3.3, the relations between thresholds and labels are

depicted in the following figure. This shows the classification labels for Hostile vs.

Friend two-class data represented by the two histograms with different grey colors.
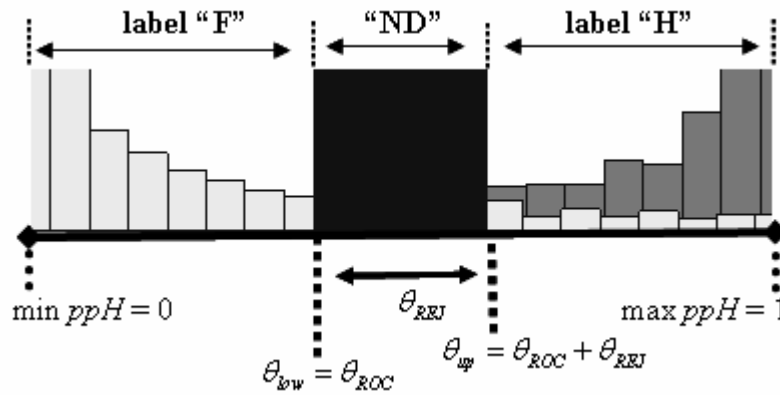
Figure 5.18 Example Relations and Labels for given Values of θ_{low} and θ_{up}

Table 5.5 Sample Sensor Performance by Target Type using Training and Test Data for $\theta_{low} = \theta_{up} = 0.5$ (No Rejection Option, Classify as "H" if $ppH > 0.5$)

Type	Label	Sensor A Training Data			Sensor B Training Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	4%	96%	0%	7%	93%	0%
SMERCH	OH	7%	94%	0%	3%	97%	0%
SA-Radar	OH	6%	94%	0%	15%	85%	0%
T-72	OH	8%	92%	0%	13%	87%	0%
SA-6 TEL	OH	5%	95%	0%	15%	86%	0%
Med Truck	FN	91%	9%	0%	98%	2%	0%
HMMWV	FN	90%	10%	0%	98%	2%	0%
M113	FN	90%	10%	0%	98%	2%	0%
Sm Truck	FN	82%	18%	0%	98%	2%	0%
Lg Truck	FN	98%	2%	0%	99%	1%	0%
		mean True Class		92.2%	mean True Class		93.7%
		mean False Class		7.9%	mean False Class		6.3%
		mean rejection		0.0%	mean rejection		0.0%

Type	Label	Sensor A Test Data			Sensor B Test Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	11%	89%	0%	21%	79%	0%
SMERCH	OH	11%	90%	0%	11%	90%	0%
SA-Radar	OH	12%	88%	0%	27%	74%	0%
T-72	OH	21%	79%	0%	35%	65%	0%
SA-6 TEL	OH	14%	86%	0%	32%	68%	0%
Med Truck	FN	70%	30%	0%	92%	8%	0%
HMMWV	FN	81%	19%	0%	97%	3%	0%
M113	FN	76%	24%	0%	96%	4%	0%
Sm Truck	FN	66%	34%	0%	93%	7%	0%
Lg Truck	FN	84%	16%	0%	96%	5%	0%
		mean True Class		80.8%	mean True Class		84.8%
		mean False Class		19.2%	mean False Class		15.2%
		mean rejection		0.0%	mean rejection		0.0%

Thus, without a Rejection option, the sensors each perform Hostile vs. Friend classification at an apparent 90% +/-3% or better for training data, while the test data shows classification accuracy closer to 80% +/-4%. Table 5.6 shows considerable improvement in classification accuracy given a centered rejection window of width 0.80. Test data classification accuracy is now close to 90% +/-4% for each individual sensor. Yet, with a desired critical error of 2% or less, significant improvement will need to be realized by the fusion systems to obtain feasible solutions.

194

Table 5.6 Sample Sensor Performance by Target Type using Training and Test Data for $\theta_{low} = 0.10$ and $\theta_{up} = 0.9$ (Rejection Occurs if $0.10 < ppH < 0.90$)

Type	Label	Sensor A Training Data "F"	"H"	% Rej	Sensor B Training Data "F"	"H"	% Rej
SCUD	TOD	1%	88%	11%	1%	84%	15%
SMERCH	OH	1%	83%	16%	0%	91%	9%
SA-Radar	OH	1%	81%	18%	4%	65%	31%
T-72	OH	1%	74%	25%	3%	70%	27%
SA-6 TEL	OH	1%	77%	23%	4%	67%	29%
Med Truck	FN	79%	3%	18%	93%	1%	6%
HMMWV	FN	70%	2%	28%	95%	1%	5%
M113	FN	69%	1%	30%	93%	1%	6%
Sm Truck	FN	55%	4%	41%	93%	1%	7%
Lg Truck	FN	93%	0%	7%	97%	0%	3%

		mean True Class | dec	97.9%	mean True Class | dec	98.3%
mean False Class | dec	2.1%	mean False Class | dec	1.7%
mean rejection	21.7%	mean rejection	13.7%

Type	Label	Sensor A Test Data "F"	"H"	% Rej	Sensor B Test Data "F"	"H"	% Rej
SCUD	TOD	7%	84%	9%	16%	74%	11%
SMERCH	OH	5%	70%	26%	6%	79%	16%
SA-Radar	OH	4%	70%	27%	11%	57%	31%
T-72	OH	8%	58%	34%	17%	45%	38%
SA-6 TEL	OH	5%	65%	30%	15%	43%	42%
Med Truck	FN	54%	15%	31%	83%	5%	12%
HMMWV	FN	58%	11%	31%	93%	1%	7%
M113	FN	52%	11%	37%	89%	2%	9%
Sm Truck	FN	39%	17%	44%	85%	3%	12%
Lg Truck	FN	73%	8%	19%	92%	2%	7%

		mean True Class | dec	87.3%	mean True Class | dec	90.5%
mean False Class | dec	12.7%	mean False Class | dec	9.5%
mean rejection	28.7%	mean rejection	18.3%

To provide some insight as to why the test data sensor performance degraded, the number of correct angular template matches was assessed for the true target types. From Table 5.7, the search of 3 templates by the training data resulted in a correct template match approximately 90% of the time, while the test data only selected the correct angular template associated with the true target type about 60% of the time.

Table 5.7 Correct Matches by Aspect Angle for Training & Test Sets

3 TEMPLATE SEARCH

SENSOR A - TRAINING				SENSOR B - TRAINING			
10 deg templates, HH-polar, PBR algorithm				15 deg templates, VV-polar, AFRL algorithm			
target type	correct template	next template	previous template	target type	correct template	next template	previous template
SCUD	89.2%	5.4%	5.4%	SCUD	85.1%	8.8%	6.1%
SMERCH	93.4%	3.6%	3.0%	SMERCH	89.2%	4.3%	6.5%
SA-6 radar	89.0%	6.5%	4.6%	SA-6 radar	83.0%	8.7%	8.3%
Med Truck	91.3%	2.9%	5.8%	Med Truck	89.1%	7.5%	3.5%
HMMWV	96.1%	1.9%	1.9%	HMMWV	90.6%	5.1%	4.3%
T-72	91.9%	3.6%	4.6%	T-72	86.7%	6.6%	6.6%
M113	94.2%	3.5%	2.4%	M113	86.7%	6.6%	6.6%
Sm Truck	93.7%	3.7%	2.6%	Sm Truck	89.5%	4.8%	5.7%
SA-6 TEL	92.3%	3.6%	4.1%	SA-6 TEL	85.8%	7.7%	6.5%
Lrg Truck	90.2%	5.4%	4.4%	Lrg Truck	92.8%	3.0%	4.1%
mean	**92.1%**	**4.0%**	**3.9%**	**mean**	**87.9%**	**6.3%**	**5.8%**

SENSOR A - TEST				SENSOR B - TEST			
target type	correct template	next template	previous template	target type	correct template	next template	previous template
SCUD	51.4%	18.4%	30.3%	SCUD	55.4%	24.2%	20.4%
SMERCH	62.8%	15.0%	22.2%	SMERCH	56.5%	16.8%	26.7%
SA-6 radar	57.4%	22.4%	20.2%	SA-6 radar	56.5%	22.2%	21.3%
Med Truck	59.0%	21.8%	19.3%	Med Truck	65.3%	20.2%	14.6%
HMMWV	59.9%	22.2%	17.9%	HMMWV	62.8%	15.0%	22.2%
T-72	63.9%	19.5%	16.6%	T-72	59.0%	19.3%	21.8%
M113	61.9%	20.9%	17.3%	M113	61.0%	20.6%	18.4%
Sm Truck	62.1%	18.4%	19.5%	Sm Truck	57.2%	20.4%	22.4%
SA-6 TEL	59.6%	21.1%	19.3%	SA-6 TEL	59.4%	19.7%	20.9%
Lrg Truck	56.7%	25.6%	17.7%	Lrg Truck	64.8%	19.1%	16.1%
mean	**59.5%**	**20.5%**	**20.0%**	**mean**	**59.8%**	**19.8%**	**20.5%**

Overall, the sensor data generated to represent Sensors A and B appears to do a relatively good job of classification at the dichotomous Hostile vs. Friend level. Classification improvement can be made by allowing rejection as demonstrated by the single look mean True class assessments presented in Tables 5.5 (no rejection) and Table 5.6 (with rejection). Using these two sensors, with relatively good performance, as input for two fusion systems, it is hoped that the desired Combat ID requirements can be achieved as identified by the operational constraints levied by the warfighter.

5.3.3 Generation of Data Sets with Different Correlation within and across Sensor Looks

In order to assess some of the potential effects of possible observed correlation, four different methods were developed to determine the next look by a sensor and to determine the relationship between the two different sensors. The first of these methods used the DCS data in the natural order (ord) it was collected to obtain the next look with approximately 4 degrees of aspect angle separation between looks. Sensors A and B were also co-registered with simultaneous looks of each ground vehicle. This natural ordering provides for a continuous progression of both aspect and depression angles. All flight passes selected for use by the Training and Test sets included 22 or 23 images in each polarization. Sequences of 5-looks were generated. This was accomplished by starting with the 1st five observations, skipping the 6th observation, then taking another 5-look sequence, skipping the next if 23 images were available, then selecting the next 5, skipping the next, and using the last 5 looks as the final sequence. Thus, using $t\#$ to indicate the sequential observation,

for 22 looks: $seq1 = \{t1 - t5\}, seq2 = \{t7 - t11\}, seq3 = \{t12 - t16\}, seq4 = \{t18 - t22\}$
for 23 looks: $seq1 = \{t1 - t5\}, seq2 = \{t7 - t11\}, seq3 = \{t13 - t17\}, seq4 = \{t19 - t23\}$

By including a one-look temporal space between sequences, hopefully effects of autocorrelation between naturally occurring sequences will be minimized. Using this method for the 32 flight passes in the Training data set yielded 128 sequences of 5-looks for each vehicle, with 1280 total sequences. The 20 flight passes in the Test data set provided 80 five-look sequences for each vehicle, with 800 total sequences. Three more

data sets were generated with different correlation structures within and across sensors. All data sets were generated with the same number of training and test sequences.

The remaining three data sets were generated to represent sensors with lower levels of correlation across sensors at a given time or within each sensor through time. Data to represent autocorrelated individual sensors was generated by randomly pairing one of the naturally occurring sequences of data from Sensor A with a naturally occurring sequence from Sensor B. This may be representative of two different platforms imaging a ground target at the same time with re-looks, but at different aspect angles. The next data set was generated using co-registered aspect angle looks by sensors A and B, but instead of using a natural sequence of looks, each of the co-registered looks was randomly selected from the available data, without replacement. This generation of co-registered data may be representative of the data collected by 5 different flight passes at different aspect angles by one platform hosting both sensors A and B. The final data set was generated in an attempt to create independent sensor data both across sensors and through temporal looks. For a given time t, both Sensor A and B represent looks at random aspect angles. Each sensor's multiple looks are also at random angles. This data set may represent two platforms each hosting a different sensor and each taking five different flight passes in the attempt to ID a ground target. As indicated in Table 5.8, the abbreviations of "ord," "aut," "cor," and "ind" will be used to refer to these four data sets representing the different correlation structures and are summarized as follows:

- ord = naturally ordered data, co-registered & autocorrelated through time
- aut = autocorrelated individual sensors, not co-registered
- cor = co-registered sensors independent through time
- ind = independent sensors independent through time.

Table 5.8 Summary of Characteristics for Each of the Four Data Sets

Naturally ordered (ord):

- Sensors A & B collected at same time and same aspect angle
- Temporal looks occurred naturally in data (approximately 4 degrees between looks)
- Both correlation across Sensors and autocorrelation within multi-looks by a sensor
- ex. One 2-sensor platform collecting data with real time re-looks during 1 flight pass

Autocorrelated individual sensors (aut):

- Sensors A & B collected at different aspect angles
- Temporal looks occurred naturally in data (approximately 4 degrees between looks)
- Autocorrelation between multi-looks
- Independence between sensors A & B
- ex. Two 1-sensor platforms collecting data with real time re-looks during 1 flight pass

Co-registered sensors independent through time (cor):

- Sensors A & B collected at same aspect angles (co-registered)
- Temporal looks taken randomly from data
- Independence between multi-looks
- Co-registration between sensors A & B at any time t
- ex. Up to 5 flight passes at different aspects angles by one 2-sensor platform

Independent sensors independent through time (ind):

- Sensors A & B collected at random aspect angles
- Temporal looks taken randomly from data
- Independence between multi-looks of each sensor through time
- Independence between sensors A & B at any time t
- ex. Up to 5 flight passes at different aspects angles by two 1-sensor platforms

5.4 Majority Vote Boolean (MVB) Fusion Methodology

The Majority Vote Boolean (MVB) fusion method uses predetermined Boolean logic to determine the final output label at any time t. The input for this fusion logic includes the labels for Sensors A and B associated with time t along with the labels from both sensors for all preceding looks. With input labels from both sensors, a majority vote winner for the 1st look would be a "Non-declaration" unless both sensors concurred as

199

"TOD," "OH" or "FN." In addition, if any ties occur with an equal number of labels being fused at a particular time period a "Non-declaration" decision is made. Another situation leading to a "Non-declaration" label is if at time t the majority of labels were "ND." This Boolean logic allows for many situations in which a "Non-declaration" may occur. Thus, this logic appears to be inherently conservative with many options to provide a "ND" rejection label when there is disagreement between the sensor labels.

Sensor labels were generated from the posterior probabilities associated with each HRR profile. To optimize the Boolean fusion, an upper and lower threshold was varied independently between the two sensors to determine the four optimal thresholds for the system. These four thresholds were used to make an initial "Hostile," "Friend," or "Non-declaration" label for each of the two sensors. Since $ppH + ppFN$ sum to one, decisions for each sensor were made based on just ppH:

$$label = \left\{ \ "H" \text{ if } ppH \ > \ \theta_{up}, \ "F" \text{ if } ppH \ \leq \ \theta_{low}, \ "ND" \text{ if } \ \theta_{low} < ppH \leq \theta_{up} \right\} \quad (5\text{-}27)$$

If an initial label was "H," then a set threshold was used to determine if a final label of "TOD" or "OH" was used, where

$$label = \left\{ \ "TOD" \text{ if } ppTOD \ > \ \theta_{TOD}, \ "OH" \text{ if } ppTOD \ \leq \ \theta_{TOD} \right\}, \quad (5\text{-}28)$$

$ppTOD = \dfrac{ppTOD}{ppTOD + ppOH}$, and θ_{TOD} was initially set at 0.8 with a ratio of TOD:OH = 1:4 for the Training data. The lower and upper thresholds were varied from 0.0 through 1.0 with a maximum difference of 0.90. For each sensor, the rejection threshold was equal to the difference in the lower and upper thresholds, $\theta_{REJ} = \theta_{up} - \theta_{low}$. The ROC threshold, for each sensor, equals the lower bound of the rejection window, $\theta_{ROC} = \theta_{low}$.
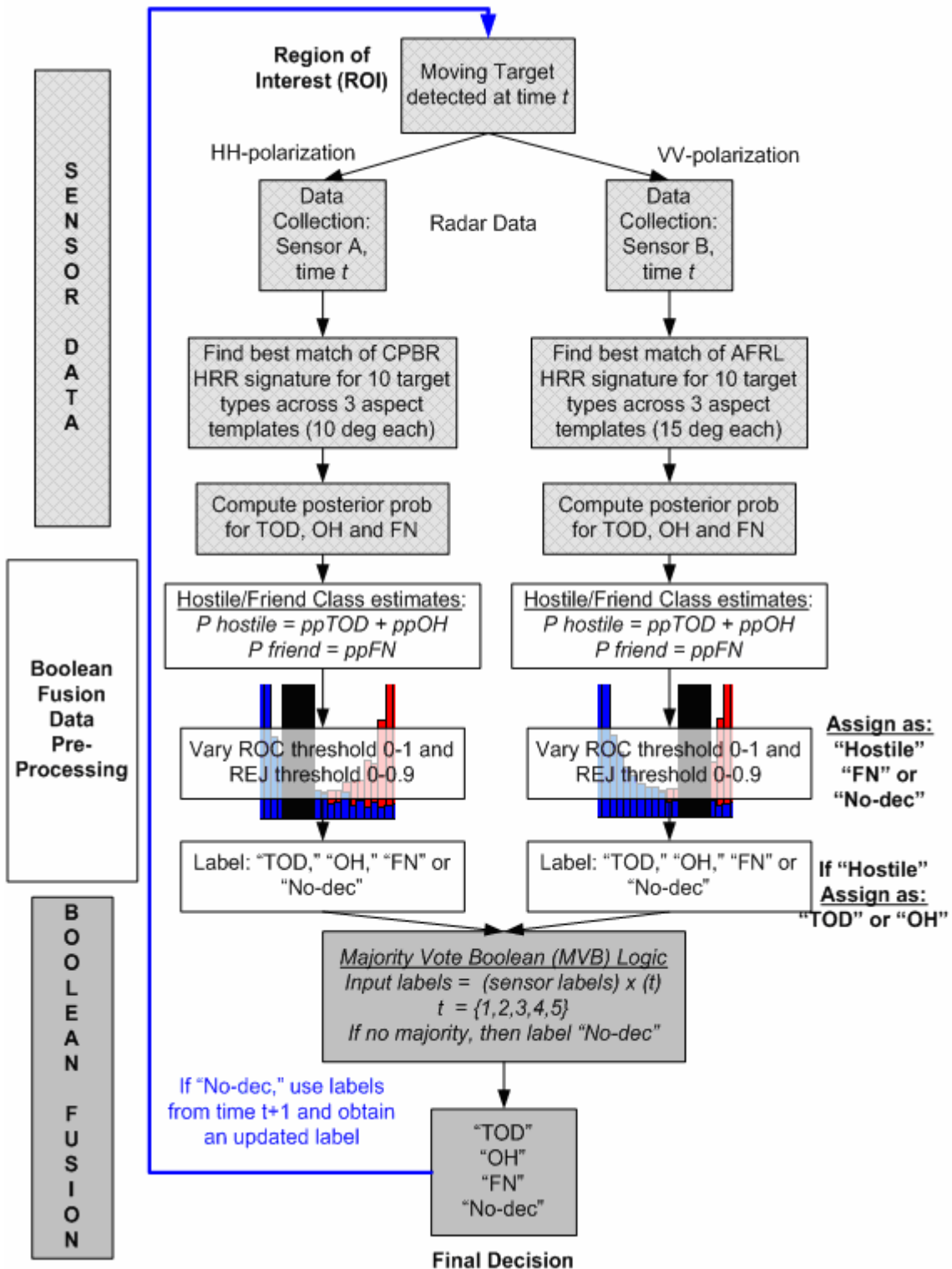
Figure 5.19 Overview of Majority Vote Boolean Fusion

To search the threshold space of $\Theta^{MVB} = \{\Theta^{SA} \times \Theta^{SB}\} = \{\theta^{SA}_{ROC} \times \theta^{SA}_{REJ} \times \theta^{SB}_{ROC} \times \theta^{SB}_{REJ}\}$,

the rejection window was varied uniformly across 10 values {0.0, 0.1, 0.2, 0.3, 0.4, 0.5,

0.6, 0.7, 0.8, 0.90} independently for each sensor. The ROC window was then varied

independently for each sensor uniformly across 10 values from 0.0 through $1 - \theta_{REJ}$. This

process provides for systematic generation of "H," "F," and "ND" labels for each sensor.

A "TOD" vs. "OH" declaration was then made using the set threshold if an "H" was

declared. A search of $\Theta^{MVB} = \{\Theta^{SA} \times \Theta^{SB}\} = (10 \times 10) \times (10 \times 10) = 10,000$ thresholds was

evaluated by the MVB fusion for each data set and number of minimum forced looks.

By forcing a minimum number of looks, the rejection threshold was effectively

set to 1.0 ($\theta_{REJ} = 1.0$, $\theta_{ROC} = 0.0$, $\theta_{low} = 0.0$ and $\theta_{up} = 1.0$) for time periods less than the

minimum looks. This allowed for systems to collect a minimum number of looks before

generating an output label other than "ND." The fusion of looks greater than or equal to

the minimum looks could then be feasible, with declaration rates meeting the operational

constraints. For example, if the rejection window was set to 1.0 across all looks, then the

final output label would always yield "ND" and would never meet the final declaration

constraint of 70% or better. While use of the majority vote Boolean logic may not be the

optimal Boolean logic, it does provide for a reasonable fusion rule. For this pre-selected

Boolean logic, optimization of the minimum look and threshold constraints is

subsequently performed to optimize the fusion algorithm. This is performed by

determining the maximum TPR of the system, without use of cost information and

without assumptions of independence between the sensors.

5.5 Probabilistic Neural Network (PNN) Fusion Methodology

The Probabilistic Neural Network (PNN) fusion method uses multiple trained PNN models to determine a fused posterior probability of "TOD," "OH" or "FN" using probability estimates from Sensors A and B as input features. The final output label at any time t, is determined by post processing fused posterior probability estimates of "TOD," "OH" and "FN" for given ROC and rejection thresholds, where $\mathbf{\theta}^{PNN} = \left(\theta_{ROC}, \theta_{REJ} \right)^{T}$. The input for PNN fusion at time t includes the posterior probabilities generated from Sensors A and B associated with time t along with the posterior probabilities from both sensors for all preceding looks. Thus, five total PNNs were trained, with one for each time period, to incorporate the 3-class posterior probabilities as input features across all available looks (1, 2, 3, 4 or 5).

Figure 5.19 shows the overall PNN fusion process. Similar to the label generation for each of the two sensors in the Boolean fusion, the final label for the PNN fusion began with the top-level "H," "F" or "ND" decision. As before, this decision was made based on just ppH, where $ppH = ppTOD + ppOH$, and

$$label = \left\{ \ "H" \ \text{if} \ ppH \ > \ \theta_{up}, \ "F" \ \text{if} \ ppH \ \leq \ \theta_{low}, \ "ND" \ \text{if} \ \theta_{low} < ppH \ \leq \ \theta_{up} \right\}, (5\text{-}29)$$

where $\theta_{low} = \theta_{ROC}$ and $\theta_{up} = \theta_{low} + \theta_{REJ}$. If the initial label was "H," then the same threshold $\left(\theta_{TOD} = 0.8 \right)$ was used to determine a final label of "TOD" or "OH," where

$$label = \left\{ \ "TOD" \ \text{if} \ ppTod \ > \ \theta_{TOD}, \ "OH" \ \text{if} \ ppTod \ \leq \ \theta_{TOD} \right\}, \quad\quad (5\text{-}30)$$
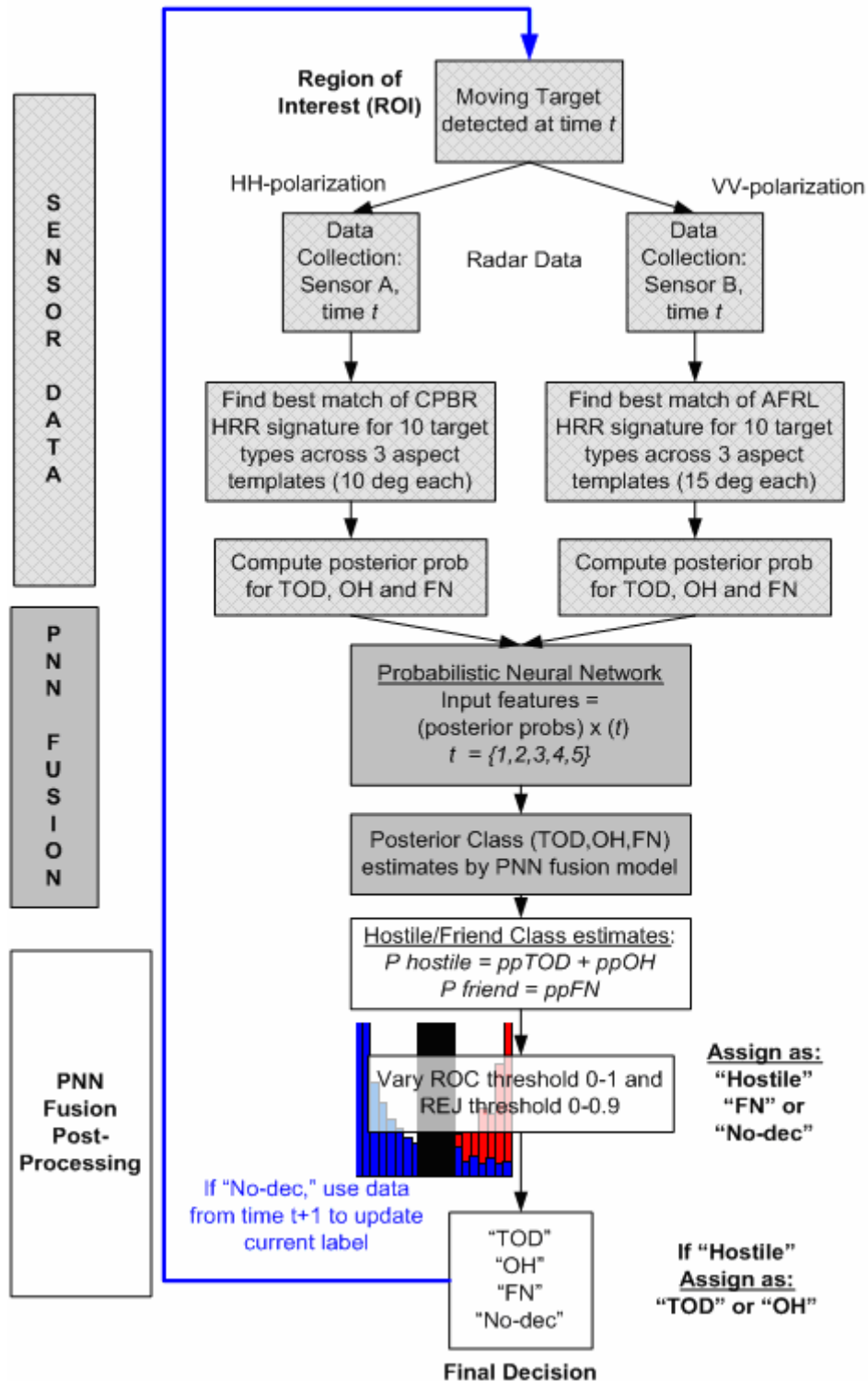
Figure 5.20 Overview of PNN Fusion

and *ppTod* is the posterior probability of a TOD given a hostile declaration. As used for the generation of sensor labels for Boolean fusion, the lower and upper thresholds were varied from 0.0 through 1.0 with a maximum difference of 0.90. Optimization of TPR was performed across $\Theta = \{\boldsymbol{\theta}^{PNN} : \boldsymbol{\theta}^{PNN} = (\theta_{ROC}, \theta_{REJ})\} = (100) \times (100) = 10,000$ thresholds for each data set and for each number of minimum forced looks.

The PNN fusion was accomplished using *Matlab's Neural Network Toolbox*. All available data from the Training set were used to train each of the five PNNs. The training set included 1280 exemplars for each time period, evenly divided between the Hostile and Friend classes, with 1/5 of the Hostile class generated from SCUD data to represent the TOD. Initial training was performed across a range of PNN spread values using a subset of the training data. From these initial runs, the default spread = 0.10 of the PNN function appeared to be an appropriate value with good training and test classification accuracy at the "H" vs. "F" level. A sample plot using all 5 looks of autocorrelated (aut) data is provided, where 1280 Training data samples were divided between a training and test set to assess different values of the PNN spread from 0.05 through 2.0. Similar plots were visually assessed across the range of minimum looks and across all four data sets with different within and across sensor correlation. In general, perfect CA was obtained by the training set over a large range, while the test set performance may start to degrade as the spread value increased. From these plots, a spread value was selected based on the dichotomous top-level Hostile vs. Friend decision without assessing the effects of Non-declarations.
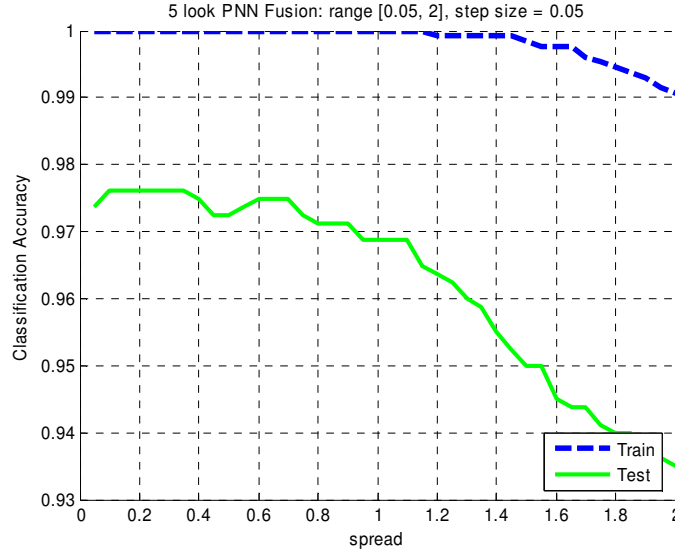
Figure 5.21 PNN Fusion across a Divided Training Set used to Select the Spread

The PNN models used for this fusion experiment were then trained using all 1280 samples of available data for each time period, which may not accurately represent some multi-look scenarios, in which only those hard to classify "Non-declaration" vehicles are sensed an additional time. If the multi-look PNN models were trained with only those exemplars previously rejected, a limited number of previously "non-declared" observations may severely limit data for training the PNN fusion models associated with a next look. The set of "ND" exemplars would also vary significantly by the specific values of θ_{ROC} and θ_{REJ} being used by the multi-look fusion scheme. Therefore, each PNN model was trained using all 1280 available Training observations which would hopefully converge reasonably to generate unbiased probability estimates of all three class labels, regardless of previous "Non-declarations." In summary, five PNNs were trained for each of the four data sets defined by the sensor correlation structure, and subsequent optimization of thresholds will then determine the best PNN fusion model.

5.6 Initial Comparison of Fusion Systems

The initial comparison of fusion models includes the evaluation of both PNN and MVB fusion methods across the range of minimum looks required before making a final system declaration and use of Training or Test data with one of the four correlation structures. Evaluation using the training data was primarily performed to validate performance of the fusion methods prior to comparing them using the Test data. The initial evaluation of both fusion methods required assessment across 5 levels of minimum looks {1,2,3,4,5}, 2 data types {TR,TE}, and 4 data correlation types {ord, cor, aut, ind}, for a total of 40 different estimates for each of the two fusion methods. A total of 80 evaluations were performed. The evaluation of the 40 MVB fusion model/data combinations required approximately 12 hours to compute sensor labels, fuse the labels, and analyze all 10,000 threshold gridpoints. The mixed variable programming formulation was implemented using code developed in *Matlab* and processed on a dedicated 2.66 GHz dual processor desktop with 2.0 GB of RAM. The PNN fusion was completed in two stages. First, 20 total PNN models were trained with input associated with 1 to 5 looks and a given correlation structure. Next, the output associated with all 5-looks for each sequence was saved as a single data file for all 40 Training and Test data sets. Finally, a *Matlab* routine was developed to analyze the output data across all threshold values, Θ^{PNN}. This evaluation was much quicker than the Boolean fusion with assessment of all 40 data sets completed in approximately 30 minutes.

Some initial results are presented by plotting the ROC curves associated with all thresholds evaluated, along with indicating where the maximum TPR occurs, provided a

feasible solution could be obtained. For the training data, only the ROC plots associated with a minimum of 1-forced look are presented. The plot of the ROC curves generated for each of the four correlation structures (ord, cor, aut & ind) are included in the next figure and show nearly perfect ROC curves obtained using PNN fusion with Training data. The stars indicate the point of maximum TPR while the dark dots behind the stars indicate other feasible operating points associated with different thresholds.
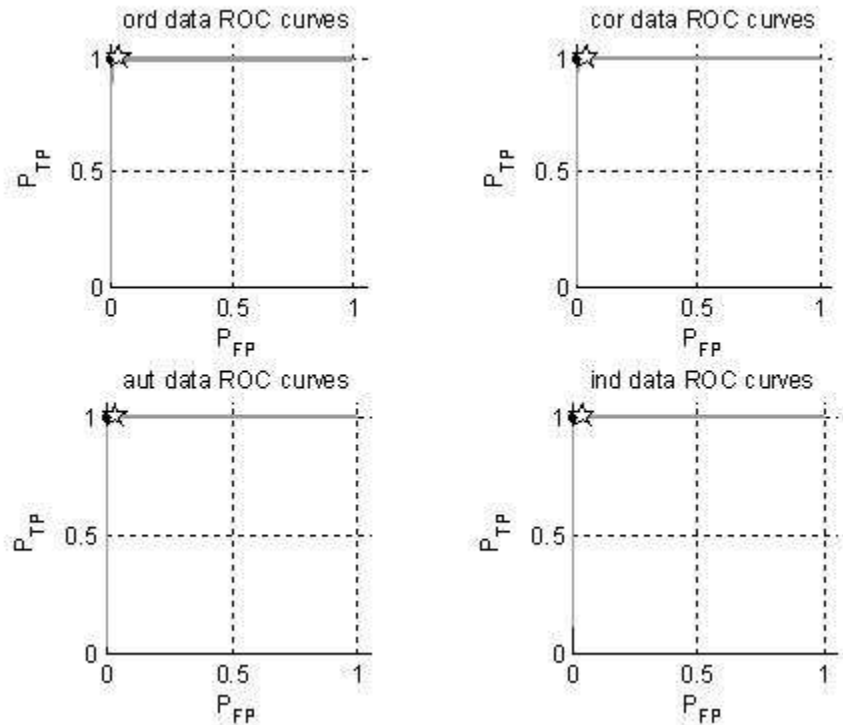
Figure 5.22 ROC Curves Generated from Training Data using PNN Fusion

Each PNN fusion subplot includes 100 ROC curves generated with 100 different Non-declaration windows. From the Table 5.9, the optimal TPR is obtained with a rejection window set to declare "Friend/Neutral" if the posterior probability is < 0.01, followed by a rejection window with a width of approximately 0.25. "Hostile" declarations are made for any PNN hostile probability output above $0.19 – 0.32$ depending on the data set.

Table 5.9 Training Data Summary for PNN Fusion with 1 Forced Look

Data	$P_T:P_F$	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ_{low}	θ_{up}	$\Delta\theta$
ord	1:1	1.00	1.00	1.00	0.04	0.02	0.00	1.00	0.91	0.01	0.19	0.18
aut	1:1	1.00	1.00	1.00	0.03	0.02	0.00	1.00	0.92	0.01	0.23	0.22
cor	1:1	1.00	1.00	1.00	0.03	0.02	0.00	1.00	0.75	0.01	0.28	0.26
ind	1:1	1.01	0.99	1.00	0.03	0.02	0.00	1.00	0.68	0.01	0.32	0.31

Each MVB fusion subplot in Figure 5.23 includes 1000 ROC curves generated by 10 different ROC thresholds for Sensor B, while ROC and rejection thresholds are held constant for Sensor A across 100 (10x10) values and Sensor B's rejection threshold is held constant at one of ten values. The optimal TPR is obtained without exercising a rejection window. When comparing the PNN to MVB fusion using Training data, slightly higher TPR rates are obtained by the near perfect ROC curves using PNN fusion. Also, significantly more variability is apparent for the MVB fusion producing 1000 ROC curves from 4 variable thresholds vs. the 100 ROC curves generated for PNN fusion using only 2 thresholds. The slightly higher TPR values obtained by the PNN fusion using Training data may be contributable to the fact that PNN fusion may reduce the number of "Non-declarations" by optimally selecting a small rejection window. On-the-other-hand, even when MVB fusion has no rejection window, as the case with the Test data, "Non-declaration" labels are still generated by the fusion system when the sensors disagree on the first look or when a majority vote is not obtained. Each of the "Non-declaration" labels then forces an additional look which reduces TPR.
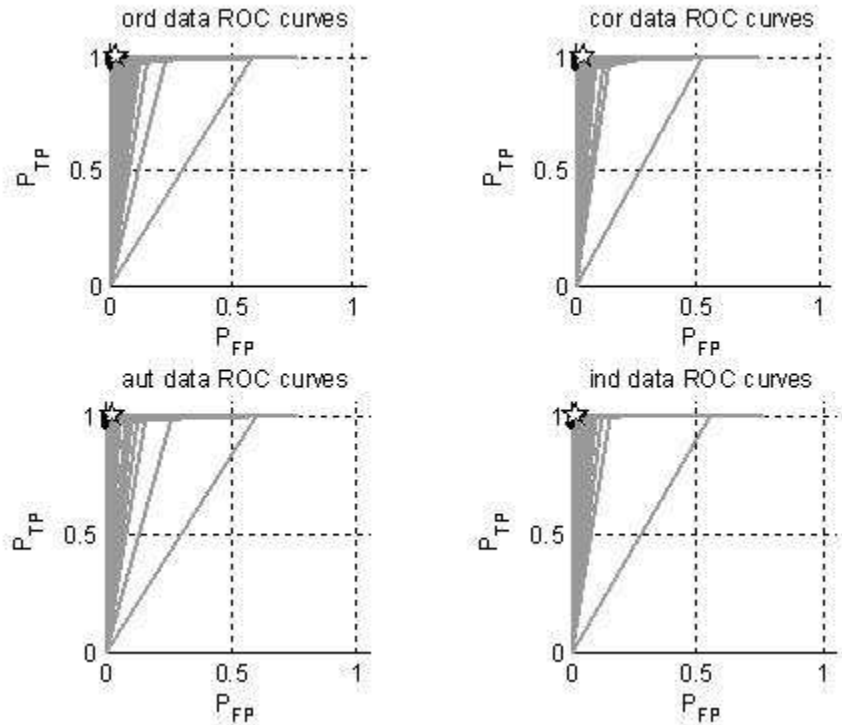
Figure 5.23 ROC Curves Generated from Training Data using MVB Fusion

Table 5.10 Training Data Summary for MVB Fusion with 1 Forced Look

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	1:1	1.06	0.94	1.00	0.03	0.02	0.00	0.99	0.68	0.11	0.11	0.11	0.11
aut	1:1	1.06	0.94	1.00	0.03	0.02	0.00	0.99	0.74	0.11	0.11	0.11	0.11
cor	1:1	1.07	0.93	1.00	0.03	0.02	0.00	1.00	0.76	0.11	0.11	0.11	0.11
ind	1:1	1.09	0.92	1.00	0.02	0.01	0.00	1.00	0.76	0.11	0.11	0.11	0.11

The next plots include ROC curves for each of correlation levels using the Test data set. The following figures include results using a minimum of 1 to 5 looks and for the prior probability of Hostiles to Friend (H:F) = 1:1 & 10:1. When generating these figures with various prior probabilities, only the overall Hostile to Friend class priors are changed. A constant ratio of TOD:OH is held constant at 1:4, and the use of the term Friend implies all five vehicles included in the Friend/Neutral class.

Figure 5.24 PNN Fusion Test Data ROC Curves with a Minimum of 1-look

Table 5.11 Test Data Summary for PNN Fusion with 1 Forced Look

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ_{low}	θ_{up}	$\Delta\theta$
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ord	10:1	1.15	0.870	1.000	1.000	0.019	0.028	0.917	0.008	0.000	0.191	0.191
aut	10:1	1.09	0.920	1.000	1.000	0.020	0.032	0.922	0.300	0.000	0.236	0.236
cor	10:1	1.16	0.860	1.000	1.000	0.017	0.048	0.924	0.013	0.000	0.355	0.355
ind	10:1	1.06	0.948	1.000	1.000	0.019	0.033	0.928	0.235	0.000	0.091	0.091

From the plots and tables above, the PNN fusion does not generalize well to the Test

data. This is indicated by no feasible solutions for equal priors. A target rich excursion

with H:F = 10:1 shows feasible regions of the ROC curves denoted by a dark area, with

the max TPR indicated by a star. All four optimum thresholds aggressively label all

objects as "Unknown" or as a "Hostile" as indicated by the values of θ in Table 5.11.

211

Figure 5.25 PNN Fusion Test Data ROC Curves with a Minimum of 2-looks

Table 5.12 Test Data Summary for PNN Fusion with 2 Forced Looks

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ_{low}	θ_{up}	$\Delta\theta$
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ord	10:1	2.10	0.476	1.000	1.000	0.011	0.009	0.906	0.932	0.000	0.009	0.009
aut	10:1	2.11	0.473	1.000	1.000	0.008	0.009	0.902	0.990	0.000	0.091	0.091
cor	10:1	2.07	0.484	1.000	1.000	0.008	0.018	0.914	0.792	0.000	0.009	0.009
ind	10:1	2.06	0.487	1.000	1.000	0.006	0.016	0.915	0.990	0.000	0.018	0.018

Again, with a minimum of 2-looks, PNN fusion does not generalize well to the Test data.

The Hostile target rich excursion shows feasible regions of the ROC curves denoted by

dark areas, and all optimal thresholds aggressively label all objects as "Unknown" or

"Hostile."

212

Figure 5.26 PNN Fusion Test Data ROC Curves with a Minimum of 3-looks

Table 5.13 Test Data Summary for PNN Fusion with 3 Forced Looks

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ_{low}	θ_{up}	$\Delta\theta$
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ord	10:1	3.11	0.321	1.000	1.000	0.007	0.011	0.884	0.990	0.000	0.009	0.009
aut	10:1	3.09	0.323	1.000	1.000	0.005	0.014	0.896	0.990	0.000	0.073	0.073
cor	10:1	3.04	0.329	1.000	1.000	0.003	0.000	0.905	0.990	0.000	0.036	0.036
ind	10:1	3.02	0.331	1.000	1.000	0.002	0.005	0.911	0.990	0.000	0.009	0.009

Similar results are obtained for PNN fusion with a minimum of 3-looks. The target rich excursion shows feasible regions of the ROC concentrated at the upper NW corner "knees" in all four ROC curves along with the optimal TPR thresholds denoted by a star. Of interest is that 99% of all thresholds assessed for the hostile target rich environment occur at those dark areas and at the optimal TPR indicated by the star in the upper NE plot corners.

213

Figure 5.27 PNN Fusion Test Data ROC Curves with a Minimum of 4-looks

Table 5.14 Test Data Summary for PNN Fusion with 4 Forced Looks

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ_{low}	θ_{up}	$\Delta\theta$
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ord	10:1	4.06	0.246	1.000	1.000	0.004	0.014	0.877	0.990	0.000	0.018	0.018
aut	10:1	4.05	0.247	1.000	1.000	0.003	0.009	0.889	0.990	0.000	0.073	0.073
cor	10:1	4.02	0.249	1.000	1.000	0.001	0.000	0.901	0.990	0.000	0.436	0.436
ind	10:1	4.02	0.249	1.000	1.000	0.000	0.000	0.903	0.990	0.000	0.009	0.009

PNN fusion with a minimum of 4-looks indicates very good looking ROC curves, yet they remain infeasible for the case of equal priors. The target rich excursion shows very similar results to that using 3-looks, where very aggressive thresholds yield the maximum TPR, with 100% TP declaration and 100% FP declarations for all objects declared.
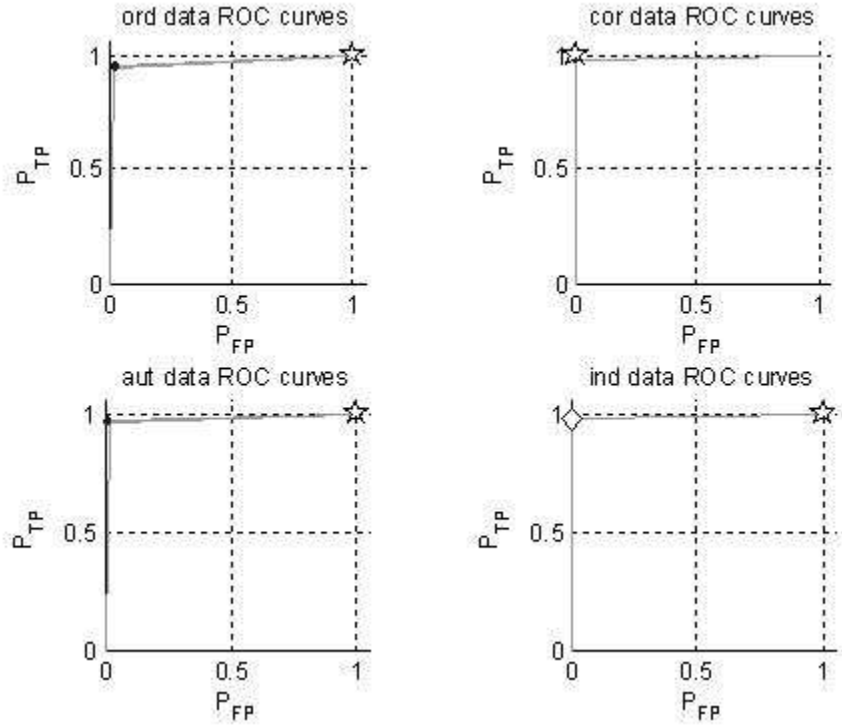
Figure 5.28 PNN Fusion Test Data ROC Curves with a Minimum of 5-looks

Table 5.15 Test Data Summary for PNN Fusion with 5 Forced Looks

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ_{low}	θ_{up}	$\Delta\theta$
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	5.10	0.196	0.980	0.003	0.018	0.000	1.000	0.980	0.099	0.999	0.900
ord	10:1	5.00	0.200	1.000	1.000	0.002	0.002	0.863	0.990	0.000	0.900	0.900
aut	10:1	5.00	0.200	1.000	1.000	0.001	0.007	0.881	0.990	0.000	0.336	0.336
cor	10:1	5.00	0.200	1.000	0.000	0.000	0.000	0.889	0.990	0.000	0.900	0.900
ind	10:1	5.00	0.200	1.000	1.000	0.000	0.000	0.891	0.990	0.000	0.900	0.900

PNN fusion with a minimum of 5-looks finally yields a feasible solution for the case of independent (ind) data. In fact, 98% of the thresholds assessed using ind data and equal priors are feasible and occur at the knee in the ROC curve located at the diamond shape. Each of the evaluations using the target rich priors indicates the best TPR of 0.20 was obtained, with 5 looks used to make every TP Hostile class declaration.

The following figures will now be used to present and assess the Majority Vote Boolean fusion rule across 1-5 forced looks using Test data.
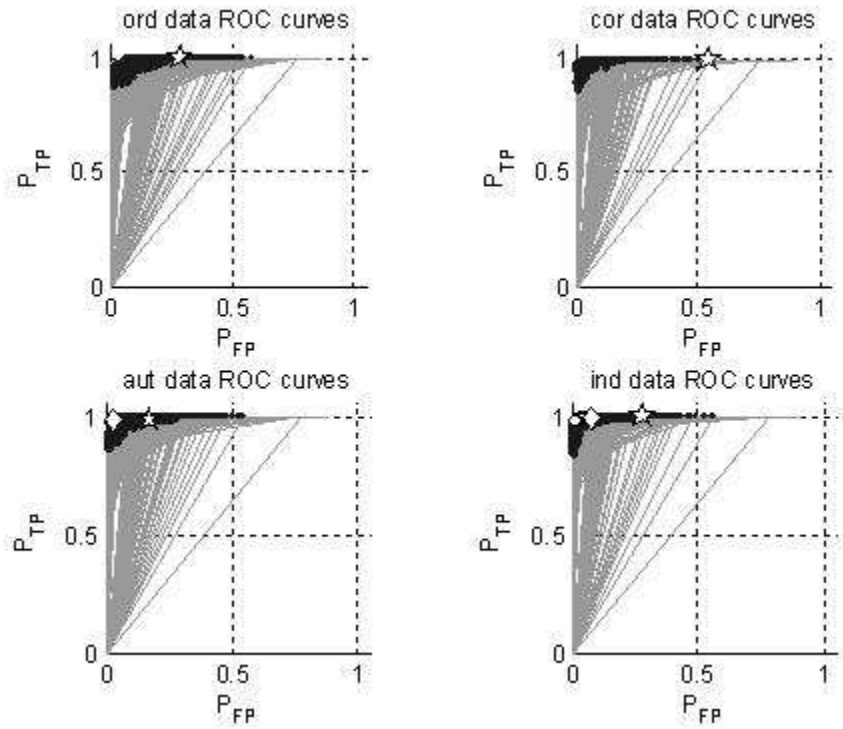
Figure 5.29 MVB Fusion Test Data ROC Curves with a Minimum of 1-look

Table 5.16 Test Data Summary for MVB Fusion with 1 Forced Look

Data	$P_T:P_F$	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	2.32	0.431	0.988	0.034	0.020	0.008	0.828	0.001	0.033	0.733	0.011	0.911
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	1.77	0.566	0.997	0.078	0.019	0.008	0.710	0.013	0.133	0.533	0.000	0.600
ord	10:1	1.37	0.731	1.000	0.281	0.017	0.012	0.964	0.610	0.222	0.222	0.000	0.100
aut	10:1	1.38	0.726	0.987	0.173	0.015	0.011	0.977	0.725	0.000	0.100	0.111	0.111
cor	10:1	1.32	0.760	0.995	0.548	0.018	0.012	0.938	0.709	0.000	0.100	0.000	0.100
ind	10:1	1.33	0.750	1.000	0.282	0.017	0.012	0.968	0.770	0.111	0.111	0.000	0.100

The plots for MVB fusion with a minimum of 1-look now indicate feasibility using equal priors for the case of autocorrelated (aut) and independent (ind) data. Feasible ROC points are indicated by white circles and a diamond is at the optimal TPR. Significantly more feasible points are found for the target rich excursion with feasible dark areas and a

star used to denote the optimal TPR. For a minimum of 1 look, the MVB fusion achieves

feasibility with equal priors for aut and ind data. Yet, the PNN fusion achieves a higher

TPR for the target rich environment. As in the training data, significantly more

variability is observed for 1000 MVB ROC curves vs. the 100 PNN ROC curves in

Figures 5.24-5.28.
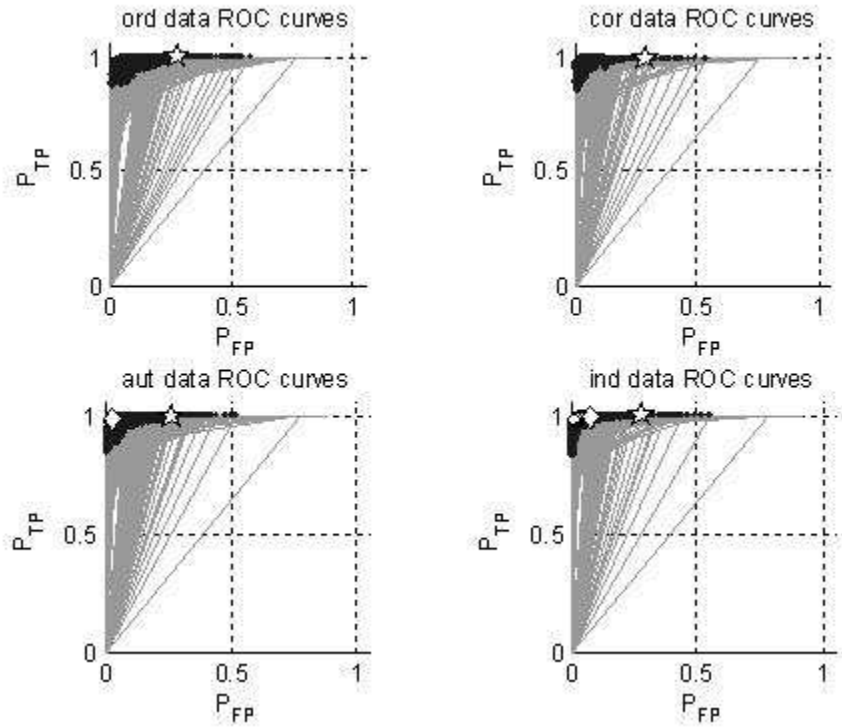
Figure 5.30 MVB Fusion Test Data ROC Curves with a Minimum of 2-looks

Table 5.17 Test Data Summary for MVB Fusion with 2 Forced Looks

Data	$P_T{:}P_F$	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aut	1:1	2.73	0.366	0.988	0.029	0.019	0.010	0.851	0.002	0.044	0.644	0.011	0.911
cor	1:1	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ind	1:1	2.25	0.445	0.997	0.082	0.019	0.007	0.700	0.016	0.078	0.378	0.000	0.600
ord	10:1	2.11	0.474	1.000	0.279	0.017	0.009	0.963	0.648	0.222	0.222	0.000	0.100
aut	10:1	2.11	0.473	0.997	0.263	0.016	0.014	0.964	0.751	0.222	0.222	0.000	0.100
cor	10:1	2.07	0.483	0.995	0.289	0.018	0.002	0.963	0.742	0.100	0.200	0.000	0.100
ind	10:1	2.06	0.485	1.000	0.282	0.017	0.009	0.968	0.775	0.111	0.111	0.000	0.100

The MVB fusion results for 2 forced looks appear similar to that of 1 look. The ordered (ord) and co-registered (cor) data sets remain infeasible, but now the max TPR associated with both PNN and MVB appear to be equivalent, with differences less than 0.003.
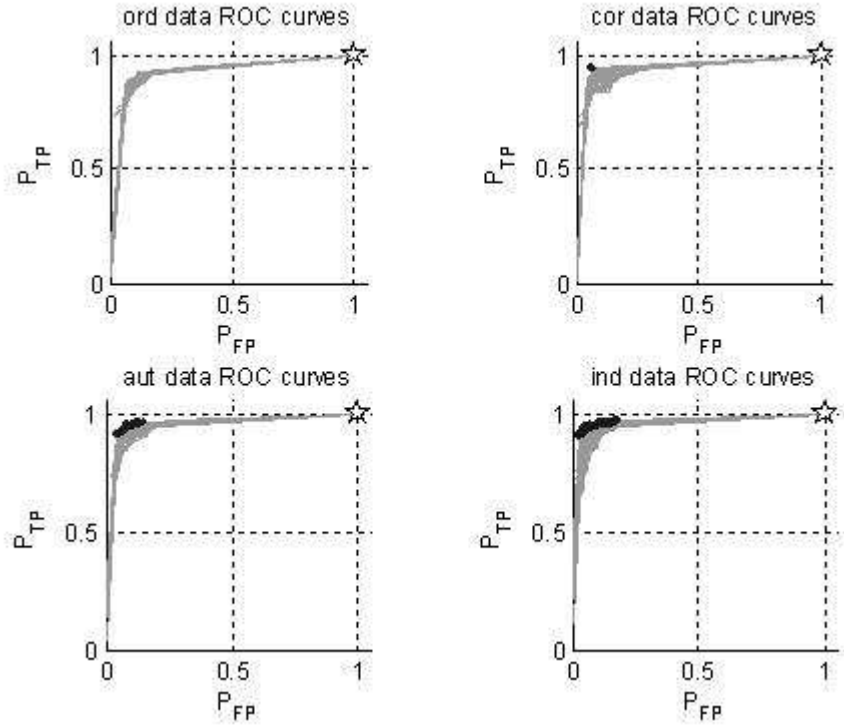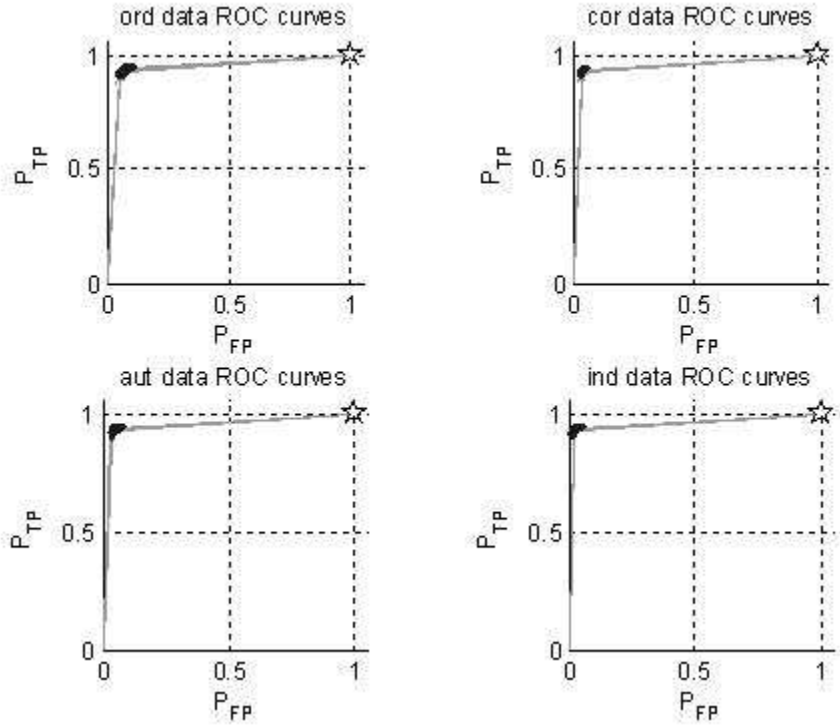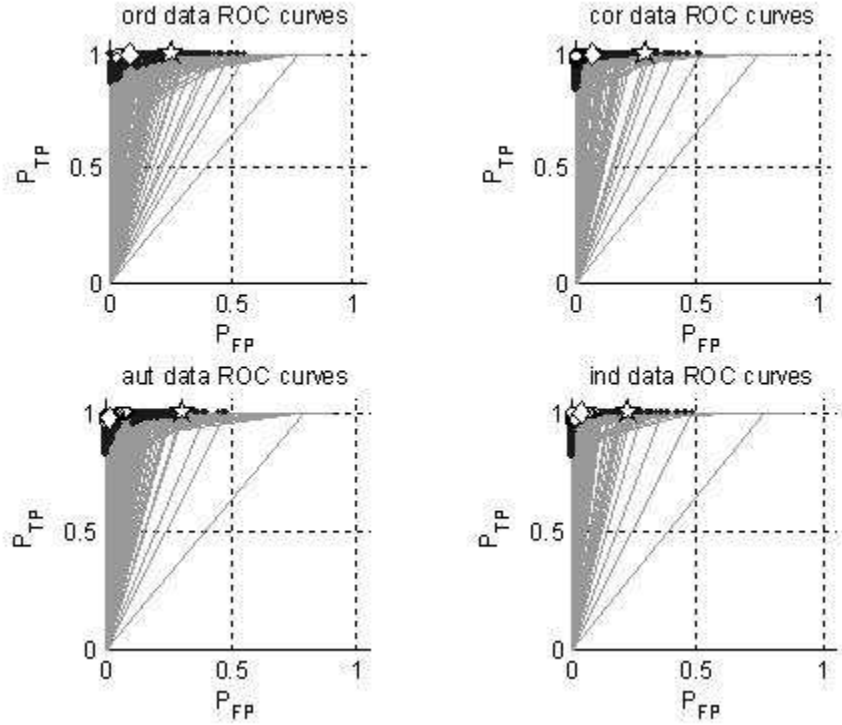

Figure 5.31 MVB Fusion Test Data ROC Curves with a Minimum of 3-looks

Table 5.18 Test Data Summary for MVB Fusion with 3 Forced Looks

Data	P_T:P_F	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	1:1	3.24	0.308	1.000	0.081	0.020	0.008	0.713	0.001	0.389	0.889	0.000	0.100
aut	1:1	3.15	0.317	0.980	0.015	0.020	0.005	0.980	0.065	0.222	0.722	0.111	0.111
cor	1:1	3.03	0.330	1.000	0.071	0.020	0.001	0.780	0.267	0.356	0.556	0.000	0.100
ind	1:1	3.01	0.332	1.000	0.041	0.020	0.001	0.994	0.389	0.111	0.111	0.056	0.556
ord	10:1	3.04	0.329	1.000	0.251	0.014	0.012	0.958	0.696	0.222	0.222	0.000	0.100
aut	10:1	3.05	0.328	1.000	0.306	0.018	0.009	0.960	0.768	0.111	0.111	0.000	0.100
cor	10:1	3.01	0.332	1.000	0.408	0.007	0.002	0.926	0.794	0.000	0.200	0.000	0.100
ind	10:1	3.01	0.333	1.000	0.081	0.005	0.002	0.970	0.796	0.444	0.444	0.000	0.100

MVB fusion results for 3 forced looks now begin to show more feasible regions indicated by white circles for equal priors and dark areas for the target rich environment. All four test data sets have feasible operating thresholds. Assessment of the optimal TPR sensor

thresholds reveal each sensor has been tuned to perform a different function. For

example, the last line in Table 5.18 shows Sensor A declares only "FN" if $< \theta_{low} = \theta_{up}$

o.w. "H," and Sensor B, with $\theta_{low} = 0$, only declares "ND" if $< \theta_{up}$ or "H" if $> \theta_{up}$.
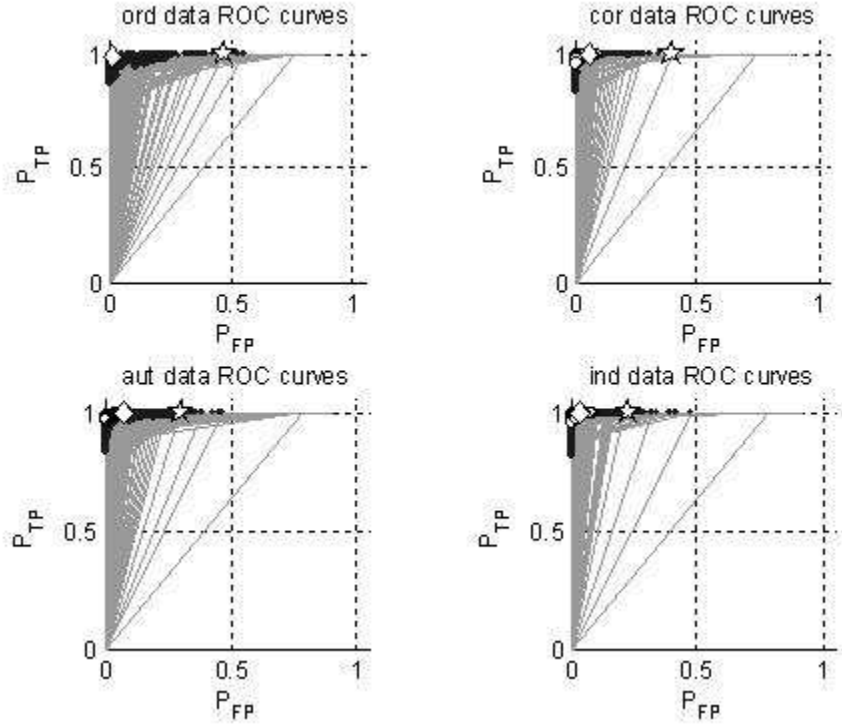
Figure 5.32 MVB Fusion Test Data ROC Curves with a Minimum of 4-looks

Table 5.19 Test Data Summary for MVB Fusion with 4 Forced Looks

Data	$P_T{:}P_F$	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	1:1	4.19	0.238	0.989	0.009	0.018	0.005	0.716	0.000	0.711	0.911	0.000	0.300
aut	1:1	4.04	0.247	1.000	0.078	0.020	0.004	0.744	0.074	0.356	0.556	0.000	0.200
cor	1:1	4.00	0.250	1.000	0.053	0.016	0.001	0.805	0.553	0.556	0.556	0.000	0.100
ind	1:1	4.00	0.250	1.000	0.048	0.019	0.000	0.894	0.470	0.000	0.400	0.111	0.111

ord	10:1	4.04	0.248	0.997	0.474	0.010	0.009	0.913	0.707	0.000	0.200	0.000	0.100
aut	10:1	4.02	0.249	1.000	0.137	0.007	0.009	0.953	0.767	0.333	0.333	0.000	0.100
cor	10:1	4.00	0.250	1.000	0.216	0.003	0.002	0.921	0.793	0.000	0.300	0.000	0.100
ind	10:1	4.00	0.250	1.000	0.226	0.016	0.000	0.980	0.796	0.000	0.000	0.111	0.611

With 4 forced looks, MVB fusion obtains close to the maximum obtainable 0.25 TPR as

calculated by 1 TP per 4 forced looks. In general, a higher percentage of assessed MVB

thresholds are feasible. Again, inspection of the sensor thresholds for the optimal TPR,

reveals each sensor has been tuned to perform a different function, with some sensors

capable of declaring three-class labels, others tuned to two labels and one case where

only "H" labels are declared (last line of Table 5.19, ind data $\theta^A_{low} = \theta^A_{up} = 0$).
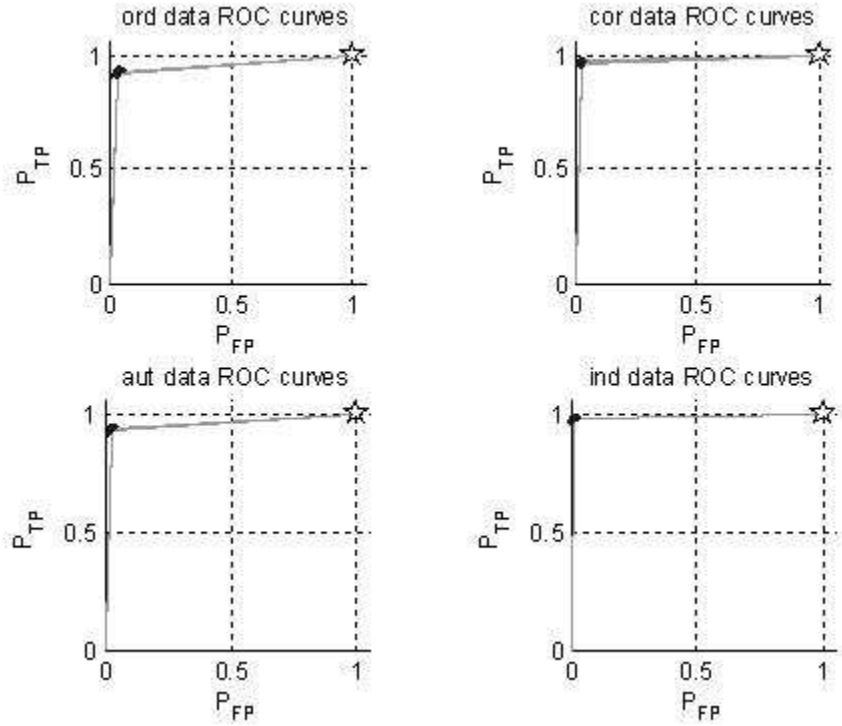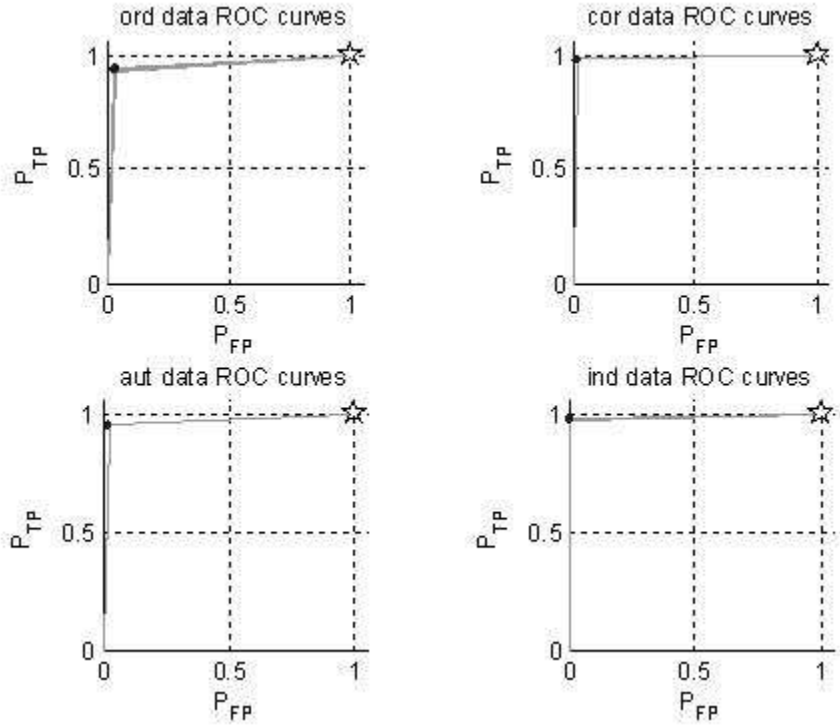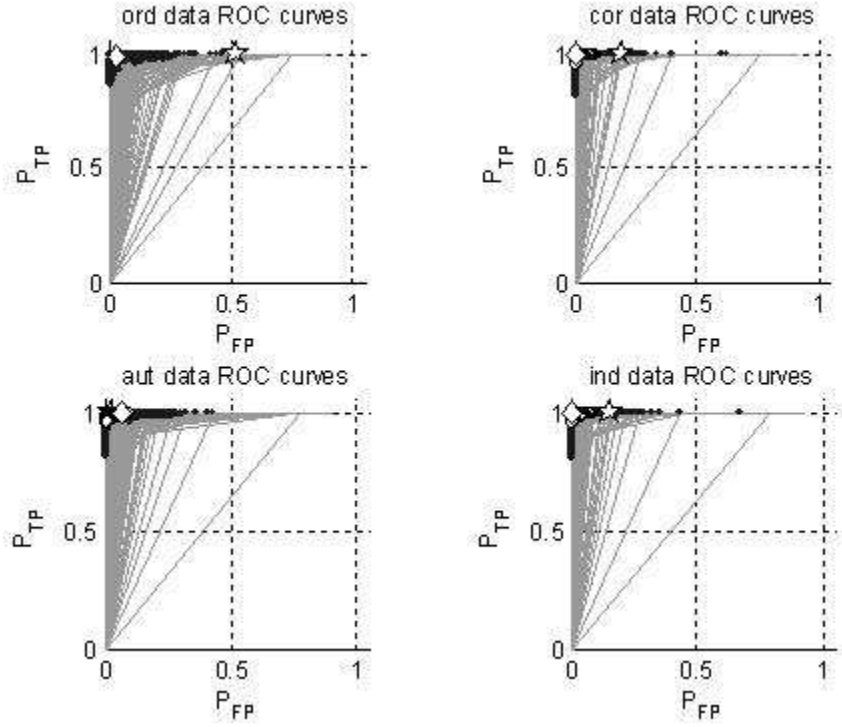

Figure 5.33 MVB Fusion Test Data ROC Curves with a Minimum of 5-looks

Table 5.20 Test Data Summary for MVB Fusion with 5 Forced Looks

Data	$P_T{:}P_F$	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	1:1	5.03	0.199	0.995	0.024	0.019	0.004	0.719	0.000	0.622	0.922	0.000	0.100
aut	1:1	5.00	0.200	1.000	0.037	0.011	0.003	0.786	0.165	0.000	0.500	0.111	0.111
cor	1:1	5.00	0.200	1.000	0.000	0.000	0.000	0.724	0.640	0.044	0.944	0.089	0.989
ind	1:1	5.00	0.200	1.000	0.000	0.000	0.000	0.738	0.498	0.078	0.978	0.089	0.989

Data	$P_T{:}P_F$	L_TP	maxTPR	TP	FP	E_{CR}	E_{NC}	P_{Dec}	%Feas	θ^A_{low}	θ^A_{up}	θ^B_{low}	θ^B_{up}
ord	10:1	5.01	0.199	0.997	0.519	0.010	0.005	0.907	0.718	0.000	0.100	0.000	0.200
aut	10:1	5.00	0.200	1.000	0.025	0.001	0.002	0.711	0.768	0.089	0.989	0.000	0.200
cor	10:1	5.00	0.200	1.000	0.186	0.014	0.000	0.727	0.782	0.900	1.000	0.000	0.000
ind	10:1	5.00	0.200	1.000	0.154	0.011	0.000	0.733	0.790	0.900	1.000	0.000	0.000

With 5 forced looks, MVB fusion is feasible for all data sets; yet, the percentage of

feasible thresholds is very limited (1 of 10,000 points is feasible) for equal priors

assessed using the naturally ordered (ord) data. At the optimal TPR, all two sensor combinations are tuned to perform slightly different label functions. With the optimal TPR close to 0.20 for all data sets, differences between use of MVB or PNN fusion in the Hostile target rich environment appear small, where both fusion methods obtains the same optimal TPR solution.

The next two tables include Test data summary information sorted by max TPR across all minimum looks and all sensor data correlation structures. From Table 5.21, with equal prior probabilities of Hostiles and Friends, the maximum TPR was either obtained by MVB fusion or was found to be equivalent when comparing each fusion system using the same data correlation structure and the same minimum number of looks. For the only point where PNN fusion was feasible, it provided a comparable TPR to MVB fusion (0.196 vs. 0.200), which is highlighted in gray. The bottom four rows with gray indicate the four minimum look/data correlation combinations which did not yield a feasible solution for either MVB or PNN fusion. These points are associated with the naturally ordered data with the highest level of correlation and the co-registered data with the naturally occurring correlation across sensors at each look. In general, the independent and autocorrelated (aut) data tend to yield higher TPR results, while the naturally ordered (ord) data tends to provide the lowest TPR. In addition, while the maximum TPR is obtained with the minimum of 1 forced look, by requiring additional looks, the MVB fusion gains feasibility across all four data sets. Thus, robustness to obtain a feasible combat ID system across the generated correlation structures is obtained by incorporating at least 3-forced looks.

Table 5.21 Sorted max TPR Summary for All Test Data and Equal Priors

minimum Looks	data correlation	optimal Fusion	max TPR	PNN max TPR	MVB max TPR	TPR diff MVB-PNN	PNN % Feas	MVB % Feas
1	ind	**MVB**	0.566	0.000	0.566	0.566	0.000	0.013
2	ind	**MVB**	0.445	0.000	0.445	0.445	0.000	0.016
1	aut	**MVB**	0.431	0.000	0.431	0.431	0.000	0.001
2	aut	**MVB**	0.366	0.000	0.366	0.366	0.000	0.002
3	ind	**MVB**	0.332	0.000	0.332	0.332	0.000	0.389
3	cor	**MVB**	0.330	0.000	0.330	0.330	0.000	0.267
3	aut	**MVB**	0.317	0.000	0.317	0.317	0.000	0.065
3	ord	**MVB**	0.308	0.000	0.308	0.308	0.000	0.001
4	cor	**MVB**	0.250	0.000	0.250	0.250	0.000	0.553
4	ind	**MVB**	0.250	0.000	0.250	0.250	0.000	0.470
4	aut	**MVB**	0.247	0.000	0.247	0.247	0.000	0.074
4	ord	**MVB**	0.239	0.000	0.239	0.239	0.000	0.000
5	cor	**MVB**	0.200	0.000	0.200	0.200	0.000	0.640
5	ind	MVB	0.200	0.196	0.200	**0.004**	0.980	0.498
5	aut	**MVB**	0.200	0.000	0.200	0.200	0.000	0.165
5	ord	**MVB**	0.199	0.000	0.199	0.199	0.000	0.0001
1	ord	**equal**	0.000	0.000	0.000	0.000	**0.000**	0.000
2	ord	**equal**	0.000	0.000	0.000	0.000	**0.000**	0.000
1	cor	**equal**	0.000	0.000	0.000	0.000	**0.000**	0.000
2	cor	**equal**	0.000	0.000	0.000	0.000	**0.000**	0.000

The next table presents the same information summary for the case of a Hostile target rich environment. From this table, all Fusion models were feasible and in the case of 5-looks using autocorrelated data both fusion methods obtained the maximum achievable TPR of 0.20. Further, for all 16 comparisons with 2 or more minimum looks the maximum TPR achieved was less than 3% different between the two fusion methods. Thus, while MVB fusion is optimal for 8 of the 20 comparisons, PNN fusion is optimal for the 4 cases with 1 minimum look, where a significant difference is found between PNN and MVB fusion

Since comparison of the fusion methods across two different prior probabilities provided some evidence of different fusion preferences based on an assumed prior probability of Hostiles and Friends, further excursions should be performed. By just

varying H:F priors, MVB fusion appears preferred for an environment with equal priors. Requiring a forced number of looks appears to significantly aid either fusion algorithm to meet the warfighter constraints. The Hostile target rich environment then showed optimal performance of the PNN fusion if using only 1-forced look, while the fusion algorithms appeared equivalent if requiring more than 1-look prior to labeling a target. To gain additional insights of fusion model preferences and differences, sensitivity analysis across different priors and other parameters is performed in the next section.

Table 5.22 Sorted max TPR Summary for All Test Data and $P_T:P_F = 10:1$

minimum Looks	data correlation	optimal Fusion	max TPR	PNN max TPR	MVB max TPR	TPR diff MVB-PNN	PNN % Feas	MVB % Feas
1	ind	**PNN**	0.948	0.948	0.751	-0.197	0.235	0.770
1	aut	**PNN**	0.920	0.920	0.726	-0.194	0.300	0.725
1	ord	**PNN**	0.870	0.870	0.731	-0.138	0.008	0.610
1	cor	**PNN**	0.860	0.860	0.760	-0.101	0.013	0.709
2	ind	PNN	0.487	0.487	0.485	**-0.001**	0.990	0.775
2	cor	PNN	0.484	0.484	0.483	**-0.001**	0.792	0.742
2	ord	PNN	0.476	0.476	0.474	**-0.002**	0.932	0.648
2	aut	**equal**	0.473	0.473	0.473	**0.000**	0.990	0.751
3	ind	MVB	0.333	0.331	0.333	**0.002**	0.990	0.796
3	cor	MVB	0.332	0.329	0.332	**0.003**	0.990	0.794
3	ord	MVB	0.329	0.321	0.329	**0.007**	0.990	0.696
3	aut	MVB	0.328	0.323	0.328	**0.005**	0.990	0.768
4	cor	MVB	0.250	0.249	0.250	**0.001**	0.990	0.793
4	ind	MVB	0.250	0.249	0.250	**0.001**	0.990	0.796
4	aut	MVB	0.249	0.247	0.249	**0.002**	0.990	0.767
4	ord	MVB	0.248	0.246	0.248	**0.001**	0.990	0.707
5	aut	**equal**	0.200	0.200	0.200	**0.000**	0.990	0.768
5	cor	**equal**	0.200	0.200	0.200	**0.000**	0.990	0.782
5	ind	**equal**	0.200	0.200	0.200	**0.000**	0.990	0.790
5	ord	PNN	0.200	0.200	0.200	**-0.001**	0.990	0.718

5.7 Sensitivity Analysis of ATR Fusion Systems

A sensitivity analysis for assessment of the fusion methods is presented next. The focus of this analysis is perturbation of three variables that appear to have the most

influence on the operating characteristics of the ID systems, including ATR system feasibility and the maximum feasible *TPR*. Classical sensitivity analysis using dual variables, etc. was not performed, since the assessments of the fusion systems were made exhaustively across all thresholds and across all desired sensitivity variables. This exhaustive search facilitated determination of the percentage of feasible thresholds along with other summary statistics measured across all threshold gridpoints evaluated. Sensitivity analysis performed in this manner is consistent with advice given by Brown (2004) for the modeling of military applications, where he states,

> Classical sensitivity analysis is bunk…Just plan on solving a lot of model excursions…In our world, it's more important to seek "scenario- (i.e., warplan-) robust" solutions than to worry about individual parameter changes.

Brown's theme suggests mathematical optimization for military applications needs to be robust with ever changing operational needs and models need to consider alternate future scenarios from which an overall best solution may be synthesized.

As shown in the previous section, the ratio of Hostiles to Friends can significantly affect the optimal tuning of the thresholds associated with each system. This tuning, performed by the mixed variable optimization, allows for identification of different feasible operating points on the same ROC curves associated with a single fusion system and data combination. To further assess the impact of various priors, all systems will be assessed under environments of sparse Hostile targets through dense Hostile targets. In addition, to assess the sensitivity of the critical error and declaration rate constraints, these right hand side values in the optimization framework will be varied between more and less restrictive values than those used in the previous section. Since all fusion systems assessed yielded 93-100% feasibility with respect to the non-critical error

constraint, sensitivity analysis of this constraint is not performed. Initial comparison of the critical and non-critical error constraints, shows the critical error constraint as a more restrictive, binding constraint. This is reasonable since these constraints both require high classification accuracy for a 2-class problem; yet, the desired critical error is much lower with desired rate of 0.02 vs. a non-critical classification error rate of 0.05. Further, for this experiment the non-critical error may be associated with a slightly easier classification effort in which a large SCUD vehicle appears significantly different from the other four Hostile vehicles when fusing both sensors. This includes differentiation from the SMERCH, which is the closest Hostile confuser, as can be seen by the TPM assessments using Sensor B presented in Table 5.4b.

A full factorial experimental design was used to assess the fusion performance across 9 levels of prior probabilities, 4 levels of critical error constraint values (Π_1) and 3 levels of declaration constraint values (Π_3). A total of 108 designed levels were assessed for each of the 80 fusion model and data combinations. The 80 fusion model data combinations were identified by the use of PNN or MVB fusion with 1-5 forced looks assessed on Training and Test sets composed of the 4 different sensor correlations. Performance data associated with each of the 80 fusion model/data combinations was generated first, and included the confusion matrix information associated with each of the 10,000 thresholds evaluated for each fusion model. Sensitivity analysis was then performed by assessing each of the 10,000 thresholds for each fusion model/data combination (80), for each of the designed sensitivity analysis levels (108).

Table 5.23 Sensitivity Analysis Variables and Levels

Prior Probabilities (H:F)   Hostile Target Sparse to Target Rich
- Initial assessments performed at 1:1 and 10:1
- Non-uniform sampling at 1:20, 1:10, 1:4, 1:2, 1:1, 2:1, 4:1, 10:1, 20:1
- 9 Levels considered

Critical Error Constraint
- Initial assessment performed with $\Pi_1 = 0.02$
- Let $\Pi_1 = 0.01, 0.02, 0.03, 0.04$
- 4 Levels considered

Declaration Constraint
- Initial assessmenst performed with $\Pi_3 = 0.70$
- Let $\Pi_3 = 0.60, 0.70, 0.80$
- 3 Levels considered

Assessment of all 80 fusion model/data combinations for each one of the 108 of the designed levels required approximately 35 minutes using a dedicated 2.66 GHz dual processor desktop with 2.0 GB of RAM. This assessment included the vertical analysis required to determine output label probabilities for each level of prior probability and the subsequent assessment of feasibility across constraints for each of the 800,000 (80 models x 10,000 threshold evaluations) fusion model performance values. This evaluation across all 108 designed levels was accomplished in approximately 65 hours using the same dedicated computer. Output files consisted of 10,000 rows for each threshold setting and either 17 of 19 columns. This size difference was generated between saving the 2 PNN fusion thresholds, $\boldsymbol{\theta}^{PNN} = (\theta_{low}, \theta_{up})^T$, or 4 MVB fusion thresholds, $\boldsymbol{\theta}^{MVB} = (\theta^{SA}_{low}, \theta^{SA}_{up}, \theta^{SB}_{low}, \theta^{SB}_{up})^T$. Table 5.24 presents the data saved for

each of the 108x80 = 8640 output data files. For each of the 8640 sensitivity analysis

computations summary information was also collected.

Table 5.24 Summary of Data Collected by Column for All Sensitivity Analysis

Identification information, provided in data file name:	
Fusion ID	PNN or PCB
Data correlation	ord, cor, aut, ind
Train or Test	TR or TE
Hprior	Used to calculate prior probability of Hostile and Friendly targets
Fprior	Used to calculate prior probability of Hostile and Friendly targets
E_CR	Maximum Critical error rate (RHS of constraint)
minDEC	Minimum Declaration rate (RHS of constraint)
min Look	Minimum Looks to take prior to making a declaration

Performance information based on each threshold space evaluated :	
1. **TPR**	TPR for given thresholds
2. **H_IDR**	Hostile ID rate = TP/looks used to assess Hostile & Friendly targets
3. **TP**	TP associated with thresholds
4. **FN**	FN associated with thresholds
5. **UT**	Undeclared Targets associated with thresholds
6. **FP**	FP associated with thresholds
7. **TN**	TN associated with thresholds
8. **UF**	Undeclared Friends associated with thresholds
9. **Ec**	Critical Error associated with thresholds
10. **En**	Non-critical error associated with thresholds
11. **PRrej**	Percentage of objects Not-Declared associated with thresholds

Threshold space used for assessment:	
12. **theta 1**	Lower threshold for rejection window (Sensor A or PNN)
13. **theta 2**	Upper threshold for rejection window (Sensor A or PNN)
14. **theta 3**	Lower threshold for rejection window (Boolean Sensor B)
15. **theta 4**	Upper threshold for rejection window (Boolean Sensor B)
16. **theta 5**	TOD threshold for MVB fusion (column 14 for PNN fusion)

Feasibility:	
17. **EC Feasible**	1 if critical error is feasible, 0 o.w. (column 15 for PNN fusion)
18. **NC Feasible**	1 if non-critical error is feasible, 0 o.w. (column 16 for PNN fusion)
19. **ND Feasible**	1 if percent rejection is feasible, 0 o.w. (column 17 for PNN fusion)

Table 5.25 provides an overview of the summary data. The summary information includes 29 or 31 columns depending of the number of variable thresholds associated with PNN or MVB fusion. Along with fusion method/data information, the levels of the three sensitivity analysis parameters are included for identification purposes. Statistics associated with the maximum TPR are included as columns 7-17. Columns 18-22 include average performance values across all feasible thresholds. Columns 23-26 indicate overall feasibility and feasibility by each of the three operational constraints. Finally, the thresholds (theta 1-5) associated with the optimal TPR are included.

Table 5.25 Summary Information Collected by Column for Each Designed Run

Identification information:

1.	**Fusion ID**	PNN or PCB, correlation structure (ord, cor, aut, ind), Train or Test (2 Fusion algorithms) x (4 data structures) x (Train or Test) = 16 IDs
2.	**Hprior**	Used to calculate prior probability of Hostile and Friendly targets
3.	**Fprior**	Used to calculate prior probability of Hostile and Friendly targets
4.	**E_CR**	Maximum Critical error rate (RHS of constraint)
5.	**min DEC**	Minimum Declaration rate (RHS of constraint)
6.	**min Look**	Minimum Looks to take prior to making a declaration

Performance information based on gridpoint with maximum TPR :

7.	**L_TP**	Mean Looks required to obtain a TP, given assessing a Hostile
8.	**maxTPR**	Maximum TPR from gridspace
9.	**L_ID**	Mean Looks required to obtain a TP, while assessing any object
10.	**maxIDR**	Max hostile ID rate associated with maxTPR (changes with priors)
11.	**optTP**	TP associated with maximum TPR
12.	**optFP**	FP associated with maximum TPR
13.	**optCR**	Critical Error associated with maximum TPR
14.	**optNC**	Non-critical error associated with maximum TPR
15.	**optDT**	Declaration rate associated with maximum TPR
16.	**optUH**	% of Undeclared Hostiles associated with maximum TPR
17.	**optUF**	% of Undeclared Friendlies associated with maximum TPR

Performance information based on ALL feasible gridpoints:

18.	**meanTPR**	Mean TPR for all feasible points
19.	**meanCR**	Mean Critical Error for all feasible points
20.	**meanDT**	Mean percentage of objects declared for all feasible points
21.	**meanUH**	Mean percentage of Hostile targets declared "Unknown"
22.	**meanUF**	Mean percentage of Friendly targets declared "Unknown"

Feasibility information for all gridpoints:

23.	**%Feas**	Percentage compliant to all constraints
24.	**%FeasCR**	Percentage compliant with Critical Error constraint
25.	**%FeasNC**	Percentage compliant with Non-critical error constraint
26.	**%FeasND**	Percentage compliant with Non-declaration constraint

Gridpoints associated with maximum TPR :

27.	**theta 1**	Lower threshold for rejection window (Sensor A or PNN)
28.	**theta 2**	Upper threshold for rejection window (Sensor A or PNN)
29.	**theta 3**	Lower threshold for rejection window (Boolean Sensor B)
30.	**theta 4**	Upper threshold for rejection window (Boolean Sensor B)
31.	**theta 5**	TOD threshold (Column 29 for PNN fusion)

Reviewing Tables 5.26 and 5.27 provides a focus to the sensitivity analysis. Two primary goals will be undertaken. The first goal will seek to determine where each fusion method may be preferred. For example, the sensitivity analysis variables and levels need to be identified associated with the 1-look fusion where PNN fusion outperforms MVB fusion. Also, since TPR is estimated by evaluation of the specific fusion algorithm given limited data sets, assessment should also include determining where the fusion performance is relatively equivalent. The second goal will attempt to characterize the infeasibility space associated with each of the two fusion models and compare these conditions.

Table 5.26 Percentage of Optimal Fusion Method across All Sensitivity Analysis Levels by Test Data Correlation Structure and Minimum Number of Looks

PNN Fusion Optimal

	1-look	2-looks	3-looks	4-looks	5-looks
ord	25.0%	21.3%	0.0%	2.8%	19.4%
aut	28.7%	18.5%	0.0%	0.0%	8.3%
cor	33.3%	27.8%	0.0%	0.0%	0.0%
ind	37.0%	23.1%	0.0%	0.0%	0.0%

PNN Fusion Optimal by > 5%

	1-look	2-looks	3-looks	4-looks	5-looks
ord	25.0%	0.0%	0.0%	0.0%	0.0%
aut	28.7%	0.0%	0.0%	0.0%	0.0%
cor	30.6%	0.9%	0.0%	0.0%	0.0%
ind	37.0%	0.9%	0.0%	0.0%	0.0%

MVB Fusion Optimal

	1-look	2-looks	3-looks	4-looks	5-looks
ord	19.4%	27.8%	60.2%	56.5%	36.1%
aut	48.1%	60.2%	100.0%	97.2%	64.8%
cor	13.9%	23.1%	96.3%	100.0%	80.6%
ind	39.8%	53.7%	100.0%	100.0%	81.5%

MVB Fusion Optimal by > 5%

	1-look	2-looks	3-looks	4-looks	5-looks
ord	19.4%	16.7%	30.6%	25.0%	23.1%
aut	48.1%	46.3%	66.7%	60.2%	55.6%
cor	13.9%	17.6%	57.4%	38.0%	38.0%
ind	39.8%	39.8%	46.3%	50.0%	32.4%

Equivalent Fusion

	1-look	2-looks	3-looks	4-looks	5-looks
ord	55.6%	50.9%	39.8%	40.7%	44.4%
aut	23.1%	21.3%	0.0%	2.8%	26.9%
cor	52.8%	49.1%	3.7%	0.0%	19.4%
ind	23.1%	23.1%	0.0%	0.0%	18.5%

Fusion Equivalent within 5%

	1-look	2-looks	3-looks	4-looks	5-looks
ord	55.6%	83.3%	69.4%	75.0%	76.9%
aut	23.1%	53.7%	33.3%	39.8%	44.4%
cor	55.6%	81.5%	42.6%	62.0%	62.0%
ind	23.1%	59.3%	53.7%	50.0%	67.6%

Table 5.27 Percentage of Feasible PNN and MVB Fusion across All Sensitivity Analysis
Levels by Test Data Correlation Structure and Minimum Number of Looks

	PNN Fusion Feasibility min Looks					MVB Fusion Feasibility min Looks				
	1-look	2-looks	3-looks	4-looks	5-looks	1-look	2-looks	3-looks	4-looks	5-looks
ord	25.0%	32.4%	30.6%	34.3%	35.2%	44.4%	49.1%	60.2%	59.3%	58.3%
aut	28.7%	32.4%	33.3%	37.0%	43.5%	76.9%	87.0%	100.0%	97.2%	98.2%
cor	33.3%	33.3%	38.9%	62.0%	62.0%	47.2%	68.5%	100.0%	100.0%	100.0%
ind	37.0%	37.0%	53.7%	50.0%	67.6%	86.1%	94.4%	100.0%	100.0%	100.0%

Plots were generated to show the preferred fusion method based on TPM using

data across the minimum number of looks and across all three variables under sensitivity

analysis investigation. Each plot shows the performance between PNN and MVB fusion

across 540 values of $priors \times \Pi_1 \times \Pi_3 \times min\ Looks$, with 108 levels for sensitivity analysis

and 5 levels of minimum looks. Black areas indicate MVB fusion is preferred, with a

TPM at least 5% better than PNN fusion. White areas indicate where PNN fusion is

preferred, and gray areas indicate a difference of less than 5% between the max TPR

achieved by each fusion method. Light and dark gray indicate PNN or MVB fusion is

preferred, but by less than 5%. Each plot contains 27 rows. The y-axis on each plot

includes the associated prior ratio, starting at H:F = 1:20 along with the three values of

Π_3, required declarations = 80%, 70% and 60%. The next 3 values on the y-axis are

associated with priors of H:F = 1:10, and the last three y-axis values are associated with

the 3-levels of Π_3 for a prior ratio of H:F = 20:1. The 20 columns on the x-axis

represent the levels associated with the minimum looks and the maximum allowable

critical error, Π_1. The first four values are associated with 1 minimum forced look for

$\Pi_1 = 1\%, 2\%, 3\%$ and 4%, followed by the other minimum looks evaluated for each of

the four levels of Π_1. The most difficult area to obtain feasible solutions and a high TPR

is in the upper left hand corner, while the least restrictive constraints are in the lower

right hand corner with high Hostile target densities.

Figure 5.34 shows how each cell is associated with a specific combination of

variable settings and indicates the preferred fusion when using the naturally ordered (ord)

Test data. The vertical black spikes located at columns 4, 8, 12, 16 and 20 with relatively

low ratios of H:F, show MVB fusion is preferred when the critical error, $\Pi_1 = 4\%$.

Medium gray horizontal rows through these spikes show when the declaration rate is

required to be 80%, MVB and PNN fusion are equivalent, when H:F is low. With the

priors at 1:4 and the minimum declaration rate at 70% or 80%, neither fusion method is

preferred. The remaining medium gray areas for priors = 1:20 through 1:1 indicates no

preferred fusion method. The white area shows PNN fusion is preferred in those limited

cases with 1 minimum look across the indicated high priors of H:F for different levels of

maximum critical error, Π_1. The PNN preference boundary changes systematically as

the H:F ratio decreases and Π_1 varies. Finally, with 2 or more minimum looks and a

prior ratio of 4:1 or higher, the two fusion methods generally yield a maximum TPR

within 5% of each other, except for a few cases with priors = 4:1 and $\Pi_3 = 80\%$ and for

$\Pi_1 = 1\%$ for 2-forced looks. The predominantly dark grey area in the Hostile dense

region with 2-4 forced looks, shows MVB fusion is preferred for much of these cases, but

the differences are limited.

Prefered Fusion based on max TPR for naturally ordered (ord) Test data

row	H:F	Π_3
1		80%
2	1:20	70%
3		60%
4		80%
5	1:10	70%
6		60%
7		80%
8	1:4	70%
9		60%
10		80%
11	1:2	70%
12		60%
13		80%
14	1:1	70%
15		60%
16		80%
17	2:1	70%
18		60%
19		80%
20	4:1	70%
21		60%
22		80%
23	10:1	70%
24		60%
25		80%
26	20:1	70%
27		60%

Π_1	1%	2%	3%	4%	1%	2%	3%	4%	1%	2%	3%	4%	1%	2%	3%	4%	1%	2%	3%	4%
min Looks		1				2				3				4				5		
column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Legend

- ☐ PNN fusion preferred based on max TPR
- ▨ PNN fusion preferred, but MVB fusion within 5%
- ▨ Equivalent TPR for MVB and PNN fusion
- ▨ MVB fusion preferred, but PNN fusion within 5%
- ■ MVB fusion preferred based on max TPR

Figure 5.34 Identification of Preferred Fusion using ord Test Data across 4 Variables: Horizontally by Maximum Critical Error (Π_1) & Minimum Looks and Vertically by Minimum Declaration Level (Π_3) & Priors

The next figure shows the preferred fusion method for each of the for correlation levels using the Training data. Each of the following four subplots is similar. A white area indicates PNN fusion is definitely preferred if limited to 1-forced look, across most values of the maximum critical error, Π_1. Large light gray areas then indicate for the

remaining cases of 1-forced look, 2-forced looks, and 3-forced looks for naturally ordered

and autocorrelated data, the PNN fusion is preferred, but by less than 5%. The remaining

medium gray areas indicate no preference in the two fusion methods. These areas of

equivalence were obtained by each fusion method obtaining the best TPR achievable

using the minimum number of forced looks (3,4 or 5) and $P_{TP} = 100\%$ for all Hostile

target vehicles declared. The best TPR achievable is 0.333 for 3-forced looks, 0.25 for 4-

forced looks and 0.20 for 5-forced looks.
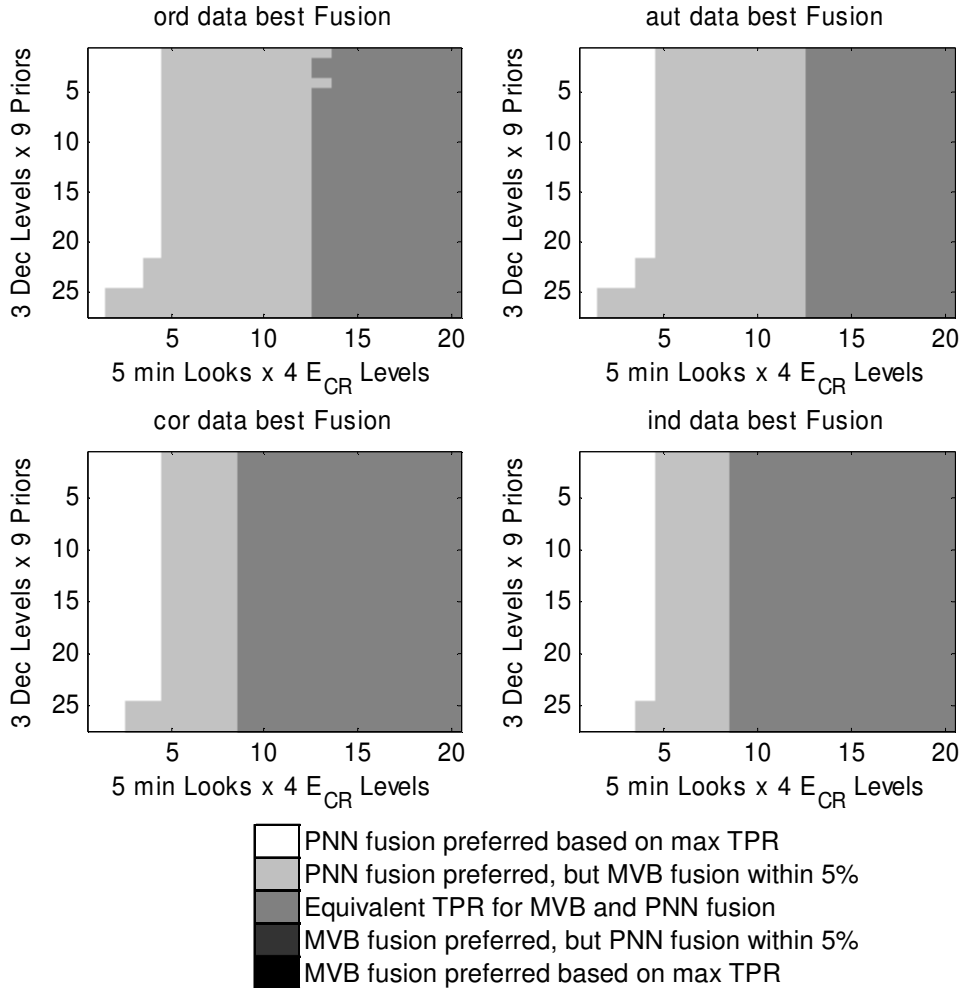
Figure 5.35 Training Data TPR Comparison across 5 Variables: Data Correlation,

Minimum Looks, Π_1, Π_3 and Priors

From the previous plots, the fusion methods appear robust to priors and minimum declaration rate, Π_3, as indicated by little horizontal differences. Both methods also appear robust with respect to the levels of maximum critical error allowable as indicated by mostly uniform vertical coloring of the plots. Finally, differences do appear based on the minimum number of forced looks with all fusion methods being equivalent for 4 and 5 forced looks, while PNN fusion is preferred by less than 5% of the maximum TPR for 2 or 3 forced looks depending on the data set. The two light gray cells in the naturally ordered data set also indicate that the minimum declaration level, Π_3, effects whether the PNN fusion is slightly preferred or equivalent with low priors, 4-forced looks and Π_1 = 1%.

As shown by the first example using naturally ordered Test data, significant differences in maximum TPR obtained by each fusion method are found as all three sensitivity analysis variables change across all five levels of minimum looks. As in the previous figure using Training data, Figure 5.36 presents a subplot associated with each sensor correlation data set. In general, all four subplots show a similar pattern. Specifically, PNN fusion is only definitely preferred in a limited number of cases with 1-minimum look and a high ratio of H:F. The area associated with definitely preferred MVB fusion tends to occur when the ratio of H:F is low (1:20 through 1:1), and with a larger number of forced looks. These areas do change depending on the specific sensor correlation data set and are not uniform across the maximum critical error, Π_1, as indicated by intermittent vertical patterns. Limited influence from the minimum declaration rate, Π_3, levels are also seen, as indicated by intermittent horizontal patterns

for the naturally ordered data set. For the cases with 2 or more forced looks and a ratio of H:F equal to or greater than 4:1, the two fusion methods usually provided a max TPR with less than 5% difference. Although for some cases with $\Pi_1 = 1\%$ and $\Pi_3 = 80\%$, the most restrictive values, and H:F = 4:1, MVB fusion is preferred as shown by black areas.

PNN fusion preferred based on max TPR
PNN fusion preferred, but MVB fusion within 5%
Equivalent TPR for MVB and PNN fusion
MVB fusion preferred, but PNN fusion within 5%
MVB fusion preferred based on max TPR
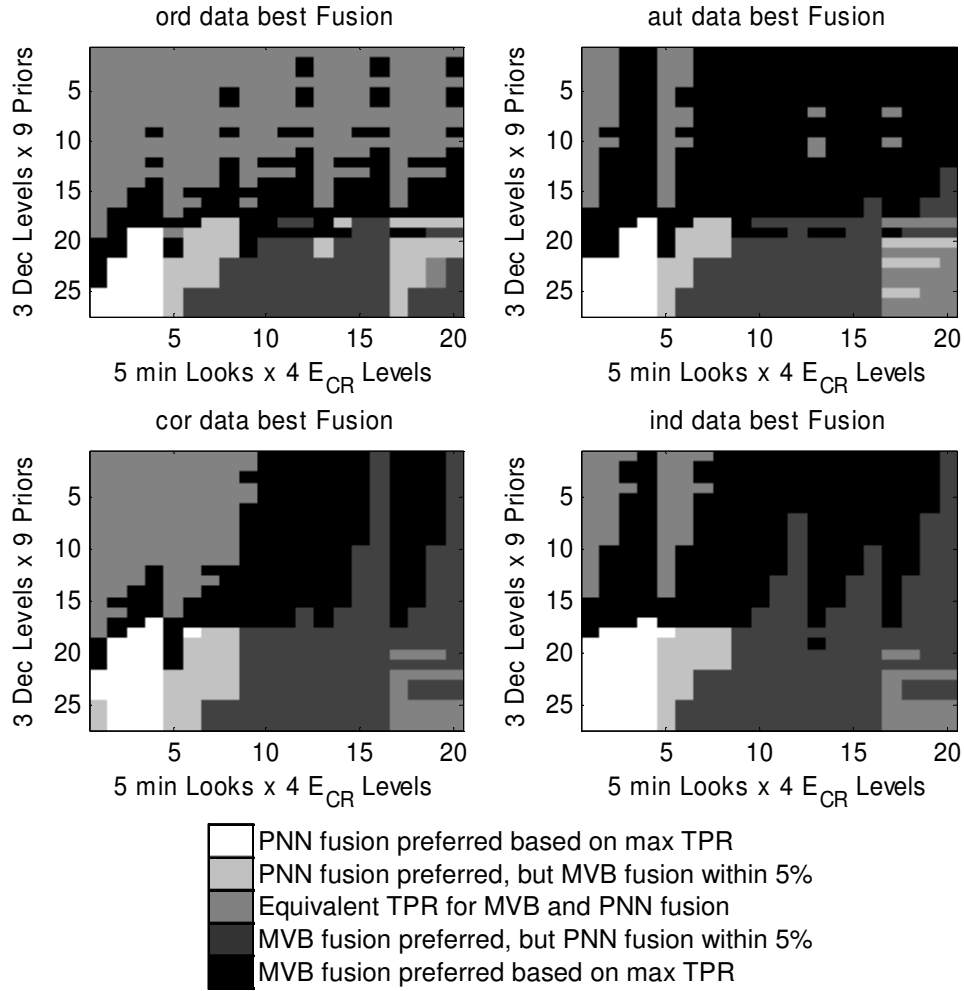
Figure 5.36 Test Data TPR Comparison across 5 Variables: Data Correlation,

Minimum Looks, Π_1, Π_3 and Priors

The second initial goal of the sensitivity analysis was to characterize the variables associated with each fusion method's ability to obtain a feasible solution. Plots similar to

those generated for the comparison of maximum TPR were generated to show the specific variable levels associated with feasibility for each fusion method. For the Training data, all 8 fusion model/data combinations were feasible across all 540 levels representing all possibilities of $priors \times \Pi_1 \times \Pi_3 \times \min Looks$. The next figure shows whether both fusion methods were feasible, only MVB fusion was feasible or neither fusion method is feasible across all levels of each of the three sensitivity analysis variables and the number of minimum forced looks. The black areas indicate neither fusion method was feasible. The gray areas indicate only the MVB fusion was feasible, and the white areas indicate both PNN and MVB fusion is feasible.

From evaluation of feasibility of both fusion methods with Test data across the four sensor correlations, it was discovered if PNN fusion was feasible, then MVB fusion was always feasible. This was true for all 540 levels of $priors \times \Pi_1 \times \Pi_3 \times \min Looks$. Thus, PNN fusion was only preferred to MVB, if it achieved a higher maximum TPR than MVB fusion. MVB fusion would be identified as preferred to PNN fusion for all cases where MVB was feasible and PNN was not. Feasibility by MVB fusion when PNN fusion is infeasible is indicated by the gray cells in Figure 5.37. These areas associated with different variable levels, coincide with many of the black areas in Figure 5.34 where only MVB fusion is feasible. Insight for areas of TPR equivalence is also obtained from viewing the feasibility figures. The black area in Figure 5.37 shows for H:F less than 1:1, most of the $priors \times \Pi_1 \times \Pi_3 \times \min Looks$ combinations are infeasible for both fusion models. These areas, where neither model is feasible, map to medium gray areas in Figures 5.34 and 5.36. Thus, while maximum TPR equivalence is indicated in the

237

Friendly rich environments, with more restrictive constraints for maximum critical error, Π_1, and for minimum declaration, Π_3, neither fusion method is feasible.

row	H:F	Π_3	Naturally ordered (ord) Test data feasibility
1		80%	
2	1:20	70%	
3		60%	
4		80%	
5	1:10	70%	
6		60%	
7		80%	
8	1:4	70%	
9		60%	
10		80%	
11	1:2	70%	
12		60%	
13		80%	
14	1:1	70%	
15		60%	
16		80%	
17	2:1	70%	
18		60%	
19		80%	
20	4:1	70%	
21		60%	
22		80%	
23	10:1	70%	
24		60%	
25		80%	
26	20:1	70%	
27		60%	

Π_1	1%	2%	3%	4%	1%	2%	3%	4%	1%	2%	3%	4%	1%	2%	3%	4%	1%	2%	3%	4%
min Looks		1				2				3				4				5		
column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Legend

☐ PNN and MVB fusion feasible
▨ MVB fusion feasible, PNN fusion not feasible
■ PNN and MVB fusion Not feasible

Figure 5.37 ord Test Data Feasibility Comparison across 4 Variables: Minimum Looks, Maximum Critical Error Π_1, Minimum Declarations Π_3 and Priors

238

This infeasible area with an equivalent TPR of 0.0 appears the same in Figures 5.34 and 5.36 as obtaining the best TPR achievable by each system, as may occur with 5 forced looks and high levels of H:F.

The next figure shows the feasibility map across all variables for Test data across all four sensor correlation levels. Overall, from the subplots of Figure 5.38, a general characterization of feasibility for each fusion method can be deduced. From these subplots, significant improvement in feasibility is observed as the sensor correlation structure changes and as the ratio of H:F increases. The large white and gray regions for autocorrelated, co-registered and independent data, indicate MVB fusion is feasible across most conditions. The MVB fusion is not feasible for cases with 1 or 2 minimum looks, H:F is low, and constraints are restricted to $\Pi_1 = 1\%$ and $\Pi_3 = 80\%$. Another region of MVB infeasibility is observed in the autocorrelated data when H:F = 1:4, $\Pi_1 = 1\%$ and $\Pi_3 = 80\%$, for both 4 and 5 minimum looks

PNN feasibility appears robust for prior ratios of H:F of 4:1 or greater across Π_1 and Π_3 and across the minimum number of looks. The top white horizontal line in all four data sets indicates PNN fusion is feasible when the ratio of priors is to 2:1, if the lowest required declaration rate of $\Pi_3 = 60\%$ is used. For the evaluation of feasibility using co-registered or independent data, PNN fusion was also feasible across reduced levels of H:F for 3-5 minimum looks as indicated by the increasing white vertical bars located at columns 15-16 and 18-20 for co-registered data and at columns 10-12, 14-16,
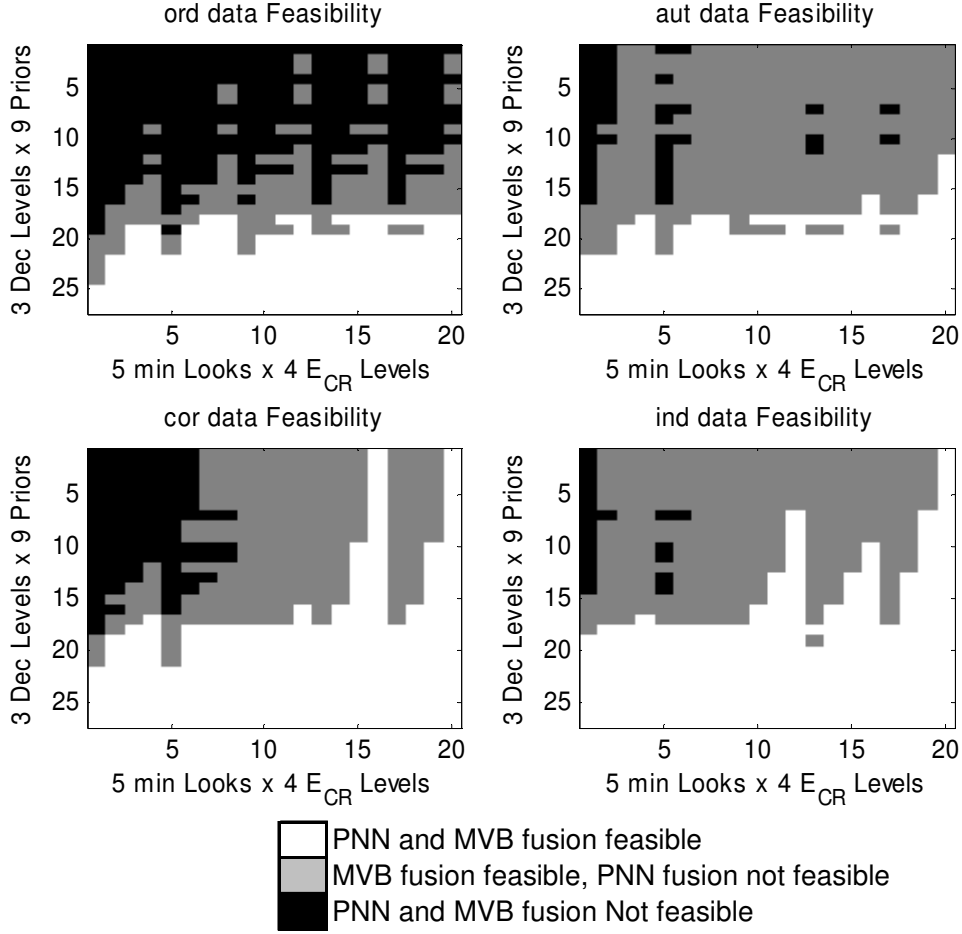
Figure 5.38 Test Data Feasibility Across 5 Variables: Data Correlation, Minimum Looks, Maximum Critical Error Π_1, Minimum Declarations Π_3 and Priors

and 17-20 for independent data. These white bars amidst the gray area are associated with a critical error constraint of $\Pi_1 = 2\text{-}4\%$. Thus, while PNN fusion does not require any assumptions of input data correlation, it does appear to benefit when the collected imagery is not autocorrelated in the naturally collected 4 degree increments. This is shown by the increased feasibility for the co-registered or independent data sets.

The MVB fusion feasibility also improves as a lower correlation is introduced either temporally within or across sensors. While evaluation with the naturally ordered

data shows a large percentage of black area, where MVB fusion is infeasible, evaluation

with the other three data sets shows significant improvement. The least improvement in

feasibility is observed from the naturally ordered to the co-registered data, giving some

indication that MVB fusion performs better in those cases with reduced correlation

between sensors. Since both the autocorrelated and independent data sets use sensor data

collected at independent aspect angles for any given look at time t, the two sensor labels

may have a higher likelihood of disagreement and force another look. While these extra

looks will reduce TPR, they may facilitate a reduction in critical error as additional looks

are obtained before declaring a final "TOD," "OH" or "FN" label. Finally, as previously

described for Figure 5.37, MVB feasibility for the naturally ordered data set was

significantly affected by levels of Π_1, Π_3 and priors, and to a lesser extent, across the

minimum looks required.

To gain additional insight of the fusion system operating characteristics,

additional plots were generated to compare specific values obtained with PNN and MVB

fusion, given each of the four Test data sensor correlations. Specific performance

measures associated with each fusion system include the maximum TPR, the associated

average looks to obtain a True Positive declaration, the percentage of feasible thresholds,

the percentage of declared targets after five looks, the percentage of targets declared

"ND" given assessment of a Hostile and the percentage of targets declared "ND" given

assessment of a Friend. Performance for PNN fusion is indicated by circles and MVB is

indicated by triangles. Values of the maximum critical error allowed were varied from

$\Pi_1 = \{1\%\text{-}4\%\}$. Each value of Π_1 was used to select a different gray scale with the

lightest gray being the most restrictive $\Pi_1 = 1\%$ and black indicating $\Pi_1 = 4\%$. The

minimum declaration rate was held at a constant $\Pi_3 = 70\%$ for all plots.

The next figure shows TPR vs. minimum forced looks across priors and Π_1.

From these plots a TPR value of 0 likely coincides with no feasible operating points.

From the plots below, with H:F priors of 1:20 and 1:10, the MVB fusion may be feasible

depending on the level of Π_1 with approximately the same maximum TPR around 0.20

obtained for 2, 3,4 or 5 forced looks. For the case of H:F at 1:4, no TPR significantly

greater than 0 appears for any number of minimum looks. TPR then shows an increase

for MVB fusion across priors of 1:2 and up, and the PNN fusion obtains feasibility when
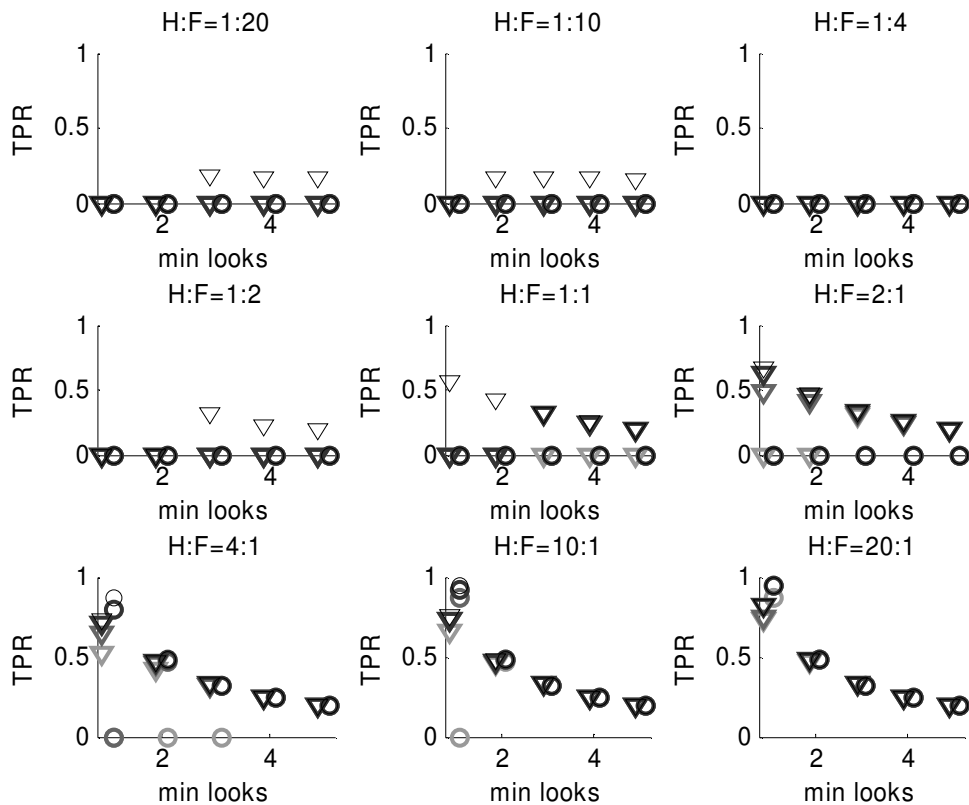
Figure 5.39a Comparison of TPR for "ord" Test Data across Priors and Π_1

priors are 4:1 or greater. In addition, the minimum 1-look Hostile rich environment

shows PNN fusion as preferred to MVB fusion. To show how TPR is affected by the

ratio of priors and minimum looks, the associated looks per True Positive hostile ID are

plotted below. These plots show for low H:F priors, on average over 5 looks are required

to make a TP declaration. With a maximum of 5-looks per vehicle, this number also

includes the looks used to misidentify a Hostile. The plots associated with H:F = 10:1

and 20:1 then show how both fusion methods obtain the maximum achievable TPR using

the minimum number of forced looks across all levels of Π_1.
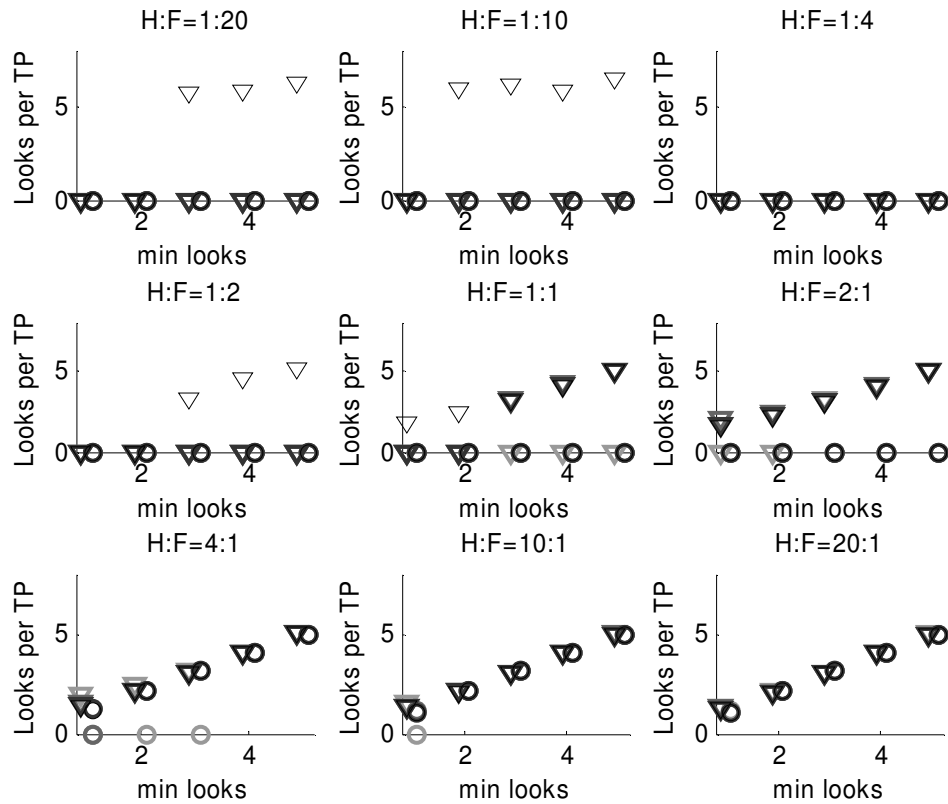
Figure 5.39b Comparison of Looks per TP for "ord" Test Data across Priors and Π_1

Previous plots have shown conditions across priors, Π_1, Π_3 and the minimum number of looks where each fusion method had at least one feasible set of thresholds out of 10,000 thresholds assessed. While the fusion method may be indicated as feasible with a very limited number of feasible thresholds, further assessments of the feasibility may indicate a more robust system. Differences in the percentage of feasible thresholds is also seen across the four levels of Π_1. For example, with a prior ratio of 4:1, the percentage of feasible thresholds for MVB fusion varies significantly between all four values of Π_1. For the same ratio of H:F = 4:1 the PNN fusion appears to behave in a bimodal manner with either close to 0% or 100% of the 10,000 thresholds feasible.
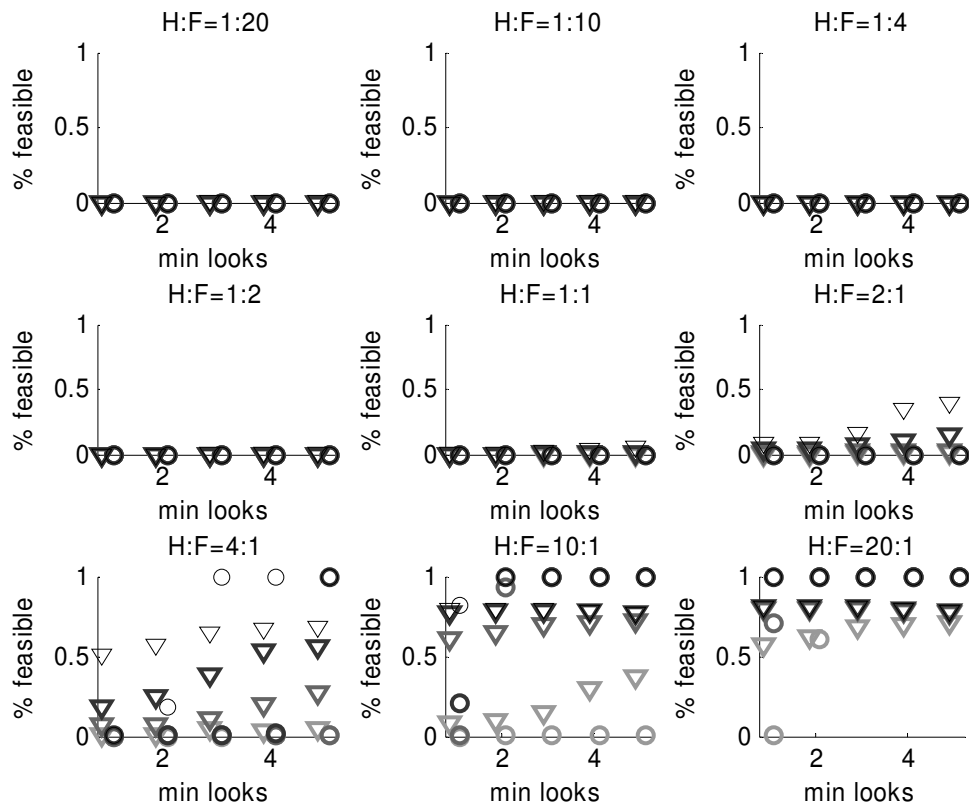
Figure 5.39c Comparison of % Feasible for "ord" Test Data across Priors and Π_1

244

The next plot shows how the percentage of declarations at the optimal TPR varies across priors, Π_1 and the minimum number of looks. These plots are all generated for a minimum declaration rate of $\Pi_3 = 70\%$. Since feasibility requires a minimum declaration rate of 70%, all indications with the %Declared at 0 indicate infeasibility. From the plots below, the variation associated with %Declared indicates that this is not always a binding constraint value at the maximum TPR. For low H:F priors, the percent declared does look to be close to 70%; yet, for values of H:F at 10:1 and 20:1, when a fusion method is feasible, the percent declared appears much higher and may even be close to 100%.
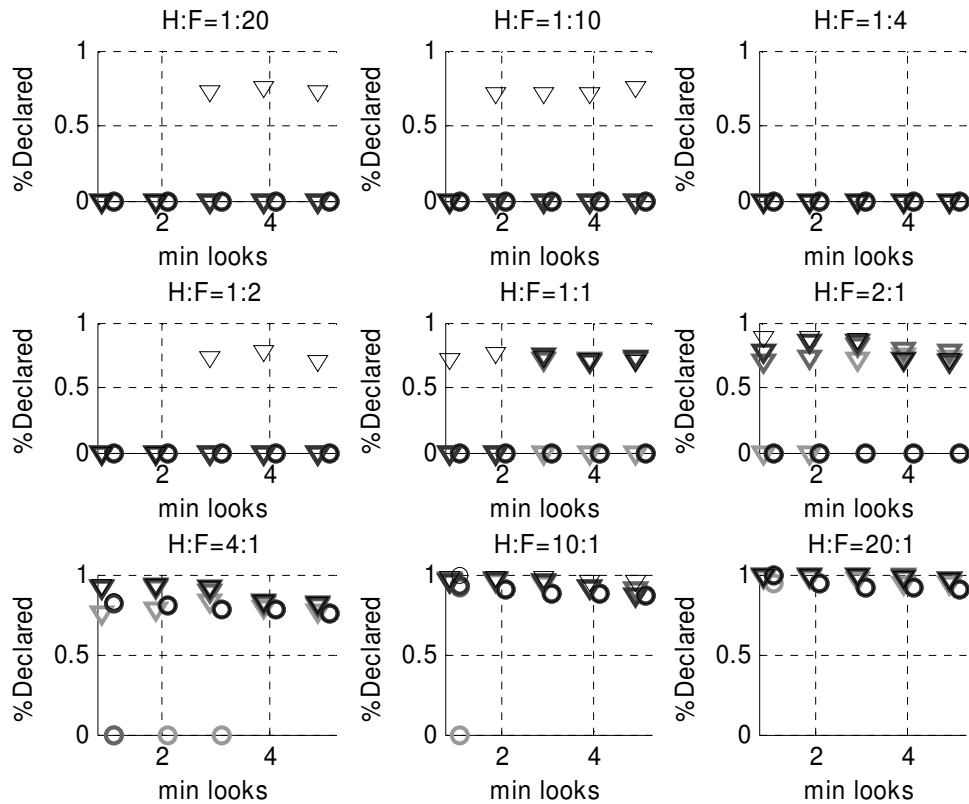
Figure 5.39d Comparison of % Declared for "ord" Test Data across Priors and Π_1

To help explain why some of the MVB fusion is feasible at low priors of H:F = 1:20 or 1:10 and no feasibility is shown for a ratio of 1:4 plots of the percentage of "Non-declarations" by Hostile and Friendly targets is useful. The next subplot show how the percentage of "ND" declarations is apportioned to Hostile targets for the optimal TPR across priors, Π_1, and the minimum number of looks. From these plots when the ratio of H:F is 1:10 or 1:20, almost 100% of the Hostile targets are declared as "ND" or "Unknown." Low target densities, with Hostiles comprising less than 5% or 10% of the total targets, allows for a system to easily classify most of the Hostiles as "ND." Other plots show little difference in the percentage of Hostile "ND's" at the max TPR.
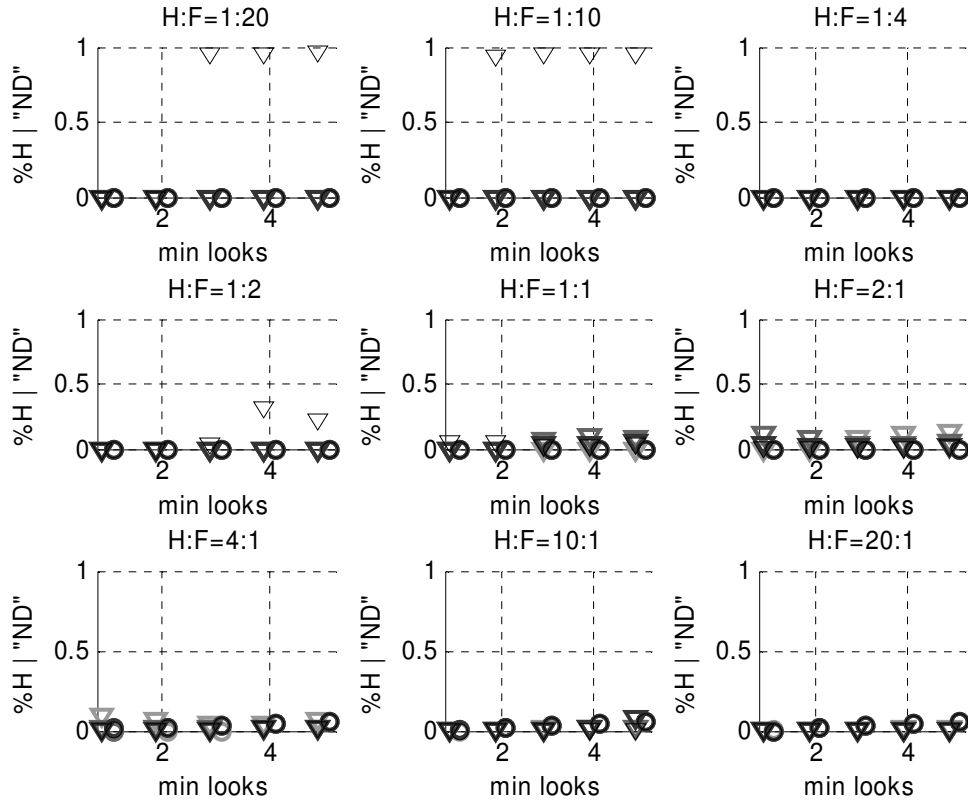
Figure 5.39e Comparison of % Hostile | "ND" for "ord" Data across Priors and Π_1

To gain insight of the fusion systems at optimal TPR, similar plots of the percentage of "ND's" being Friendly targets is useful. From these plots, considerably more variability is observed as compared to the percentage of Hostile targets labeled as "ND." Because the objective function seeks to maximize True Positive Hostile declaration across looks, the fusion systems appear to increase the proportion of Friend/Neutral targets that are labeled "ND" as the ratio of Friends gets lower. This is similar to the Hostile "ND" labels when H:F was either 1:20 or 1:10. For Hostile target rich priors of 4:1 - 20:1, PNN fusion with 4 and 5 minimum looks declares almost 100% of Friends as "ND" or "Unknown."
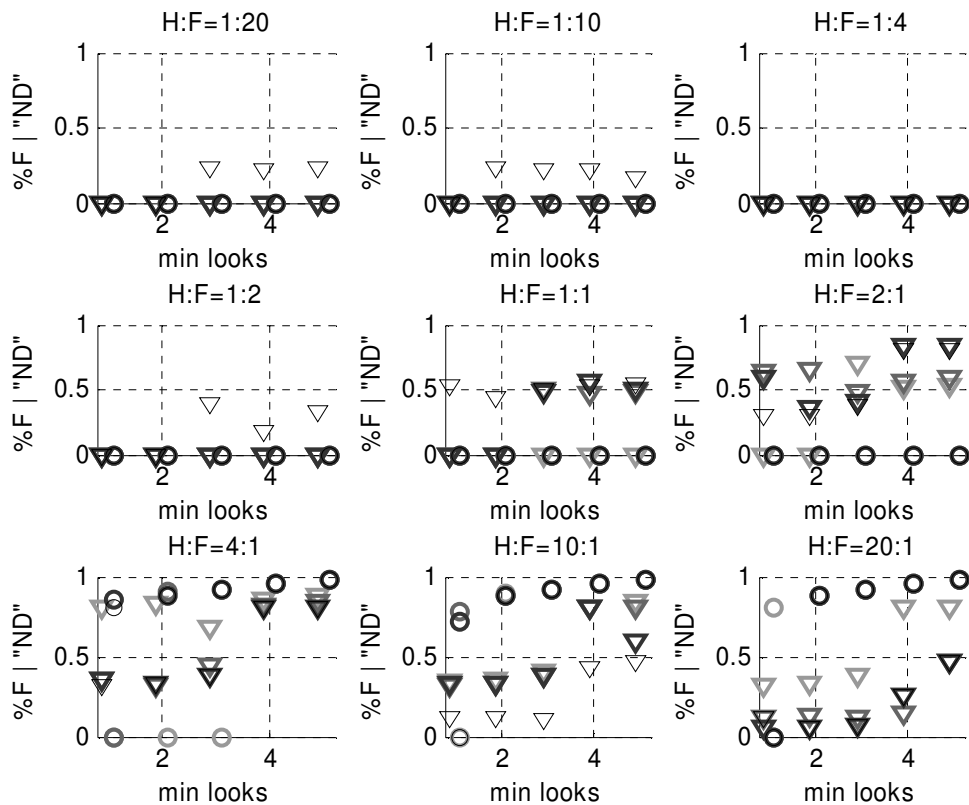
Figure 5.39f Comparison of % Friend | "ND" for "ord" Data across Priors and Π_1

The same figures associated with the autocorrelated, co-registered and independent data sets were visually analyzed as well, with comparable findings.

As a final means to compare the two fusion models across all sensitivity analysis variables and across the five minimum look levels and the four correlation structures, grayscale plots associated with the maximum TPR and percentage of feasible thresholds are presented next. These plots are similar to the previous plots where grayscale was used across the $priors \times \Pi_3$ on the vertical axis and $\min Looks \times \Pi_1$ on the horizontal axis. Instead of plotting an associated winner for each design point, the individual performance of each fusion system is given. To do so, the next two figures include eight subplots each, with four MVB subplots and four PNN subplots. The first value plotted shows the optimal TPR associated with each point, as a percentage of the best TPR obtainable. Plotting values scaled within [0,1] facilitates plotting across all min $Look$ values across the same range and is computed as shown in the next equations.

$$\%bestTPR = 1 - \left(\frac{bestTPR - \max TPR}{bestTPR} \right) = 1 - \left(\frac{\frac{1}{\min Looks} - \max TPR}{\frac{1}{\min Looks}} \right) \qquad (5\text{-}31)$$

$$\%bestTPR = 1 - \left(1 - \min Looks \bullet \max TPR \right) = \left(\min Looks \right) \left(\max TPR \right) \qquad (5\text{-}32)$$

For *%bestTPR* = 1, the fusion model achieved the maximum obtainable TPR of $1/\left(\min Looks \right)$ and is indicated by the white areas in the next plot. Black areas correspond to *%bestTPR* = 0, where no feasible thresholds were obtained. The gray scale indicates performance between these two extremes. Figure 5.40 shows performance
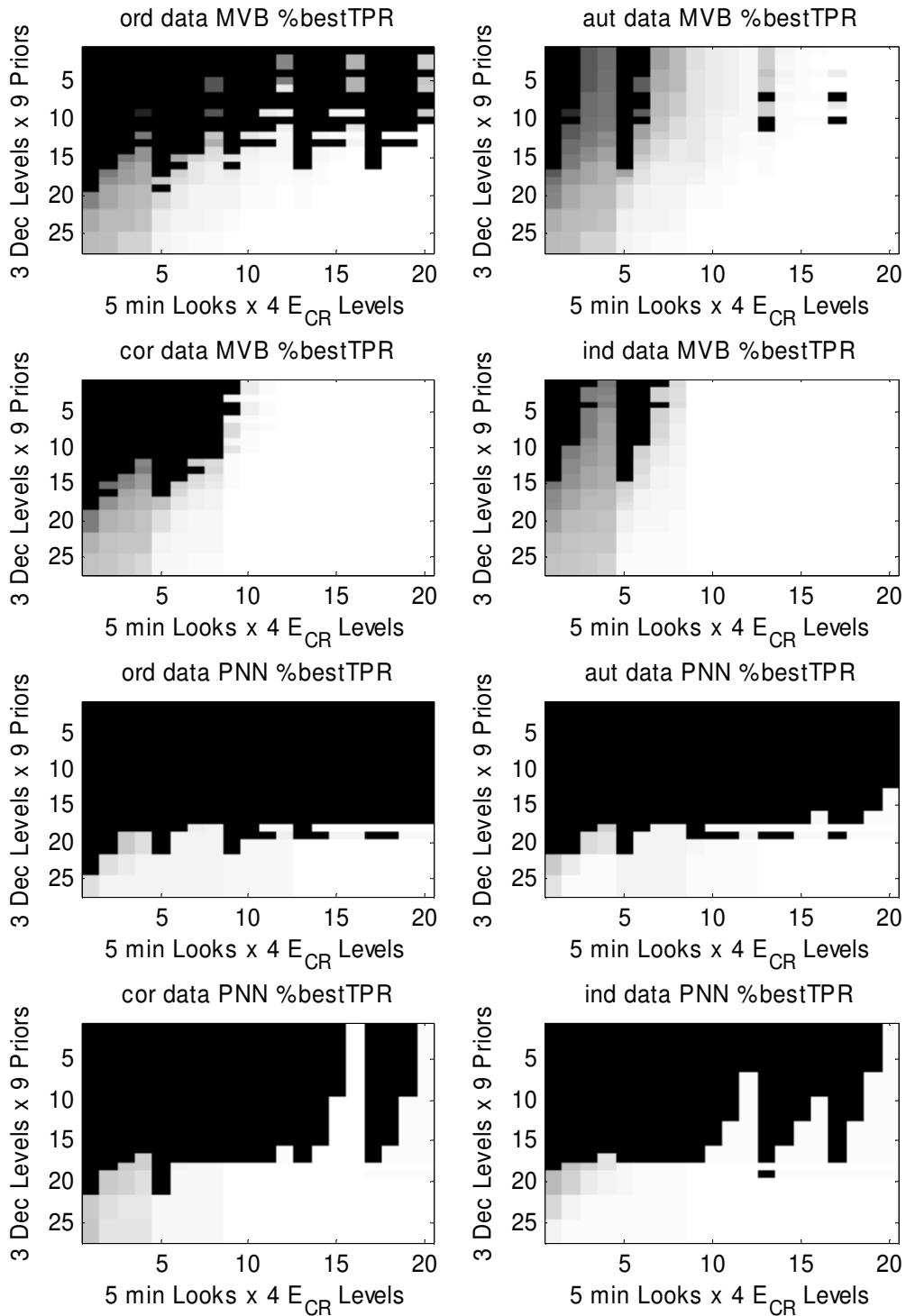
Figure 5.40 % of *best* TPR across All Variables Obtained by MVB and PNN Fusion

of MVB fusion is significantly affected by min *Looks* and the critical error constraint Π_1. Performance of PNN fusion shows less variation from the *best TPR*, with many cells either black or white indicating a bimodal function that is either infeasible or very close to the highest TPR achievable. PNN fusion clearly shows an improvement as the critical error constraint is relaxed for cases of 3 or more looks and evaluated using data sets other than naturally ordered. As previously seen, the primary factor influencing PNN feasibility appears to be ratio of priors, which gains some feasibility with a prior ratio of 4:1 and a relaxed declaration constraint of $\Pi_3 = 80\%$. MVB fusion, with both intermittent vertical and horizontal patterns, shows more sensitivity to both Π_1 and Π_3.

One goal of the sensitivity analysis is to determine the robustness of solutions, given perturbations of the variables of interest. While Figure 5.35 shows the preferred regions of each fusion method, these are based on the single best TPR obtained, given a specific Test data set. To help gain further confidence in fusion system robustness, the next eight subplots are offered to show robustness against selection of optimal thresholds. For the following figures, black indicates no feasible thresholds, while white indicates 100% of all assessed thresholds are feasible. From these plots, patterns across all four variables are observed. A 2-way interaction between priors and minimum looks appears to be the most significant relationship for determining feasibility for MVB fusion, followed by Π_1 and then Π_3 with the least influence. PNN fusion feasibility appears most influenced by the ratio of priors, followed by a 2-way interaction between forced looks and Π_1, with the least variability associated with Π_3. Also, the variability between sensor correlation data sets indicates correlation significantly affects feasibility.

250
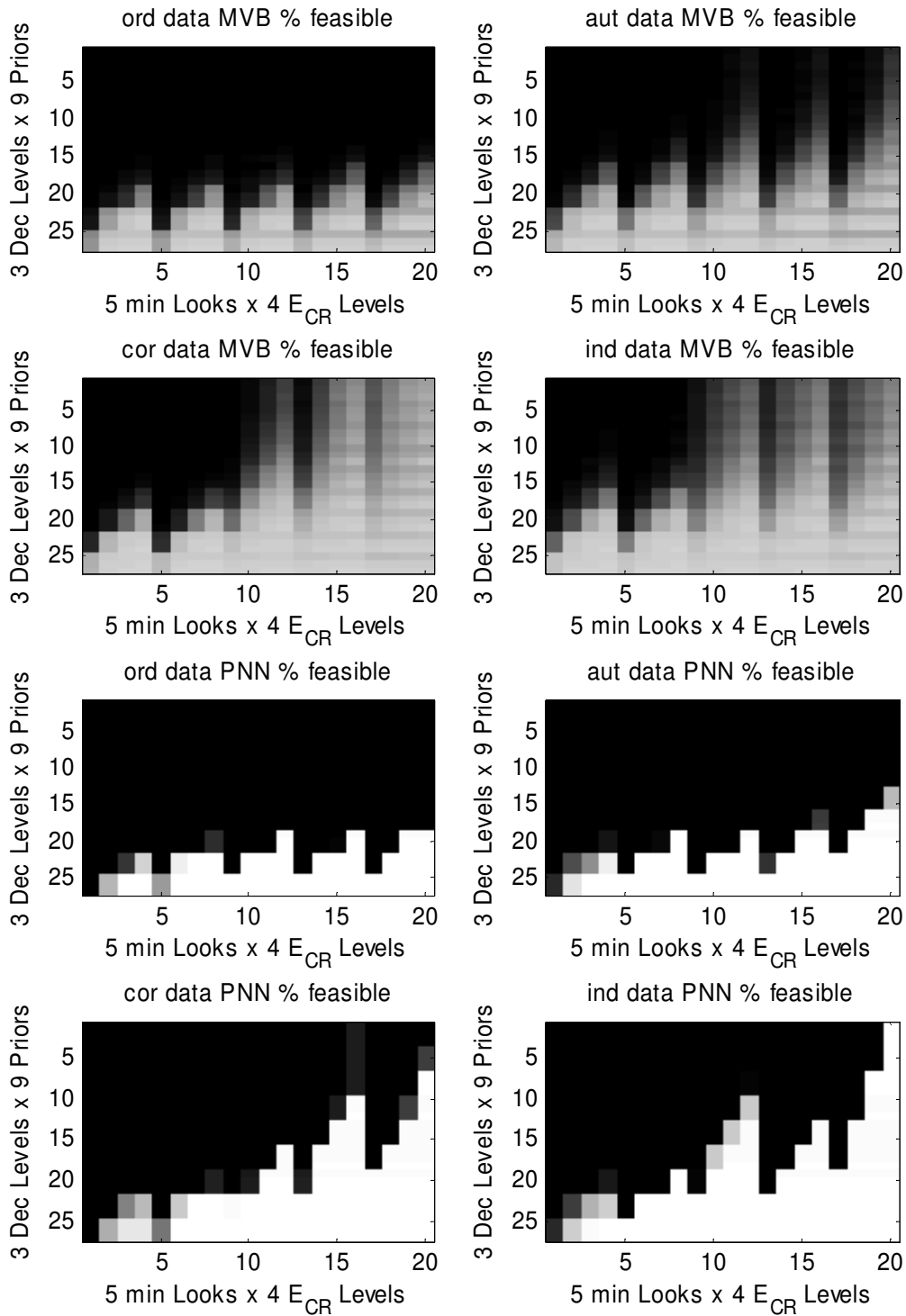
Figure 5.41 % Feasible across All Variables for MVB and PNN Fusion

These plots also give some insight for the low prior ratios of H:F = 1:20 or 1:10, in which MVB fusion was feasible and preferred to PNN fusion. For the cases of ordered and autocorrelated data, very limited feasible thresholds were obtained by the MVB fusion model. This is also true for the case of MVB fusion with 1 or 2 minimum looks, when evaluated with co-registered or independent Test data. In these situations, MVB fusion may be preferred, but with potential variations across other test data with slightly different EOC's from the Training data, these feasible points may become infeasible. Thus, for these cases of very limited feasible solutions less confidence should be placed on declaring one system better than the other.

A final sensitivity analysis summary plot is included next. Each of the subplots are generated from evaluation using one of the four Test data sets, and boundaries of fusion model preference are indicated by grayscale. Black denotes both fusion systems are infeasible or have less than 0.5% (< 50 of 10,000) feasible thresholds, which may correspond to a fusion system that is either ineffective or not robust with respect to threshold levels. White areas denote equivalent fusion performance as determined by either system achieving a max TPR within 2.5% of the other system. The light grey areas show where PNN fusion is preferred and the dark gray areas show where MVB fusion is preferred. From these plots, clear differences between data sets are observable; yet, a general trend exists. With low priors and minimal forced looks, neither system performs robustly, with few if any feasible thresholds. With low H:F, more forced looks, and less correlated data, the MVB fusion is preferred. For a limited area associated with high H:F and 1 minimum look PNN fusion is preferred. Then, with high H:F and 2-5 minimum

252

looks, both systems appear equivalent. The influence of tightening or relaxing Π_1, the critical error constraint, or Π_3, the declaration constraint, has increased influence at the boundaries of these four general area, where Π_1 has significant influence across all priors for the three generated correlation levels.
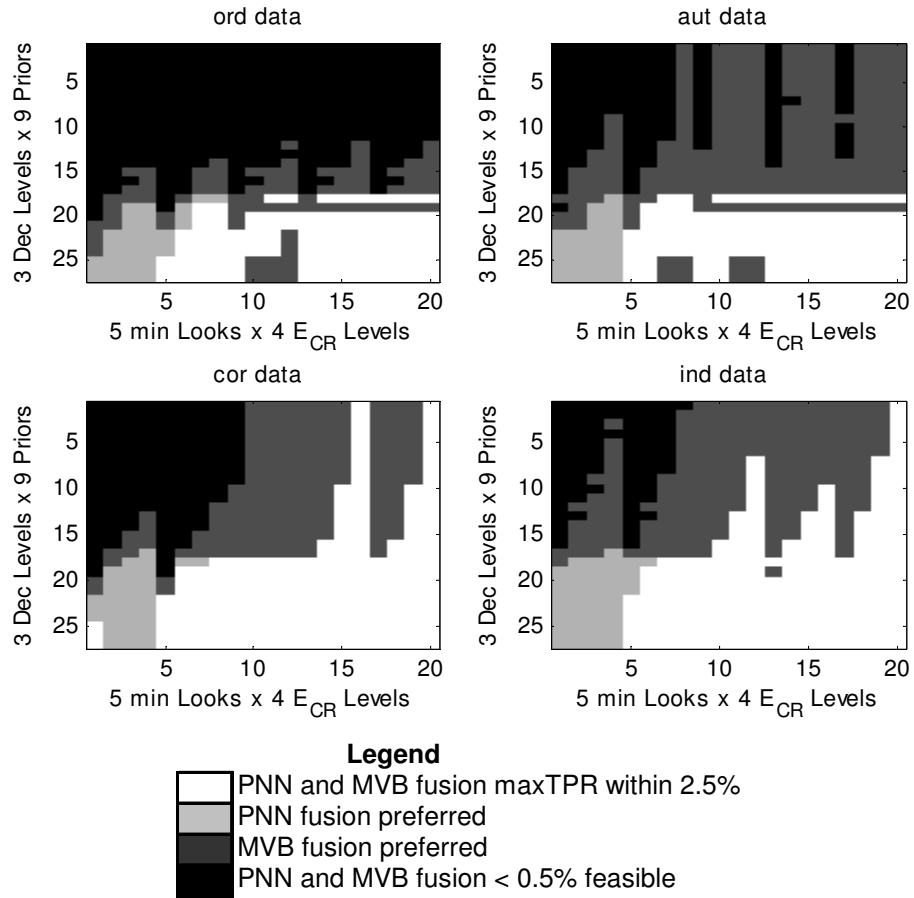
Figure 5.42 Preferred Fusion Method across Variables and Test Data

Overall, MVB fusion appears to be more robust across the entire range of operating conditions and sensitivity analysis variables. By obtaining feasibility, MVB fusion outperforms PNN fusion in many cases. PNN fusion is significantly hindered by

cases of low H:F with limited feasibility. Yet, when PNN does become feasible, it achieves the overall highest max TPR obtainable at close to 1 TP per look, and is preferred for some operating conditions.

5.8 Temporal Comparison across Correlation Levels

For the evaluation of TPR performed to this point, an assumed equivalent time per look was used to calculate the TP rate as: $\max TPR = P_{TP} / (mean\ looks\,|\,Hostile)$. Three data sets were generated by randomly ordering some of the samples and may represent multiple flight passes collecting less correlated data. Temporal performance across correlation structures and H:F priors is indicated in the next 2 tables, using the minimum mean number of looks to indicate the preferred system for each level of priors. The name indicates the fusion method (PNN or MVB) assessed using one of the correlation structures of Test data (ord, aut, cor or ind) followed by the number of minimum looks required by the fusion algorithm. The initial constraint values for maximum critical error, $\Pi_1 = 2\%$, maximum non-critical error, $\Pi_2 = 5\%$, and minimum declaration rate, $\Pi_3 = 70\%$ were held constant. Cells with less than two looks required to obtain a True Positive Hostile declaration are highlighted gray. An "*InF*" in gray print indicates the fusion system was not feasible for a given condition. These 2 tables are sorted by the number of looks per TP at the optimal TPR and show the preferences for each level of H:F. In general, the best performance occurs for the independent data followed by co-registered or autocorrelated data, and finally followed by the naturally ordered data. Also, when feasible, fusion models with less forced looks are preferred. With target densities of 1:2 and 1:1, some of the MVB fusion models with independent data and 2

254

forced looks are preferred to other feasible MVB models using data with other within and

across sensor correlation.

Table 5.28a Mean Number of Looks/TP Associated with max TPR for Each Fusion Algorithm Sorted across all Data Correlation and Minimum Looks for Low H:F

H:F= 1:20		H:F= 1:10		H:F= 1:4		H:F= 1:2		H:F= 1:1	
fusion	looks/TP	fusion	looks/TP	fusion	looks/TP	fusion	looks/TP	fusion	looks/TP
MVBind3	3.02	MVBind3	3.02	MVBind3	3.02	MVBind1	2.04	MVBind1	1.77
MVBaut3	3.28	MVBcor3	3.24	MVBcor3	3.08	MVBind2	2.44	MVBind2	2.25
MVBcor3	3.29	MVBaut3	3.28	MVBaut3	3.28	MVBaut1	2.79	MVBaut1	2.32
MVBcor4	4.01	MVBcor4	4.01	MVBind4	4.00	MVBind3	3.02	MVBaut2	2.73
MVBind4	4.01	MVBind4	4.01	MVBcor4	4.01	MVBcor3	3.04	MVBind3	3.01
MVBaut4	4.14	MVBaut4	4.14	MVBaut4	4.13	MVBaut2	3.19	MVBcor3	3.03
MVBaut5	5.00	MVBaut5	5.00	MVBcor5	5.00	MVBaut3	3.24	MVBaut3	3.15
MVBcor5	5.00	MVBcor5	5.00	MVBind5	5.00	MVBind4	4.00	MVBord3	3.24
MVBind5	5.00	MVBind5	5.00	MVBaut5	5.01	MVBcor4	4.01	MVBcor4	4.00
MVBaut1	InF	MVBaut2	5.37	MVBaut1	InF	MVBaut4	4.07	MVBind4	4.00
MVBaut2	InF	MVBaut1	InF	MVBaut2	InF	MVBaut5	5.00	MVBaut4	4.04
MVBcor1	InF	MVBcor1	InF	MVBcor1	InF	MVBcor5	5.00	MVBord4	4.19
MVBcor2	InF	MVBcor2	InF	MVBcor2	InF	MVBind5	5.00	MVBaut5	5.00
MVBind1	InF	MVBind1	InF	MVBind1	InF	MVBcor1	InF	MVBcor5	5.00
MVBind2	InF	MVBind2	InF	MVBind2	InF	MVBcor2	InF	MVBind5	5.00
MVBord1	InF	MVBord1	InF	MVBord1	InF	MVBord1	InF	MVBord5	5.03
MVBord2	InF	MVBord2	InF	MVBord2	InF	MVBord2	InF	PNNind5	5.10
MVBord3	InF	MVBord3	InF	MVBord3	InF	MVBord3	InF	MVBcor1	InF
MVBord4	InF	MVBord4	InF	MVBord4	InF	MVBord4	InF	MVBcor2	InF
MVBord5	InF	MVBord5	InF	MVBord5	InF	MVBord5	InF	MVBord1	InF
PNNaut1	InF	PNNaut1	InF	PNNaut1	InF	PNNaut1	InF	MVBord2	InF
PNNaut2	InF	PNNaut2	InF	PNNaut2	InF	PNNaut2	InF	PNNaut1	InF
PNNaut3	InF	PNNaut3	InF	PNNaut3	InF	PNNaut3	InF	PNNaut2	InF
PNNaut4	InF	PNNaut4	InF	PNNaut4	InF	PNNaut4	InF	PNNaut3	InF
PNNaut5	InF	PNNaut5	InF	PNNaut5	InF	PNNaut5	InF	PNNaut4	InF
PNNcor1	InF	PNNcor1	InF	PNNcor1	InF	PNNcor1	InF	PNNaut5	InF
PNNcor2	InF	PNNcor2	InF	PNNcor2	InF	PNNcor2	InF	PNNcor1	InF
PNNcor3	InF	PNNcor3	InF	PNNcor3	InF	PNNcor3	InF	PNNcor2	InF
PNNcor4	InF	PNNcor4	InF	PNNcor4	InF	PNNcor4	InF	PNNcor3	InF
PNNcor5	InF	PNNcor5	InF	PNNcor5	InF	PNNcor5	InF	PNNcor4	InF
PNNind1	InF	PNNind1	InF	PNNind1	InF	PNNind1	InF	PNNcor5	InF
PNNind2	InF	PNNind2	InF	PNNind2	InF	PNNind2	InF	PNNind1	InF
PNNind3	InF	PNNind3	InF	PNNind3	InF	PNNind3	InF	PNNind2	InF
PNNind4	InF	PNNind4	InF	PNNind4	InF	PNNind4	InF	PNNind3	InF
PNNind5	InF	PNNind5	InF	PNNind5	InF	PNNind5	InF	PNNind4	InF
PNNord1	InF	PNNord1	InF	PNNord1	InF	PNNord1	InF	PNNord1	InF
PNNord2	InF	PNNord2	InF	PNNord2	InF	PNNord2	InF	PNNord2	InF
PNNord3	InF	PNNord3	InF	PNNord3	InF	PNNord3	InF	PNNord3	InF
PNNord4	InF	PNNord4	InF	PNNord4	InF	PNNord4	InF	PNNord4	InF
PNNord5	InF	PNNord5	InF	PNNord5	InF	PNNord5	InF	PNNord5	InF

Table 5.28b Mean Number of Looks/TP Associated with max TPR for Each Fusion Algorithm Sorted across all Data Correlation and Minimum Looks for High H:F

H:F= 2:1		H:F= 4:1		H:F= 10:1		H:F= 20:1	
fusion	looks/TP	fusion	looks/TP	fusion	looks/TP	fusion	looks/TP
MVBind1	1.53	PNNind1	1.22	PNNind1	1.06	PNNind1	1.03
MVBaut1	1.73	PNNcor1	1.27	PNNaut1	1.09	PNNaut1	1.03
MVBcor1	1.74	MVBind1	1.39	PNNord1	1.15	PNNord1	1.06
MVBord1	2.08	MVBcor1	1.44	PNNcor1	1.16	PNNcor1	1.12
MVBind2	2.12	MVBaut1	1.52	MVBcor1	1.32	MVBcor1	1.29
MVBcor2	2.21	MVBord1	1.54	MVBind1	1.33	MVBind1	1.30
MVBaut2	2.27	PNNind2	2.06	MVBord1	1.37	MVBord1	1.34
MVBord2	2.45	PNNcor2	2.07	MVBaut1	1.38	MVBaut1	1.34
MVBind3	3.01	MVBind2	2.07	PNNind2	2.06	MVBind2	2.05
MVBcor3	3.02	PNNaut2	2.11	MVBind2	2.06	PNNind2	2.06
PNNind3	3.05	MVBcor2	2.12	PNNcor2	2.07	PNNcor2	2.07
MVBaut3	3.09	PNNord2	2.13	MVBcor2	2.07	MVBcor2	2.07
MVBord3	3.12	MVBaut2	2.16	PNNord2	2.10	MVBaut2	2.09
MVBcor4	4.00	MVBord2	2.19	MVBord2	2.11	MVBord2	2.10
MVBind4	4.00	MVBind3	3.01	MVBaut2	2.11	PNNord2	2.10
MVBaut4	4.03	MVBcor3	3.01	MVBind3	3.01	PNNaut2	2.11
PNNcor4	4.06	PNNind3	3.02	MVBcor3	3.01	MVBind3	3.00
MVBord4	4.08	PNNcor3	3.04	PNNind3	3.02	MVBcor3	3.01
PNNind4	4.08	MVBaut3	3.06	PNNcor3	3.04	PNNind3	3.02
MVBaut5	5.00	MVBord3	3.08	MVBord3	3.04	MVBord3	3.03
MVBcor5	5.00	PNNaut3	3.09	MVBaut3	3.05	MVBaut3	3.03
MVBind5	5.00	PNNord3	3.11	PNNaut3	3.09	PNNcor3	3.04
MVBord5	5.03	MVBcor4	4.00	PNNord3	3.11	PNNaut3	3.09
PNNcor5	5.09	MVBind4	4.00	MVBcor4	4.00	PNNord3	3.11
PNNind5	5.10	PNNcor4	4.02	MVBind4	4.00	MVBcor4	4.00
PNNaut1	InF	MVBaut4	4.02	PNNcor4	4.02	MVBind4	4.00
PNNaut2	InF	PNNind4	4.02	MVBaut4	4.02	PNNcor4	4.02
PNNaut3	InF	PNNaut4	4.05	PNNind4	4.02	MVBaut4	4.02
PNNaut4	InF	MVBord4	4.05	MVBord4	4.04	PNNind4	4.02
PNNaut5	InF	PNNord4	4.06	PNNaut4	4.05	MVBord4	4.02
PNNcor1	InF	MVBaut5	5.00	PNNord4	4.06	PNNaut4	4.05
PNNcor2	InF	MVBcor5	5.00	MVBaut5	5.00	PNNord4	4.06
PNNcor3	InF	MVBind5	5.00	MVBcor5	5.00	MVBaut5	5.00
PNNind1	InF	PNNaut5	5.00	MVBind5	5.00	MVBcor5	5.00
PNNind2	InF	PNNcor5	5.00	PNNaut5	5.00	MVBind5	5.00
PNNord1	InF	PNNind5	5.00	PNNcor5	5.00	MVBord5	5.00
PNNord2	InF	PNNord5	5.00	PNNind5	5.00	PNNaut5	5.00
PNNord3	InF	MVBord5	5.01	PNNord5	5.00	PNNcor5	5.00
PNNord4	InF	PNNaut1	InF	MVBord5	5.01	PNNind5	5.00
PNNord5	InF	PNNord1	InF			PNNord5	5.00

The fusion performance associated with Hostile rich environments indicates for feasible fusion, min look, and correlation combinations; fewer looks per TP are in general

taken by algorithms with fewer forced looks. Fewer forced looks are thus preferred, as indicated by lower mean estimated looks per TP, but may be become infeasible as the prior ratio of Hostiles to Friends decreases. Thus, in a Hostile rich environment both PNN and MVB fusion, with 1-forced look, surface to the top of the lists, and the best looks per TP are typically associated with the independent data. The next best performance is obtained by autocorrelated data for the PNN, but for MVB fusion, co-registered data appears to be almost on par with independent data. For both fusion methods, the lowest performance appears to be associated with the naturally ordered data.

A general preference for data containing less natural correlation is reasonable; yet, to obtain less correlated data additional time or another sensor platform may be required. If a different time unit is associated with each of the looks the four correlated data sets, the mean number of looks may provide additional insight to determine if the extra time and assets required to collect data with less correlation is advantageous. The following assessment will use the naturally ordered data as the baseline time unit, where 1 ordered look = 1 time unit. The autocorrelated data may be taken by two platforms at the same time. To facilitate the registration and information flow between two platforms, the time per autocorrelated look, may be assumed to be 1.2 ord-time units. The same registration issue arises for the independent data as well, so it to should be penalized to account for this extra requirement. Further, the independent data may be thought of as a new flight pass for each look through time, so starting with the second look, an additional 2 ord-time units will be added to the mean number of looks for each look greater than 1. Finally, the co-registered data may be thought of as a single platform taking up to 5 flight passes, so it is only penalized by the 2 ord-time units for any looks greater than 1.

Table 5.29 Example Time/TP Associated with max TPR for Each Fusion Algorithm
Sorted across All Data Correlation and Minimum Looks for High H:F

H:F= 2:1		H:F= 4:1		H:F= 10:1		H:F= 20:1	
fusion	time/TP	fusion	time/TP	fusion	time/TP	fusion	time/TP
MVBcor1	1.74	PNNcor1	1.27	PNNord1	1.15	PNNord1	1.06
MVBind1	1.83	MVBcor1	1.44	PNNcor1	1.16	PNNcor1	1.12
MVBaut1	2.07	PNNind1	1.46	PNNind1	1.27	PNNind1	1.23
MVBord1	2.08	MVBord1	1.54	PNNaut1	1.31	PNNaut1	1.23
MVBord2	2.45	MVBind1	1.67	MVBcor1	1.32	MVBcor1	1.29
MVBaut2	2.72	MVBaut1	1.82	MVBord1	1.37	MVBord1	1.34
MVBord3	3.12	PNNord2	2.13	MVBind1	1.60	MVBind1	1.56
MVBaut3	3.71	MVBord2	2.19	MVBaut1	1.65	MVBaut1	1.61
MVBord4	4.08	PNNaut2	2.54	PNNord2	2.10	MVBord2	2.10
MVBcor2	4.21	MVBaut2	2.59	MVBord2	2.11	PNNord2	2.10
MVBind2	4.54	MVBord3	3.08	PNNaut2	2.54	MVBaut2	2.51
MVBaut4	4.83	PNNord3	3.11	MVBaut2	2.54	PNNaut2	2.54
MVBord5	5.03	MVBaut3	3.67	MVBord3	3.04	MVBord3	3.03
MVBaut5	6.00	PNNaut3	3.71	PNNord3	3.11	PNNord3	3.11
MVBcor3	7.02	MVBord4	4.05	MVBaut3	3.66	MVBaut3	3.64
MVBind3	7.61	PNNord4	4.06	PNNaut3	3.71	PNNaut3	3.71
PNNind3	7.66	PNNcor2	4.07	MVBord4	4.04	MVBord4	4.02
MVBcor4	10.00	MVBcor2	4.12	PNNord4	4.06	PNNord4	4.06
PNNcor4	10.06	PNNind2	4.47	PNNcor2	4.07	PNNcor2	4.07
MVBind4	10.80	MVBind2	4.48	MVBcor2	4.07	MVBcor2	4.07
PNNind4	10.90	MVBaut4	4.82	PNNind2	4.47	MVBind2	4.46
MVBcor5	13.00	PNNaut4	4.85	MVBind2	4.47	PNNind2	4.47
PNNcor5	13.09	PNNord5	5.00	MVBaut4	4.82	MVBaut4	4.82
MVBind5	14.00	MVBord5	5.01	PNNaut4	4.85	PNNaut4	4.85
PNNind5	14.12	MVBaut5	6.00	PNNord5	5.00	MVBord5	5.00
PNNcor1	InF	PNNaut5	6.00	MVBord5	5.01	PNNord5	5.00
PNNord1	InF	MVBcor3	7.01	MVBaut5	6.00	MVBaut5	6.00
PNNord2	InF	PNNcor3	7.04	PNNaut5	6.00	PNNaut5	6.00
PNNord3	InF	MVBind3	7.61	MVBcor3	7.01	MVBcor3	7.01
PNNord4	InF	PNNind3	7.62	PNNcor3	7.04	PNNcor3	7.04
PNNord5	InF	MVBcor4	10.00	MVBind3	7.61	MVBind3	7.60
PNNcor2	InF	PNNcor4	10.02	PNNind3	7.62	PNNind3	7.62
PNNcor3	InF	MVBind4	10.80	MVBcor4	10.00	MVBcor4	10.00
PNNaut1	InF	PNNind4	10.82	PNNcor4	10.02	PNNcor4	10.02
PNNaut2	InF	MVBcor5	13.00	MVBind4	10.80	MVBind4	10.80
PNNaut3	InF	PNNcor5	13.00	PNNind4	10.82	PNNind4	10.82
PNNaut4	InF	MVBind5	14.00	MVBcor5	13.00	MVBcor5	13.00
PNNaut5	InF	PNNind5	14.00	PNNcor5	13.00	PNNcor5	13.00
PNNind1	InF	PNNord1	InF	MVBind5	14.00	MVBind5	14.00
PNNind2	InF	PNNaut1	InF	PNNind5	14.00	PNNind5	14.00

From the Table above, a different preference in data sets as indicated by 'ord', 'cor', 'aut' and 'ind' is now observed. Because the co-registered and independent data set are not

penalized for only one look, they appear to have a relatively low time to Hostile ID if not required to take a second look. In general, for forced minimum looks greater than 1, both the autocorrelated and naturally ordered data tend to surface with better performance than the other two data sets that would require additional flight passes to collect data at non-consecutive aspect angles. The preferred forced look/data correlation assessment, also shows that taking more forced looks of autocorrelated or naturally ordered data may be preferred to fewer looks of co-registered or independent data, which may require more time to obtain a correct Hostile ID. This simple example using assumed times associated with each correlation structure helps to illustrate the utility of using a performance measure like TPR or it's reciprocal (mean looks/TP), to help gain insight and assess the utility of a Combat ID system, where the time associated with obtaining the data is incorporated. Thus, by showing TP as a rate, significantly more information is available to make decisions about a preferred fusion system, which is not included in a classical TP vs. FP ROC curve.

5.9 Potential Future Experiment Excursions

The potential for several experimental excursions is supported by the DCS data used within this chapter, including variations in the generation of sensor level data from the original 2-D radar imagery and investigation of other fusion methodologies. First, investigations could be performed to explore the sensor data parameters. Specifically, the HH and VV polarized data could be processed into features using both AFRL's HRR algorithm and Çetin's PBR HRR algorithm across numerous different templates, defined by the number of range bins used, number of angles included by each template or by

259

changing internal parameters specific to each of the HRR processing algorithms, etc. The mixed variable optimization formulation could then be employed to determine the best "Sensors" to use. This could be accomplished by including a constraint only allowing n of m total available sensors to be selected. One experiment may seek to determine the best algorithm and angular template for each polarization. Another experimental excursion could explore how the final fused system performs if aspect information is degraded. This could be accomplished by varying the number of templates searched from an approximate +/- 15 degree search of 3 templates to search 5, 7 or potentially all templates available for each target type. The search of all angular templates would represent a case of no usable aspect information. Out-of-library targets could also be introduced by using the 5 held out targets from the DCS radar collection (SA-8 TZM, BMP-1, BTR-70, SA-13, and SA-8 TEL).

Finally, different fusion methods could be assessed. Other methods could include the incorporation of different Boolean rules, other neural network methods, or other fusion methods. Simple modifications to the Boolean logic could require less conservative rules for making class declarations, by not requiring a majority vote for some labels, or increasing confidence prior to declarations by requiring a majority + n vote prior to making a class declaration. PNN fusion as presented in this chapter could also be modified by experimenting with use of a reduced data set for training that may help for generalization along with further experimentation with the spread of the basis functions, or changing the desired training target values. Another PNN fusion method may also use the posterior probability estimates from all 10 target types as PNN input to see if significant information was lost as the posterior probabilities for TOD, OH, and FN

260

were calculated for each sensor. Use of ten vs. three input posteriors may potentially yield better class estimates as more information about specific vehicle likelihoods is used as input to the PNN. Other Boolean fusion schemes may also be assessed, such as trying to determine the optimal Boolean logic, rather than using predetermined rules. For example, the ISOC method developed by Haspert (2000) may be modified to fit into the optimization framework. The optimal Boolean logic associated with all labels obtained up through time t could be determined using an ISOC optimization routine. Yet, because ISOC fusion requires cost information and assumes independent sensors, many tactical issues would need to be addressed. In addition, no straight forward algorithm appears readily available to determine the best Boolean logic, given the previous observation was a "ND" and only a limited data sample is available to assess the tuning of sensor thresholds to generate the "Non-declarations."

5.10 DCS Fusion Experiment Summary and Findings

Demonstration of the mixed variable mathematical optimization using this collected DCS radar data has resulted in several interesting findings. First, the fusion of two sensors with ~80% accuracy for "Hostile" and "Friendly" identification by a single-sensor and single-look of Test data were fused across sensors and through time. With a "Non-declaration" option, feasible Combat ID systems were then obtained with respect to the warfighter's operational constraints. These constraints included a maximum critical error less than 2%, which was met across many values of prior probabilities and across 1 to 5 minimum forced looks prior to making a declaration. The warfighter's preferences were incorporated as constraints in the process of optimizing TPR. Preferred fusion

methods were then determined without using explicit costs, allowing for "Non-declarations" and across time, where the number of looks required was used as a surrogate of time.

Sensitivity analysis revealed general regions of feasibility across the range of minimum forced looks, prior probabilities of H:F, the critical error constraint, and the required declaration rate constraint. Assessments were made using EOC Test data viewed at a depression angle of 10 degrees vs. the 6-8 degree data used for Training. This sensitivity analysis was performed across a full experimental design including 9 levels of priors, 4 levels of critical error, 3 levels of declaration rate and across the minimum forced looks by each fusion model. The sensitivity analysis facilitated the determination of the general boundaries where each fusion system may be preferred. This included showing areas where both systems were infeasible with low priors and few forced looks, and areas where both fusion methods achieved similar performance at higher ratios of priors with more forced looks. Influence of varying the two constraints, Π_1 and Π_3, was then more pronounced at these preference boundaries. Thus, from the sensitivity analysis, the operational environment defined by the prior probability of encountering a Hostile vs. Friendly target and the predetermined decision to use multiple forced looks appears to provide the greatest influence for fusion system feasibility. This has a good intuitive interpretation, where a Combat ID system operating in a Hostile target sparse environment is more likely to encounter a Friend or Neutral target, and should be required to take extra looks and gain a high level of confidence prior to labeling as a "Hostile." For Hostile target rich environments, the chance of encountering

262

a Friend or Neutral is much lower, so less confidence may be required before labeling as "Hostile" prior to making a subsequent shoot decision.

In general, the Majority Vote Boolean fusion was able to achieve feasible solutions across a larger percentage of all assessed conditions, as shown throughout the sensitivity analysis. The greater feasibility may be attributable to the optimization of the two sensors using four variable thresholds for the Boolean fusion vs. only allowing the PNN fusion to optimize over two continuous thresholds after fusion had occurred. The PNN fusion optimization was limited to two degrees of freedom, while the Boolean fusion used four degrees of freedom to tune each of the 2 sensors to perform a slightly different classification task to obtain the maximum TPR. In addition, even without a "Non-declaration" label generated by either sensor, the Boolean fusion rule still forced additional looks when the individual sensors were in conflict without a majority vote. Thus, in a Hostile rich environment with only one forced look, the MVB fusion had a lower maximum TPR compared to the PNN fusion. In this environment, the PNN fusion aggressively labeled most vehicles as "Hostile" on the first look. For the more difficult environments, the nature of the majority vote logic combined with the tuning of each sensor, allowed MVB fusion to obtain feasible solutions by taking additional looks. This increased feasibility was demonstrated across significantly more of the excursions using different prior ratios across a range of maximum critical errors and declaration rates, and using different numbers of minimum forced looks. The forcing of extra looks, rather than incorrectly generating some incorrect labels, was illustrated in the Hostile target sparse environments where feasible MVB fusion with only 3 or 4 forced looks, would be seen to take an average of 5 or more looks to obtain a correct Hostile target ID.

263

Assessment across the four levels of correlation in the data confirmed that in general less correlated data is preferred. For the more difficult cases with low ratios of priors, MVB fusion preference was for the two data sets which were independent across the two sensors for any given time t. This is indicated by the larger feasible region, obtained by the MVB fusion for the independent and autocorrelated data vs. the co-registered or naturally ordered data. This occurred predominately for data with prior ratios of less than 1:1 and for low numbers of minimum forced looks. These two data sets may be preferred by the Boolean fusion logic, since at any time t, the sensors collect data at different aspect angles. Then, if one image is labeled incorrectly, the majority vote can not be obtained on the first look and will force an additional look. Similarly, if a target is confused though multiple looks by one sensor, a final fusion output label other than "Non-declaration" will not be made until the majority vote is obtained.

A small illustrative example was presented to highlight the utility of using a measure such as TPR or the mean time to TP as a measure of performance for Combat ID systems. For the example in section 5.8, the preferred fusion models were first ordered by the mean number of minimum Looks to TP (reciprocal of max TPR), and shows a preference for the sensor data structures with less inherent correlation. Preference was also shown for the feasible fusion models based on a lower number of minimum looks. After penalizing the less correlated data structures to require more time per look, differences arose as to which data set and minimum number of forced looks may be preferred. These preferences were illustrated across the higher levels of H:F, where more of the systems were feasible. Thus, when designing a Combat ID system or trying to determine optimal CONOPS for flight passes, the mixed variable optimization provides a

means to compare and assess the value of obtaining data with potentially less inherent correlation. Less correlation may be obtained by altering the flight passes; yet, collecting data from another flight pass may take considerably more time. In the same time period, multiple consecutively ordered images may be taken, or if multiple ISR platforms are available, real-time fusion across a small formation may be highly desired. The Hostile target rich or target dense boundaries may also be determined, where collection of relatively independent data from multiple flight passes may be required to meet the desired constraints of a feasible ID system. Some of these differences are visible from the increased feasible solutions when comparing the naturally ordered data vs. the other three data sets with less inherent correlation.

Overall, not only were two fusion systems optimized across ROC and rejection thresholds, but this was accomplished through time and without the use of difficult to determine costs. These costs are associated with undesirable critical errors, potentially having grave consequence, non-critical error leading to sub-optimal sorties, and "Non-declarations" requiring additional dedicated ISR asset time before obtaining a final target label, and may be difficult to place in comparable units. This mixed variable optimization framework provided a means to assess Combat ID systems with desirable performance characteristics, without these cost estimates placed in equivalent units as required by many of the reviewed methods to assess classification systems using a minimum cost function.

VI. Contributions and Avenues for Future Research

This research was not intended to advocate use of one fusion method over another, but to facilitate the future assessment of ATR systems required fusion sensor data to obtain a desired level of confidence prior to making a declaration. These systems may always yield a "Non-declaration" output label and the subsequent sensor data collected and fused may be highly correlated. Further, it was highly desired to evaluate competing fusion methods without inclusion of explicit costs of misclassification. While all examples presented were focused on military ATR applications, it should be noted, this general framework for the evaluation of classification systems may be developed in a similar manner for other classification systems. Other areas employing ATR systems with the potential for fused sensors include the medical community for diagnosis, automatic system prognosis, financial forecasting, robotics, and environmental monitoring.

6.1 Contributions

Chapter 1 defined principal research areas and objectives. The contributions made by this research are presented in the context of these areas.

1. Comprehensive review of the literature as applicable to the investigation of assessing ATR systems with the fusion of correlated data and "Non-declarations"

2. Development of a mathematical programming formulation to assess and compare fusion systems without explicit misclassification costs and inclusive of temporal considerations and "Non-declarations"

3. Development of multivariate data generation for a synthetic classifier fusion-testing environment

4. Demonstration of the proposed mathematical programming formulation on various data sets

5. Empirical evidence for some general data correlation effects for ATR systems

6.1.1 Comprehensive Review of the Literature

A review of the literature was performed to determine the state-of-the-art methods to assess ATR fusion systems, given "Non-declarations," uncertain misclassification costs across known classes and "Non-declarations," and inclusive of temporal assessment. Some methods were found to assess performance inclusive of "Non-declarations." Yet, this was performed with either a predetermined level of rejection or through use of estimated misclassification costs. Review of the literature also identified potential feature generation techniques and general levels of expected correlation, although limited measured of correlation were reported in the open literature. Review of the literature also identified Boolean rules as a common fusion method to perform decision level fusion, while use of neural networks, was a common method for the fusion of feature level sensor data. Overall, review of the literature showed this specific investigation of sensor fusion for ATR with "Non-declarations" and correlated input data offers extensions from previously performed fusion research.

6.1.2 Mathematical Framework for the Evaluation of ATR

As a proposed research goal, the development of a ROC like measure of performance inclusive of "Non-declarations" and temporal assessment of identification

systems was developed. The projection of a set of ATR system ROC curves could then be plotted on a traditional 2D axis, or a 3D surface may be used to help show the trade-offs as rejection levels are varied. In each of these plots, feasible regions may be identified, along with an optimal operating point. The optimal point was determined from optimization of a mixed variable program, and the optimal thresholds associated with each ATR system were identified. Without using explicit cost, the optimization may be performed across an entire range of rejection thresholds, which could subsume those optimal rejection thresholds identified by rejection methods suggested by Chow (1970) or Fumera *et al.* (2000). This optimization strategy also included the "vertical" analysis of ATR system output labels, from which actionable decisions are made. Further constraints may be added to help in the design process of an ATR system. Finally, while developed using the *TP Rate* as the preferred objective function, the mathematical formulation may easily be modified to include alternative objective functions.

6.1.3 Multivariate Data Generation for a Synthetic Fusion Test Environment

As presented in Chapter 4, multivariate Gaussian data may be generated with desired correlation levels across features and through time using a VAR process. This data may then be used to represent features associated with different sensors or the output associated with an ATR system. Justification for use of a multivariate Gaussian representation of features derived from processed data, inclusive of linear mappings, was also presented. Use of this data then provided an efficient means to test fusion algorithms across a variety of data correlation structures. Different fusion experiments were then performed using these data generation techniques. This data generation also supported

recent fusion research with correlated input data as performed by Storm (2003), Clemans (2004), Leap (2004) and Mindrup (2005).

6.1.4 Implementation of the MVP Formulation to Assess Fusion Methods

Comparison of DCS radar data for 10 target types with fusion of two template based classifiers compared a Majority Vote Boolean fusion algorithm with use of a Probabilistic Neural Network (PNN) for fusion. An extended operating condition (EOC) test set used to compare fusion methods. While new to the ATR community, this data set was collected to support ATR research and has similar characteristics to the MSTAR data set, which has been used to support the research associated with over 150 published ATR related articles (Wise *et al.*, 2004). This new collection of DCS radar data includes a new variety of ground targets and offers polarimetric radar data collected for both HH and VV channels of X-band radar polarizations. Within the ground targets are likely friendly and neutral target types, with radar data collected on the HMMWV and M113 along with three different versions of Budget moving trucks. The data set also includes both the SCUD and SMERCH which offer two targets of the same relative large size. This provides a challenge for classifiers, with the potential confusion of target types if a feature relies on the relative size of a target. Thus, this unclassified data set may be used in a similar manner as the MSTAR data which has been used to support a significant portion of open-literature ATR research. In summary, the DCS radar data collection may likely continue to be used for significant future ATR and fusion research, with this effort being one of the preliminary investigations using this data.

269

6.1.5 Demonstrated Effects of Data Correlation for ATR

From use of the DCS radar data across four generated data sets several effects of sensor correlation were observed. These observations were made across systematic variations to the mathematical programming formulation as sensitivity analysis was performed. In general, lower correlation levels across sensors or through time may contribute to increased system performance as measured by the maximum *TPR* achieved. But, more significantly, the lower levels of correlation provided for a significant increase in feasible operating conditions. Thus, given a fusion algorithm such as the PNN or use of a Majority Vote with a predetermined number of minimum forced looks, the associated feasible operation of these fusion methods varied significantly across different variables. The largest differences in feasibility appeared associated with the ratio of Hostile to Friendly targets. Increases in feasible fusion models were also obtained as correlation was reduced from the naturally ordered data set with sensor data collected at the same time by two sensors with approximately 4 degrees of aspect angle between looks, to sensors generated to represent independent collection across the two sensors and within multiple looks by the same sensor through time. In addition, by using a measure of performance such as the *TPR* which incorporates time, an associated value could be placed on a time requirement to obtain independent looks that would yield operationally equivalent *TPR* for the naturally collected data. This type of analysis was briefly demonstrated, and may be of potential help for ATR design and ATR concept of operations (CONOPS) development. This may assist in determining what is ultimately preferred in an operational use of ATR, quickly collected data with lower single look

performance, or less dependent data, potentially collected across multiple flight passes, which requiring more time.

6.2 Future Research

The contributions of this research effort immediately suggest several promising areas for related research. Chapter 5 includes several related extensions that may be of interest using the DCS radar data collection. Development of a mixed variable programming formulation was presented in Chapter 3, but was solved using complete enumeration across the desired discrete variables and a grid of thresholds. The application of mixed variable programming algorithms, such as those presented by Abramson (2002), Audet and Dennis (2000) or Sriver (2004), may provide for a more efficient optimization. Yet, these techniques typically just search for a single optimal solution. Thus, associated algorithms to identify all feasible operating points may be desired to help assess ATR system robustness. In addition, to support new sensor fusion research, the following two general areas are presented and outlined below.

6.2.1 Potential Sensor Saliency Research

Since multi-layer perceptron (MLP) ANNs, time delayed neural networks (TDNNs) and recurrent neural networks (RNNs) have all been used to fuse correlated input features with successful implementation of saliency screening (Laine and Bauer, 2003; Laine *et al*., 2002; Greene, 1998), an extension of saliency screening may be developed to measure the relative contribution by each sensor. This saliency investigation would be performed for feature level fusion of an object, with features from

multiple sensors fused to generate class estimates. The goal would be to gain insight as to which of the sensors provide salient information to the neural fusion model, and if using a TDNN or RNN, relative temporal saliency for information associated with re-looks could be evaluated. An initial look at contributions made by different sensors under a designed environment with injected noise is presented by Dasarathy (2000a) in which Case-Based Reasoning (CBR) is used to determine output class labels based on the minimum dissimilarity of an observation from known samples. A similar experiment could be designed building on the current of neural network saliency research.

For instance, the relative saliency of a group of features (sensor A) could be compared to another group of features (sensor B) as a measure of the relative output influence by each sensor. Either weight based saliency measures, such as the signal-to-noise ratio (SNR) (Bauer *et al.,* 2000) or performance based saliency methods, such as sensitivity based pruning (SBP) (Moody, 1998) may be used to obtain a measure of the relative value of a sensor's input or potentially the value of additional looks obtained by a sensor. Unlike previously applied input feature saliency measures, sensor saliency measures would need to consider a set of features associated with a sensor. Under a weight based approach, using the sum of SNR values for each feature from a sensor may be a first approach. Likewise, an output based measure such as SBP could use the relative change in the model's output when all values associated with a given sensor are set to mean values to assess the relative impact of information provided by each sensor.

6.2.2 Potential Research for "Non-declarations" at Various Fusion Levels

As stated in Chapter 2, research has shown MLP ANNs are capable of performing any mapping to a desired degree of accuracy (Hornik *et al.*, 1989, 1990) and a well-trained ANN yields *a posterior* probability estimates of class membership (Ruck *et al.,* 1990; Wan 1990). Thus, with more information available from input features representative of sensor features or estimated class probabilities, one research question is whether the search for optimal decision thresholds of the continuous valued neural network output space generated from an appropriate "one big net" fusion model may be superior to Boolean sensor output fusion. The experiment presented in Chapter 5 using the DCS radar data compared a PNN fusion approach to a predetermined Boolean logic, with the majority vote Boolean method preferred in many target sparse environments. This preference was achieved by obtaining feasible solutions when the PNN fusion remained infeasible. Determining whether this increased feasibility is associated with the Boolean logic or the increased degrees of freedom used to optimize across 4 variable ROC and rejection thresholds vs. 2 variable thresholds used by the PNN fusion is of interest.

With "Non-declarations" required, the one-big-net PNN fusion approach was only able to generate a "Non-declaration" at the end of the fusion process. In contrast, the Majority Vote Boolean fusion generated "Non-declaration" labels for both the input to the fusion rule by individual sensors, and as output from the fusion rule. Overall, new research may look to provide a theoretical basis to explain why preprocessing a sensor's output data may be preferred, prior to fusion by any method. As was observed from analysis of the optimal thresholds, the best Boolean fusion was often obtained by tuning

each senor to perform a slightly different classification task. Experimental approaches could attempt to generate "tuned" sensor data more similar to the label generation used by the Boolean fusion, but may attempt to retain a continuous value associated with different consolidated classes, inclusive of generating a "Non-declaration" input value. In performing this research, a defendable answer to whether "Non-declarations" should be performed prior to sensor fusion or post sensor fusion under different input assumptions is of interest.

6.3 Final Conclusions

Overall, the research contained within this document extends the research found within the open literature. As desired, a ROC-like performance measure was developed inclusive of temporal assessment for ATR systems. The ROC-like nature simply identifies those feasible points on a ROC curve which meet the warfighter operational constraints. These operational constraints include the analysis of Critical and Non-critical errors via vertical analysis and the assessment of "Non-declarations." Hopefully, this mathematical optimization may be a significant aid for the evaluation and comparison of competing ATR systems, which are required to fuse data to reach desired levels of correct class declarations. The proposed methodology goes beyond the traditional ATR system evaluation methods and determines the preferred ATR operating thresholds and other system parameters without use of explicit costs. This measure can then be used to help determine the relative value of obtaining correlated data quickly or of obtaining less correlated data across a longer time period. In summary, the optimization methodology incorporates a flexible framework to establish a decision

maker's primary objective, subject to constraints, and does so across both the warfighter's "vertical" view of declared targets and the engineer's "horizontal" view of actual types of objects classified.

Appendix A. DCS Experiment Figures and Tables

The following figures and tables provide more detailed information with respect to the experiment using the DCS radar data.

A.1 Sensor Posterior Probabilities for Training and Test Data by Aspect Angle

Figure A.1 SCUD Posterior Probabilities by Sensor for Training & Test Data

Figure A.2 SMERCH Posterior Probabilities by Sensor for Training &Test Data

Figure A.3 SA-6 Radar Posterior Probabilities by Sensor for Training &Test Data

Figure A.4 Med Truck Posterior Probabilities by Sensor for Training &Test Data
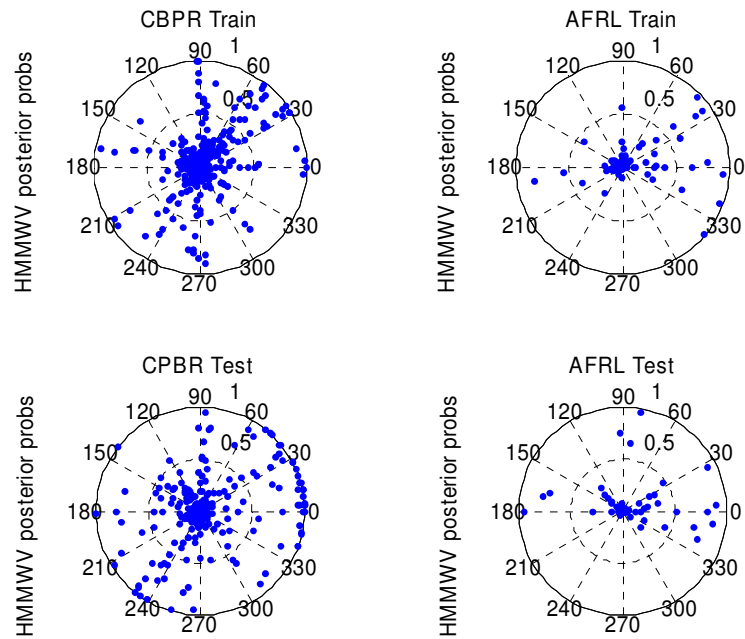
Figure A.5 HMMWV Posterior Probabilities by Sensor for Training &Test Data
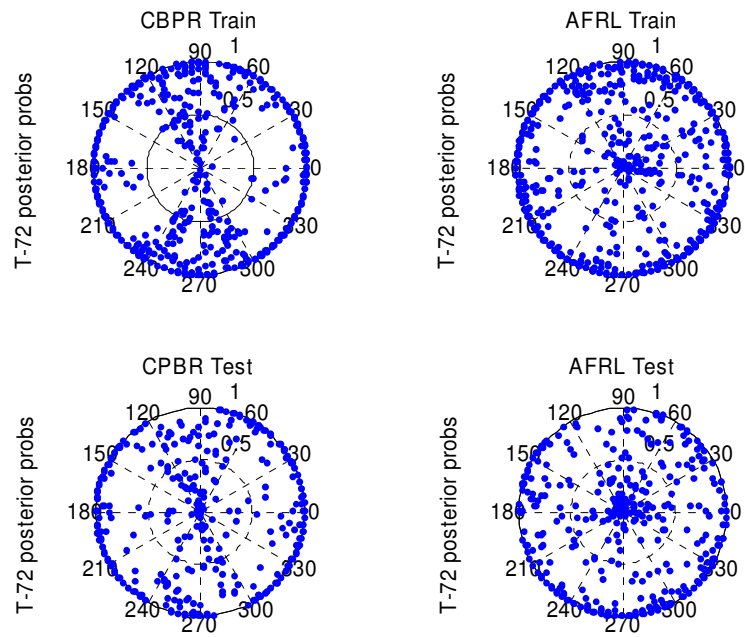
Figure A.6 T-72 Posterior Probabilities by Sensor for Training &Test Data
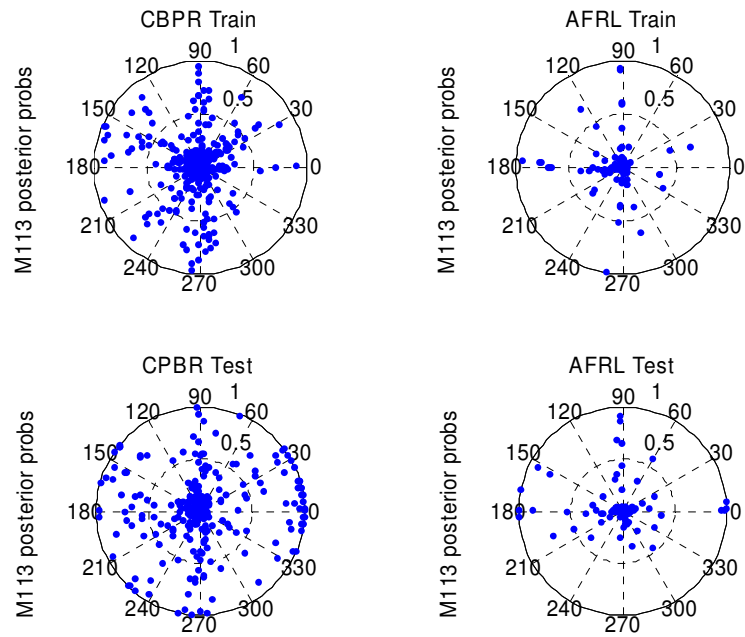
Figure A.7 M113 Posterior Probabilities by Sensor for Training &Test Data
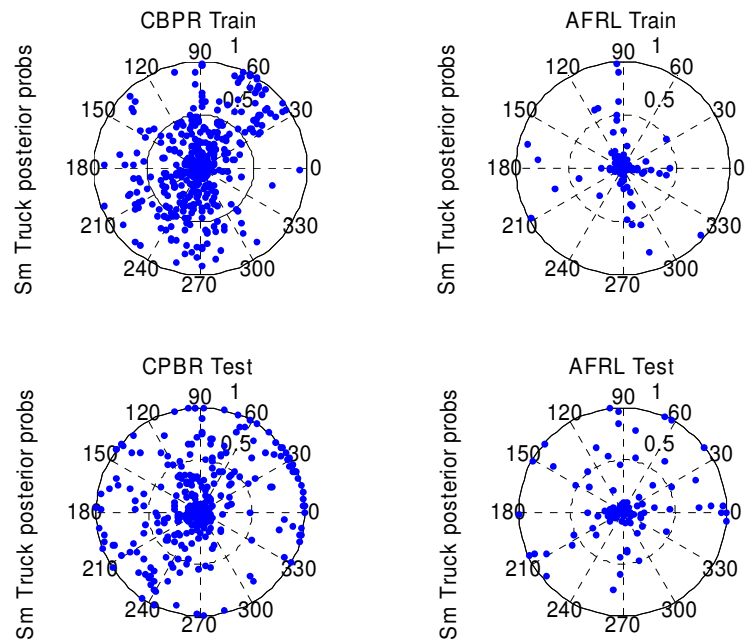
Figure A.8 Small Truck Posterior Probabilities by Sensor for Training &Test Data

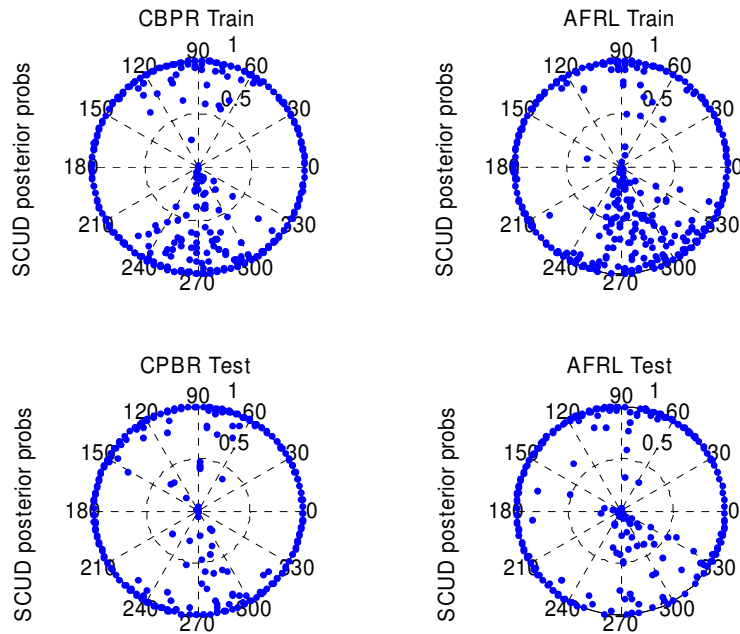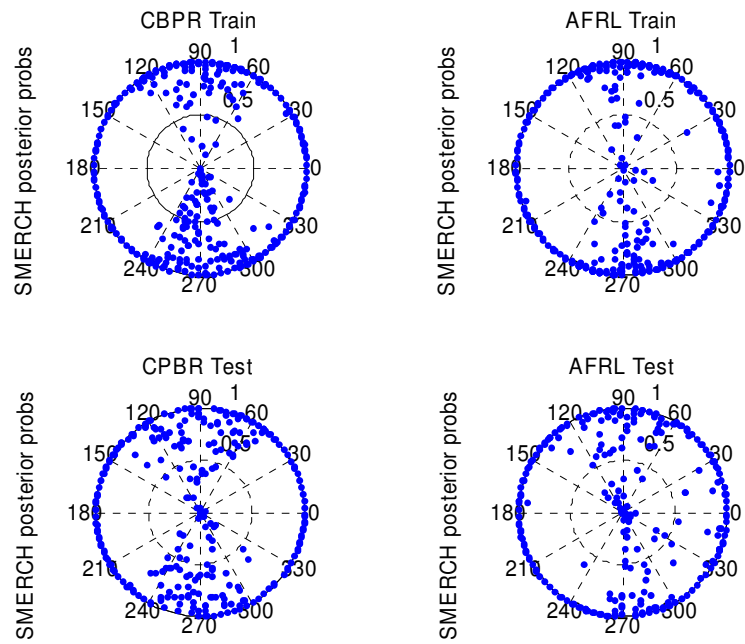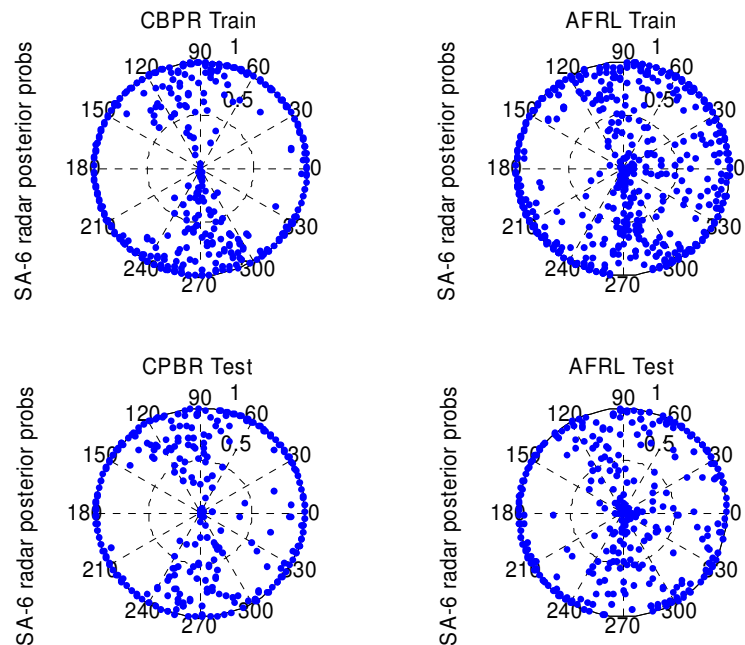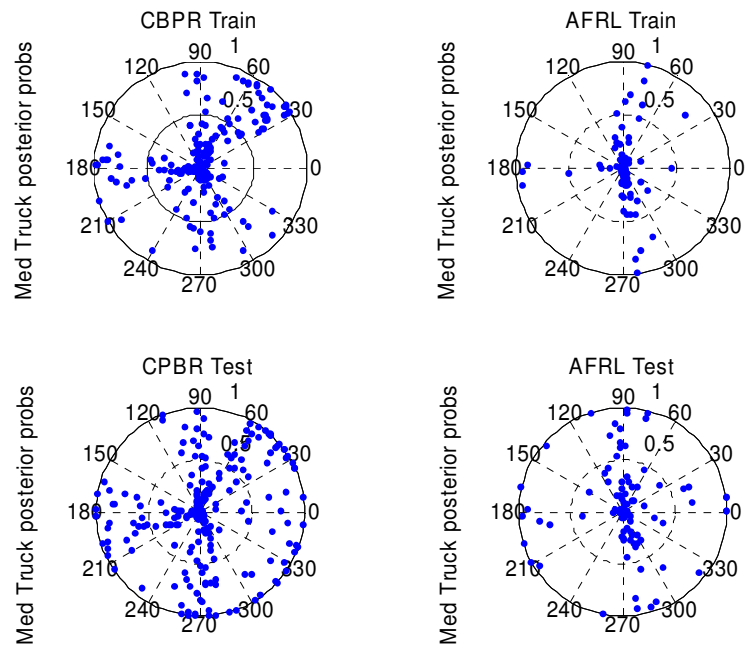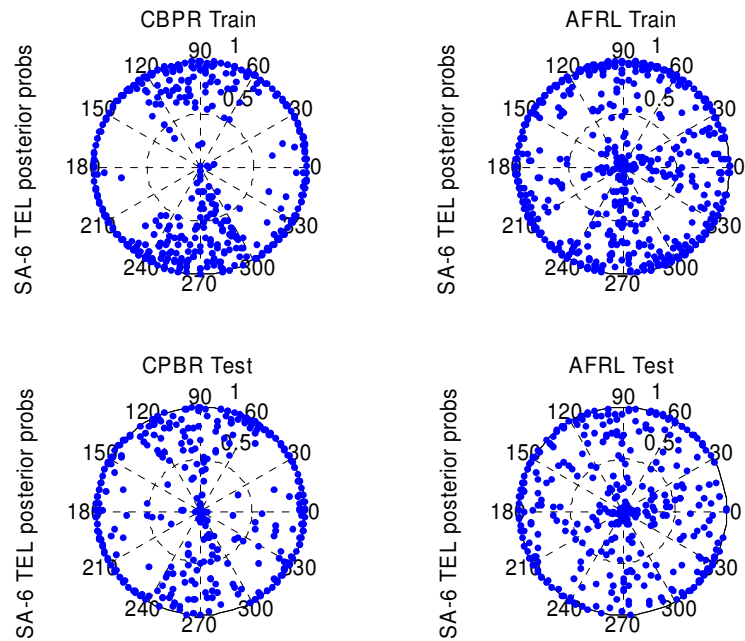**Figure A.9 SA-6 Posterior Probabilities by Sensor for Training &Test Data**
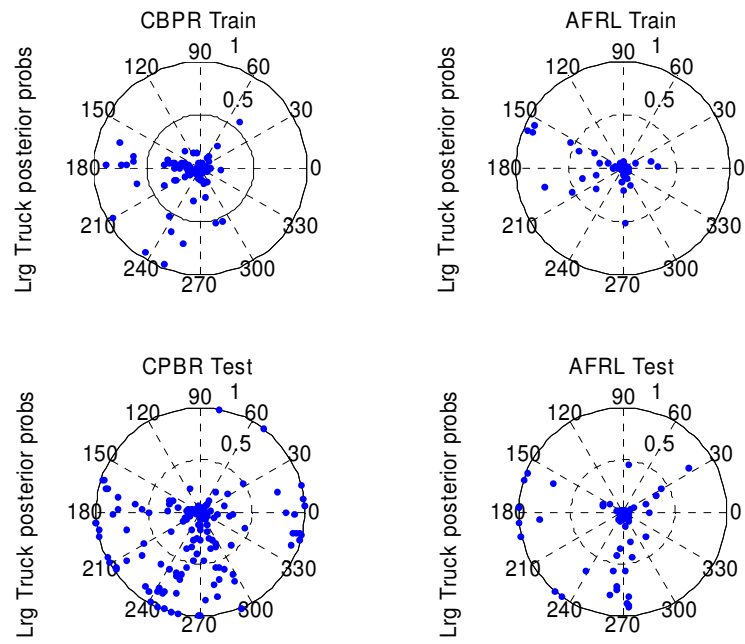
Figure A.10 Large Truck Posterior Probabilities by Sensor for Training &Test Data

Table A.1 Sample Sensor Performance by Target Type using Training and Test Data for

$\theta_{REJ} = 0.0$ Centered at θ_{ROC} 0.5 (No Rejection Option)

Type	Label	**Sensor A Training Data**			**Sensor B Training Data**		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	4%	96%	0%	7%	93%	0%
SMERCH	OH	7%	94%	0%	3%	97%	0%
SA-Radar	OH	6%	94%	0%	15%	85%	0%
T-72	OH	8%	92%	0%	13%	87%	0%
SA-6 TEL	OH	5%	95%	0%	15%	86%	0%
Med Truck	FN	91%	9%	0%	98%	2%	0%
HMMWV	FN	90%	10%	0%	98%	2%	0%
M113	FN	90%	10%	0%	98%	2%	0%
Sm Truck	FN	82%	18%	0%	98%	2%	0%
Lg Truck	FN	98%	2%	0%	99%	1%	0%
		mean True Class		92.2%	mean True Class		93.7%
		mean False Class		7.9%	mean False Class		6.3%
		mean rejection		0.0%	mean rejection		0.0%

Type	Label	**Sensor A Test Data**			**Sensor B Test Data**		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	11%	89%	0%	21%	79%	0%
SMERCH	OH	11%	90%	0%	11%	90%	0%
SA-Radar	OH	12%	88%	0%	27%	74%	0%
T-72	OH	21%	79%	0%	35%	65%	0%
SA-6 TEL	OH	14%	86%	0%	32%	68%	0%
Med Truck	FN	70%	30%	0%	92%	8%	0%
HMMWV	FN	81%	19%	0%	97%	3%	0%
M113	FN	76%	24%	0%	96%	4%	0%
Sm Truck	FN	66%	34%	0%	93%	7%	0%
Lg Truck	FN	84%	16%	0%	96%	5%	0%
		mean True Class		80.8%	mean True Class		84.8%
		mean False Class		19.2%	mean False Class		15.2%
		mean rejection		0.0%	mean rejection		0.0%

Table A.2 Sample Sensor Performance by Target Type using Training and Test Data for $\theta_{REJ} = 0.4$ Centered at θ_{ROC} 0.5 (Rejection Occurs if $0.30 < ppH < 0.70$)

Type	Label	Sensor A Training Data			Sensor B Training Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	3%	93%	4%	4%	89%	7%
SMERCH	OH	4%	90%	6%	2%	95%	3%
SA-Radar	OH	3%	89%	8%	9%	77%	14%
T-72	OH	4%	85%	12%	8%	82%	10%
SA-6 TEL	OH	3%	89%	8%	10%	79%	12%
Med Truck	FN	86%	7%	7%	96%	1%	2%
HMMWV	FN	85%	6%	9%	97%	1%	2%
M113	FN	84%	4%	12%	96%	1%	3%
Sm Truck	FN	69%	10%	21%	96%	1%	3%
Lg Truck	FN	97%	1%	2%	99%	1%	1%

		mean True Class | dec		95.1%	mean True Class | dec		96.1%
		mean False Class | dec		4.9%	mean False Class | dec		3.9%
		mean rejection		8.7%	mean rejection		5.6%

Type	Label	Sensor A Test Data			Sensor B Test Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	9%	88%	4%	18%	77%	5%
SMERCH	OH	7%	82%	11%	9%	85%	7%
SA-Radar	OH	6%	81%	13%	18%	67%	15%
T-72	OH	14%	72%	14%	29%	57%	14%
SA-6 TEL	OH	8%	79%	13%	24%	59%	17%
Med Truck	FN	65%	22%	13%	89%	6%	5%
HMMWV	FN	74%	16%	11%	96%	3%	2%
M113	FN	69%	17%	14%	93%	3%	4%
Sm Truck	FN	56%	26%	18%	90%	5%	5%
Lg Truck	FN	80%	12%	8%	94%	4%	3%

		mean True Class | dec		84.4%	mean True Class | dec		87.2%
		mean False Class | dec		15.6%	mean False Class | dec		12.8%
		mean rejection		11.9%	mean rejection		7.6%

Table A.3 Sample Sensor Performance by Target Type using Training and Test Data for $\theta_{REJ} = 0.8$ Centered at θ_{ROC} 0.5 (Rejection Occurs if $0.10 < ppH < 0.90$)

Type	Label	Sensor A Training Data			Sensor B Training Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	1%	88%	11%	1%	84%	15%
SMERCH	OH	1%	83%	16%	0%	91%	9%
SA-Radar	OH	1%	81%	18%	4%	65%	31%
T-72	OH	1%	74%	25%	3%	70%	27%
SA-6 TEL	OH	1%	77%	23%	4%	67%	29%
Med Truck	FN	79%	3%	18%	93%	1%	6%
HMMWV	FN	70%	2%	28%	95%	1%	5%
M113	FN	69%	1%	30%	93%	1%	6%
Sm Truck	FN	55%	4%	41%	93%	1%	7%
Lg Truck	FN	93%	0%	7%	97%	0%	3%

	mean True Class | dec	97.9%	mean True Class | dec	98.3%
	mean False Class | dec	2.1%	mean False Class | dec	1.7%
	mean rejection	21.7%	mean rejection	13.7%

Type	Label	Sensor A Test Data			Sensor B Test Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	7%	84%	9%	16%	74%	11%
SMERCH	OH	5%	70%	26%	6%	79%	16%
SA-Radar	OH	4%	70%	27%	11%	57%	31%
T-72	OH	8%	58%	34%	17%	45%	38%
SA-6 TEL	OH	5%	65%	30%	15%	43%	42%
Med Truck	FN	54%	15%	31%	83%	5%	12%
HMMWV	FN	58%	11%	31%	93%	1%	7%
M113	FN	52%	11%	37%	89%	2%	9%
Sm Truck	FN	39%	17%	44%	85%	3%	12%
Lg Truck	FN	73%	8%	19%	92%	2%	7%

	mean True Class | dec	87.3%	mean True Class | dec	90.5%
	mean False Class | dec	12.7%	mean False Class | dec	9.5%
	mean rejection	28.7%	mean rejection	18.3%

Table A.4 Sample Sensor Performance by Target Type using Training and Test Data for

$\theta_{REJ} = 0.98$ Centered at θ_{ROC} 0.5 (Rejection Occurs if $0.01 < ppH < 0.99$)

Type	Label	Sensor A Training Data			Sensor B Training Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	0%	82%	19%	0%	74%	26%
SMERCH	OH	0%	75%	24%	0%	82%	18%
SA-Radar	OH	0%	72%	28%	1%	47%	52%
T-72	OH	0%	62%	38%	0%	50%	50%
SA-6 TEL	OH	0%	64%	36%	0%	45%	55%
Med Truck	FN	67%	1%	32%	86%	0%	14%
HMMWV	FN	41%	1%	58%	86%	0%	14%
M113	FN	42%	0%	58%	83%	0%	17%
Sm Truck	FN	34%	1%	66%	82%	0%	18%
Lg Truck	FN	86%	0%	14%	93%	0%	8%
		mean True Class | dec		99.6%	mean True Class | dec		99.8%
		mean False Class | dec		0.4%	mean False Class | dec		0.2%
		mean rejection		37.2%	mean rejection		27.0%

Type	Label	Sensor A Test Data			Sensor B Test Data		
		"F"	"H"	% Rej	"F"	"H"	% Rej
SCUD	TOD	6%	77%	17%	12%	66%	22%
SMERCH	OH	3%	61%	37%	3%	67%	31%
SA-Radar	OH	2%	63%	36%	4%	39%	58%
T-72	OH	4%	47%	49%	8%	24%	68%
SA-6 TEL	OH	2%	50%	48%	6%	29%	65%
Med Truck	FN	45%	10%	45%	74%	2%	25%
HMMWV	FN	26%	7%	68%	86%	0%	14%
M113	FN	21%	6%	73%	76%	1%	23%
Sm Truck	FN	19%	10%	71%	74%	2%	24%
Lg Truck	FN	63%	5%	32%	84%	1%	15%
		mean True Class | dec		89.7%	mean True Class | dec		94.1%
		mean False Class | dec		10.3%	mean False Class | dec		5.9%
		mean rejection		47.6%	mean rejection		34.4%

Appendix B. Matlab Code

Appendix B contains some of the code used for the analysis presented within this

document. The first section includes the specific procedures used to process the 2D DCS

SAR image chips into HRR 1D range profiles.

B.1 Matlab Code used to Process DCS Data into HRR Radar Profiles

DCS_proc1.m

```
function [void] = DCS_proc1(flight_ID)
% File collects info from Phoenix header of DCS target data for 1 pass
%  Code modification by T. Laine to read data files Oct 04
%  Initial code generated by Tim Albrecht Fall 04 to read
%  MSTAR SAR chips with Phoenix header files
tic;

% DATA SOURCE: SAR target chips taken from DCS(Public) Targets data DVD,
% containing 15 ground targets in stationary positions imaged by spot SAR

% form Taylor window, will be used on all chips, so do calculations outside
% of loops
w1 = taylorWin(200,5,35);
w2 = w1;
w = w2*w1.';

target_str = ['C:\Documents and Settings\tlaine\My Documents\DCS data\' flight_ID '\Chips\'];

% list the SAR chip files in the current target directory
% (0) performs AFRL/SN MSTAR to HRR conversion as baseline reconstruction
%     technique, then performs the following reconstruction steps adapted
%     from Cetin's dissertation
file_list = dir(target_str);
for file_num = 3:size(file_list,1)                    % file "1" == "."  file "2" == ".."
%    disp(['processing chip #' num2str(file_num-2) ' of ' num2str(size(file_list,1)-2)]);
      file_str = [target_str, file_list(file_num).name];  % sets file_str as the file name indexed as
     % disp([file_str ' file #' num2str(file_num-2)]);

     outStruct = struct('type', [], 'serialNum', [], 'aspect', [], ...
         'rangeProfile', [], 'normProfile', [], 'reconProfile', [], ...
         'hrrProfile', []);

% read file
% (1) read in the chip header, magnitude and phase information
     fid = fopen(file_str, 'r', 'ieee-be');
     % read Phoenix header from chip file
     i=1;
     headerLine = fgetl(fid);
     while(~(strcmpi(headerLine,'[EndofPhoenixHeader]')))
        [field{i}, value{i}] = strtok(headerLine, '=');
```

```matlab
        i = i + 1;
        headerLine = fgetl(fid);
    end
    d = cellfun('isempty', value);
    field = field(~d);
    value = value(~d);
    value = cellstr(strjust(strvcat(strrep(value, '=', '')),'left'));
    numericValue = str2double(value);
    numericInd = find(~isnan(numericValue));

    for ii = 1:length(numericInd)
        value{numericInd(ii)} = numericValue(numericInd(ii));
    end

    [nr, nc] = size(value);
    nfields = max(nr,nc);

% Don't try to read these 6 fields as they get rejected as Matlab fields
%       field(82) == 'AircraftLocationX-ECEF'
%       field(83) == 'AircraftLocationY-ECEF'
%       field(84) == 'AircraftLocationZ-ECEF'
%       field(85) == 'AircraftVelocityX-ECEF'
%       field(86) == 'AircraftVelocityY-ECEF'
%       field(87) == 'AircraftVelocityZ-ECEF'
%  Thus, load header as 2 separate header files

    header1 = cell2struct(value(1:81), field(1:81), 1);
    header2 = cell2struct(value(88:nfields), field(88:nfields), 1);

    % get chip information
    rows = header1.NumberOfRows;
    cols = header1.NumberOfColumns;

    % read mag and phase blocks
    [mag,count] = fread(fid,[cols,rows],'float32');
    if (count ~= (rows*cols))
      error('Error reading the magnitude data');
    end
    [phase, count] = fread(fid,[cols,rows],'float32');
    if (count ~= (rows*cols))
      error('Error reading the phase data');
    end

    Aspect = header2.TargetAz;
    MDep = header2.MeasuredDepression;
    MDep=MDep(:,1:6);
    MDep=str2num(MDep);
    TargetType = header2.TargetType;
    TargetPos = header2.TargetPositionNumber;
    polar_str = header1.Polarization;

    % (2) create the baseline, complex chip (256 x 256 pixels)
    % form complex chip
    chip = mag.* exp(j.*phase);
    chip = flipud(chip.'); % range increases with increasing range bin
    % close file
    fclose(fid);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% AFRL/SN MSTAR to HRR steps %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MSTAR parameters
rangePixelSpacing= 0.202148;  %  0.2032102m == 0.6667 ft
xrangePixelSpacing= 0.203125; %  0.2032102m == 0.6667 ft
nBar = 5;
SLL = 35;
BEF = 1.184; % bandwidth expansion factor
rangeResolution = .3047;    % .3047m == 1 ft
xrangeResolution = .3047;   % .3047m == 1 ft
rOsmpl = rangeResolution/rangePixelSpacing;
xrOsmpl = xrangeResolution/xrangePixelSpacing;
fullSceneFftSize = [2042 1832];

% remove weighting and oversample
orgChip = RemoveTaylor(chip, nBar, SLL, BEF, rOsmpl, xrOsmpl, ...
   fullSceneFftSize(1), fullSceneFftSize(2));
[orgRow, orgCol] = size(orgChip);
hrrRangeResolution = rangeResolution/BEF;
hrrRangePixelSpacing = rangePixelSpacing*rows/orgRow;

% apply range weighting and 2x oversample
rngWgts = repmat(taylorWin(orgRow, 6, 40), 1, orgCol);
numRngSmp = round(2/1.25*orgRow);   % BEF = 1.25 for Taylor nbar = 6, SLL = 40
phaseHistory = fftshift(ifft2(orgChip));
osChip = fft(fft(phaseHistory.*rngWgts, numRngSmp),[],2);
[osRow, osCol] = size(osChip);
hrrRangeResolution = hrrRangeResolution*1.25;
hrrRangePixelSpacing = hrrRangePixelSpacing*orgRow/osRow;

% covert to range/angle domain without segmenting target
hrrVsAspect2 = ifft(osChip,[],2);

% form profile (detect, normalize, transform and average) for
% non-segmented (non-masked) version
hrrVsAspect2 = abs(hrrVsAspect2).^2;
for kk = 1:osCol
   hrrVsAspect2(:,kk) = sqrt(osRow)*hrrVsAspect2(:,kk)/norm(hrrVsAspect2(:,kk));
end
% hrrVsAspect2 = hrrVsAspect2.^.8; % power transform .2
%hrrVsAspect = 10*log10(hrrVsAspect); % dB
hrrProfile2 = mean(hrrVsAspect2,2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% perform chip transforms (Albrecht/Cetin) %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% (3) take the 2-D FFT of the chip
   chip_fft = fft2(chip);

% (4) shift smaller freq to center
   chip_fft_shift = fftshift(chip_fft);
   % chip_mag = abs(chip_fft_shift);
```

```
% (5) crop a 28 pixel wide ban of zero-padding (200 x 200 pixels)
    % chip_cropped_mag = chip_mag(29:228,29:228);
    chip_cropped_fft_shift = chip_fft_shift(29:228,29:228);

% (6) remove the Taylor windowing that was performed in the MSTAR
%     collection. 100 coefficient, 35 dB sidelobe suppresion, n-bar of 4,
%     yielding an unwindowed phase history of the chip (100 x 100 pixels)

    % remove Taylor windowing
    chip_unwinPhaseHist = chip_cropped_fft_shift./w;

% (7) take the 1-D FFT to get the complex range profiles (200 x 200 pixels)
    % form complex range profiles
    rngProfiles = fft(chip_unwinPhaseHist,[],1);

% (8) form the complex modulus (mag) of the complex range profiles, take
%     the mean, and normalize using the inf norm
    % complex modulus of complex range profiles, then mean
    rngProfiles_mag = abs(rngProfiles);
    meanProfile = mean(rngProfiles_mag,2);

    % populate outStruct
    outStruct.type = upper(strtok(strrep(header2.TargetType,'_',' ')));
    if( isnumeric(header2.TargetSerialNum) )
      header2.TargetSerialNum = num2str(header2.TargetSerialNum);
    end

    outStruct.serialNum = header2.TargetSerialNum;
    outStruct.aspect = header2.TargetAz;
    outStruct.rangeProfile = meanProfile; % (200 x 1 vector)
    outStruct.hrrProfile = hrrProfile2; % (322 x 1 vector)

    % give unique name to the newly formed structured array
    name_str = outStruct.type;
    aspect_str = int2str(round(outStruct.aspect));
    position_str = num2str(TargetPos);

% (9) save the range profiles according to aspect angle
    file_str = ['T',position_str,'_',polar_str,'_', aspect_str];
    magic_str = [file_str,' = outStruct;'];
    eval (magic_str);
  end % end chip files in directory loop

% save the range profiles according to aspect angle
for targetID = 1:15
  dname = ([flight_ID, '_T', num2str(targetID)]);
  dsave =(['T', num2str(targetID), '_*']);
  save(dname, dsave);
end

tx=toc;
disp(['flight ' flight_ID ', time to evaluate = ' num2str(tx) ' seconds'])
clear
```

DCS_proc2.m

```
function [void] = DCS_proc2(flight_ID)
% Original code by Tim Albrecht
% AFIT/ENS
% HMM Fusion Project,  Fall 04
%  Minor modification by T. Laine to act as a function call Oct 04
%  applied for DCS data

% this script performs the following batch operations on MSTAR SAR
% target chips (training data):
% (1) read in the target data created in 'batch_trn1.m'
% (2) normalize the range profiles across aspect angle across all targets
% (3) save normalized profiles to structured arrays

tic;
%  Read in all data chips associated with one flight
for tt = 1:15
   load_str = ([flight_ID, '_T', num2str(tt)]);
   load(load_str);
end
data_list = whos;
num_profiles = size(data_list,1) - 3
                              % there are 3 non-profile
                              % related variables in the
                              % workspace. they occur at the
                              % end of the list of variables
                              % so we decrement our list by
                              % 3 to avoid indexing into
                              % non-profile related data
                              % structures
   % find max val by searching through the target records
   max_val = 0;
   for i = 1:num_profiles
      name_str = data_list(i).name;
      data_str = '.rangeProfile';
      temp_profile = eval([name_str data_str]);

      if norm(temp_profile,inf) >= max_val
         max_val = norm(temp_profile,inf);
      end
   end

   % normalize by dividing through by the max_val
   for i = 1:num_profiles
      name_str = data_list(i).name;
      data_str = '.rangeProfile';
      temp_profile = eval([name_str data_str]);

      temp_profile = temp_profile./max_val;
      data_str = '.normProfile';
      eval([name_str data_str ' = temp_profile;'])
   end

   % save data
   for tt = 1:15
      save_str = ([flight_ID '_T' num2str(tt)]);
      save2_str = (['T' num2str(tt) '_*']);
```

290

```
        save(save_str, save2_str);
    end
    ttp=toc;
    disp(['Processed Phase 2 of 4 for FP ' flight_ID ' in ' num2str(ttp) ' seconds'])
    % remove variables from workspace before bringing in the next target
    clear all
```

DCS_proc3.m
```
function [void] = DCS_proc3(flight_ID);
% clear all;
% flight_ID = ('FP0110')

% Original code by Tim Albrecht
% AFIT/ENS
% HMM Fusion Project,  Fall 04
% Modified by T. Laine Oct 04 to process DCS instead of MSTAR data

% this script performs the following batch operations on DCS SAR
% target chips (training data):
% (1) reads transformed and normalized range profiles from 'DCS_proc2.m'
% (2) performs Cetin's point-based reconstruction algorithm
% (3) saves reconstructed range profile to structured array

tic;
%  Read in all data chips associated with one flight
for tt = 1:15
    load_str = ([flight_ID, '_T', num2str(tt)]);
    load(load_str);

    data_list = whos;
    num_profiles = size(data_list,1) - 3;
    disp([flight_ID ', target #' num2str(tt) ', ' num2str(num_profiles) ' looks'])
                                % there are 3 non-profile
                                % related variables in the
                                % workspace. they occur at the
                                % end of the list of variables
                                % so we decrement our list by
                                % 3 to avoid indexing into
                                % non-profile related data
                                % structures

    % cetin point-enhancement reconstruction parameters
    lambdasq = 0;      % region-based (smoothing) regularization parameter
    lambdasq2 = 20;    % point-based (energy) regularization parameter
    gamma = 1e-3;      % stopping criterion (try e.g. 10^{-3})
    type = 'LP';       % type of potential function used in prior (use 'LP'
                % for l_p-norms)
    p = 0.1;          % determines shape of the lk-norm prior
                % p=2 ==> Gaussian prior, Tikhonov-type
                % p=1 ==> Laplacian pror, Total variation-type
    N3 = 30;                      % length of data matrix y
                % (used just for initialization)

    for i = 1:num_profiles
        name_str = data_list(i).name;
        data_str = '.normProfile';
```

```matlab
        temp_profile = eval([name_str data_str]);

        % begin Cetin code (point-enhanced reconstruction)
        N1=size(temp_profile,1);
        % N3=30;

        q = temp_profile;

        %dftmtx.m should be in the MATLAB signal processing toolbox.
        %Let me know if you do not have it.
        %Actually, let me send it anyway.
        F=dftmtx(N1);
        F=F(1:N3,:);

        %target phase history
        h=F*q;

        %addition of observation noise
        % hn=h+0.1*(randn(N3,1)+sqrt(-1)*randn(N3,1));

        %conventional HRR profile reconstruction
        q_conv = F'*h/N3; % q_conv=F'*hn/N3;

        Fthn=F'*h; % Fthn=F'*hn;
        FF=F'*F;
        q_point_rec=hrr_point_rec(Fthn,FF,lambdasq,lambdasq2,gamma,type,p,N3);

        data_str = '.reconProfile';
        eval([name_str data_str ' = abs(q_point_rec);'])
    end % end profile loop

    % save data
        save_str = ([flight_ID '_T' num2str(tt)]);
        save2_str = (['T' num2str(tt) '_*']);
        save(save_str, save2_str);

    % remove variables from workspace before bringing in the next target
    % type
    data_list = whos;
    for i = 1:size(data_list,1)
        if strcmp(data_list(i).name,'path_str')
        elseif strcmp(data_list(i).name,'target_type')
        elseif strcmp(data_list(i).name,'data_list')
        elseif strcmp(data_list(i).name,'i')
        elseif strcmp(data_list(i).name,'tt')
        elseif strcmp(data_list(i).name,'flight_ID')
        elseif strcmp(data_list(i).name,'tic')
        else clear(data_list(i).name)
        end
    end
    clear data_list i
end % end target type loop

    ttp=toc;
    disp(['Processed Phase 3 of 4 for FP ' flight_ID ' in ' num2str(ttp) ' seconds'])
    % remove variables from workspace before bringing in the next target
    clear all
```

292

Appendix C. Glossary of Acronyms and Abbreviations.

ACC	Air Combat Command
AF	Air Force
AFRL	Air Force Research Laboratory
AGRI	Air-to-Ground Radar Imaging
ANN	Artificial Neural Network
AR	Autoregressive
ATO	Air Tasking Order
ATR	Automatic Target Recognition
AUC	Area Under the Curve
BDA	Battle Damage Assessment
CA	Classification Accuracy
CBR	Case Based Reasoning
CC&D	Camouflage, Concealment and Deception
CI	Confidence Interval
CID	Combat Identification
COMINT	Communications Intelligence
COMPASE	Comprehensive ATR Scientific Evaluation
CPBR	Çetin Point Based Reconstruction
CS	Classification System
DA	Decision Analysis
DAI	Data In
DAO	Data Out
DEI	Decision In
DEO	Decision Out
DOE	Design of Experiments
EADSIM	Extended Air Defense Simulation
ELINT	Electronic Intelligence
EUROC	Expected Utility Receiver Operating Characteristic
F	Friend
FEI	Feature In
FEN	Friend, Enemy or Neutral
FEO	Feature Out
FN	False Negative or Friend or Neutral
FP	False Positive
FSINT	Foreign Instrumentation Signals Intelligence
GUI	Graphical User Interface
H	Hostile
HRR	High Range Resolution
HSI	Hyperspectral Imagery
HMMWV	High Mobility Multi-purpose Wheeled Vehicle
HUMINT	Human Intelligence
IFF	Identification Friend or Foe
IMINT	Imagery Intelligence
ISOC	Identification System Operating Characteristic

ISR	Intelligence, Surveillance, Reconnaissance
JDL	Joint Directors of Laboratories
LA	Label Accuracy
LGP	Linear Goal Program or Programming
MASINT	Measurement and Signature Intelligence
MBT	Main Battle Tank
MLP	Multilayer Perceptron
MOE	Measure of Effectiveness
MOP	Measure of Performance
MRLS	Mobile Rocket Launcher System
MSI	Multispectral Imagery
MSP	Multinominal Selection Procedure, Multinomial Selection Problem
MSTAR	Moving and Stationary Target Acquisition and Recognition
MVB	Majority Vote Boolean
OBN	One Big Network
OH	Other Hostile
OODA	Observe, Orient, Decide, Act
OSINT	Open-Source Intelligence
PBR	Point Based Reconstruction
PDF	Probability Distribution Function
PNN	Probabilistic Neural Network
RBF	Radial Basis Function
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SAR	Synthetic Aperture Radar
SDMS	Sensor Data Management System
SHADE	Shallow Hide Airborne Deception Experiment
SME	Subject Matter Expert
SNR	Signal to Noise Ratio
T	Target
TDNN	Time Delayed Neural Network
TGT	Target
TN	True Negative
TOD	Target of the Day
TP	True Positive
TPR	True Positive Rate
TT	Target Type
UAV	Unmanned Aerial Vehicle
USAF	United States Air Force
VAR	Vector Autoregressive
VV&A	Verification, Validation, and Accreditation

References

Abramson, M. A. *Pattern Search Algorithms for Mixed Variable General Constrained Optimization problems.* Ph.D. Thesis, Dept. of Computational and Applied Mathematics, Rice University, Houston TX, August 2002.

Adams, N.M. and Hand, D.J. "Improving the Practice of Classifier Performance Assessment." *Neural Computation*, Vol 12, 305-311 (2000).

Adams, N.M. and Hand, D.J. "Comparing classifiers when the misallocation costs are uncertain." *Pattern Recognition*, Vol 32: 1139-1147 (1999).

Albrecht, T.W., AFIT/ENS, Wright-Patterson AFB, OH. Personal correspondence. Various dates throughout fall quarter 2004.

Albrecht, T.W. and Bauer, K.W. "Target identification through time using hidden Markov models," *Intelligent Engineering Systems through ANNs, Vol 14.* 27-32. New York: ASME Press, 2004.

Alsing, Stephen G. *The evaluation of competing classifiers.* Dissertation AFIT/DS/ENS/00-01, Air Force Institute of Technology, Wright-Patterson AFB OH, 2000.

Alsing, Stephen G. and Bauer, Kenneth W. *Survey of statistical analysis and experimental design in ATR evaluation.* Tech. Rep. WP98-05, Air Force Institute of Technology, Wright-Patterson AFB OH, 1998.

Alsing, Stephen G. and Bauer, Kenneth W. *Evaluation of Competing Classifiers for extremely unbalanced data using receiver operating characteristic (ROC) type analyses and a multinomial selection procedure: a preliminary investigation.* Tech. Rep. WP99-01, Air Force Institute of Technology, Wright-Patterson AFB OH, 1999.

Alsing, Stephen G., Bauer, Kenneth W. and Miller, John O. "A multinomial selection procedure for evaluating pattern recognition algorithms," *Pattern Recognition*, Vol 35: 2397-2412, (2002).

Alsing, S.G., Blasch, E.P. and Bauer, K.W. "Three-dimensional receiver operating characteristic (ROC) trajectory concepts for the evaluation of target recognition algorithms faced with the unknown target detection problem," *Proceedings of SPIE Vol. 3718.* 449-458. Bellingham WA: SPIE Press, 1999.

Audet, Charles and Dennis, J.E. "Pattern Search Algorithms for Mixed Variable Programming," *SIAM Journal of Optimization*, Vol 11, No 3: 573-594 (2000).

Bassham, C. Brian. *Automatic target recognition classification system evaluation methodology.* Dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH: 2002.

Bauer, K.W., Alsing, S.G. and Greene, K.A. "Feature screening using signal to-noise ratios", *Neurocomputing*, Vol 31: 29-44 (March 2000)

Bedworth, Mark D. "Source Diversity and Feature-Level Fusion," *Proceedings of Information Decision and Control 99.* 597-602. R. Evans *et al.* (eds.), New York: IEEE Press, 1999.

Bedworth, Mark and O'Brien, Jane. "The Omnibus Model: A New Model of Data Fusion?" *IEEE Aerospace and Electronic Systems Magazine*, Vol 15 No 4: 30-36 (April 2000).

Bhatnagar, V., Shaw, A. K., and Williams, R. W. "Improved automatic target recognition using singular value decomposition," *Proceedings of ICASSP 98.* 2717-2720. New York: IEEE Press, 1998.

Bishop, C. M. *Neural networks for pattern recognition.* Oxford: Clarendon Press, 1995.

Blasch, E.P., Pribilski, M., Daughtery, B., Roscoe, B. and Gunsett, J. "Fusion Metrics for Dynamic Situation Analysis," *Proceedings of SPIE Vol.* 5429. 428-438. Bellingham WA: SPIE Press, 2004.

Bonmassar, G. and Schwartz, E. "Space-Invariant Fourier Analysis: The Exponential Chirp Transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 19, No 10: 1080-1089 (1997).

Box, George E. P. and Jenkins, Gwilym M. *Time Series Analysis: forecasting and control.* San Francisco CA: Holden-Day, 1976.

Boyd, John R. "A Discourse on Winning and Losing," unpublished briefing slides and notes, M-U 43947, Air University Library, Maxwell AFB AL, 1987.

Bradley, A.P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, Vol 30, No 7: 1145-1159 (1997).

Brown, Gerald G. "Top Ten Secrets to Success with Optimization," *Phalanx*, Vol 37, No 4: 12-13, 25-26 (2004).

Cahlink, George, "Better 'Blue Force' Tracking," Air Force Magazine Online, Vol 87: No 6 (June 2004). http://www.afa.org/magazine/june2004/0604blue.asp

Calvert, D. and Kremer, S.C. "Networks and Adaptive State Transitions," *A Field Guide to Dynamical Neural Networks*. 15-25. Kolen, J.F. and Kremer, S.C. (ed.). New York: IEEE Press, 2001.

Cario, Marne C. and Nelson, Barry L. "Autoregressive to anything: Time-series input processes for simulation," *Operations Research Letters*, Vol 19: 51-58 (1996).

Cario, Marne C. and Nelson, Barry L. "Numerical methods for fitting and simulating autoregressive-to-anything processes," *INFORMS Journal on Computing*, Vol 10: 72-81 (1998).

Castellano, G. and Fanelli, A. M. "Variable selection using neural-network models," *Neurocomputing*, Vol 31: 1-13 (2000).

Catlin, Anne E., Bauer, Kenneth W., Mykytka, Edward F., and Montgomery, Douglas C. "System Comparison Procedures for Automatic Target Recognition Systems," *Naval Research Logistics*, Vol 46: 357-371 (1999).

Çetin, M. *Feature-Enhanced Synthetic Aperture Radar Imaging.* Dissertation, Boston University, Boston MA: Feb 2001.

Çetin, M. and Karl, W.C. "Feature-Enhanced Synthetic Aperture Image Formation Based on Nonquadratic Regularization," *IEEE Transactions on Image Processing*, Vol 10, No. 4: 623-631 (2001).

Çetin, M., Karl, W.C. and Castañon, D.A. "Feature Enhancement and ATR Performance using Nonquadratic Optimization-Based SAR Imaging," *IEEE Trans. on Aerospace and Electronic Systems,* Vol 39, No 4: 1375-1395 (2003).

Chang, C., Chiang, S., Du, Q., Ren, H., and Ifarragaerri, A. "An ROC analysis for subpixel detection," *Proc. of International Geoscience and Remote Sensing Symposium (IGARSS) 2001, Vol. 5*. 2355-2357. New York: IEEE Press, 2001.

Chitroub, S., Houacine, A. and Sansal, B. "Statistical characterization and modeling of SAR images," *Signal Processing*, Vol 82: 69-92 (2002).

Chow, C.K. "On optimum rejection error and reject tradeoff," *IEEE Transactions on Information Theory*, IT-16: 41-46 (1970).

Clemans, Paul. *An Investigation of the Optimal Sensor Ensemble for Sensor Fusion*, Thesis AFIT/GOR/ENS/04-03, Air Force Institute of Technology, Wright-Patterson AFB OH, 2004.

Clutz, Thomas C. *A framework for prognostics reasoning.* Dissertation AFIT/DS/ENS/03-01, Air Force Institute of Technology, Wright-Patterson AFB OH: 2003.

Cooke, Tristrom, Redding, Nicholas, Schroeder, Jim and Zhang, Jingxin. "Comparison of selected features for target detection in synthetic aperture radar imagery," *Digital Signal Processing*, Vol 10: 286-296 (2000).

Costantini, Mario, Farina, Alfonso, and Zirilli, Francesco. "The fusion of different resolution SAR images," *Proceedings of the IEEE,* Vol 85, No 1: 139-146 (1997).

Dasarathy, Belur V. "DEI-DEO Fusion as a tool for improving decision robustness across metrics in similarity measure based classifiers," *Proceedings of SPIE Vol. 5434.* 1-12. Bellingham WA: SPIE Press, 2004.

Dasarathy, Belur V. "Multi-Sensor, Multi-Source Information Fusion: Architectures, Algorithms, and Applications," *Short Course Notes, SPIE SC 149*, Bellingham WA: SPIE Press, April 2003.

Dasarathy, Belur V. "Elucidative fusion systems – an exposition," *Information Fusion*, Vol 1: 5-15 (2000a).

Dasarathy, Belur V. "Asymptotic Temporal Fusion Benefits in a Three-Sensor Suite," *Proceedings of SPIE, Vol. 4051.* 2-13. Bellingham WA: SPIE Press, (2000b).

Dasarathy, Belur V. "Sensor Fusion Potential Exploitation—Innovative Architectures and Illustrative Applications," *Proceedings of the IEEE,* Vol 85, No 1: 24-38 (January 1997).

Deler, Bahar and Nelson, Barry L. "Modeling and generating multivariate time series with arbitrary marginals and autocorrelation structures," *Proc. of the 2001 Winter Simulation Conference.* 275-282. B. Peters, J. Smith, D. Medeiros and M. Rohrer (eds.), New York: IEEE Press, 2001.

Demuth, Howard and Beale, Mark. *Neural Network Toolbox for use with Matlab, User's Guide, (Version 3).* Natick MA: The MathWorks, Inc. 1998.

Department of the Air Force (DAF). *Air Warfare. AFDD2-1.* Washington D.C.: HQ USAF, January 2000.

---- *Intelligence, Surveillance, and Reconnaissance Operations. AFDD2-5.2.* Washington D.C.: HQ USAF, April 1999.

----. *USAF Intelligence Targeting Guide. AFP 14-210.* Washington D.C: HQ USAF, February 1998.

Devijver, P.A. and Kittler, J. *Pattern Recognition - a Statistical Approach.* London: Prentice-Hall Int'l, 1982.

DeVore, M. D., Schmid, N. A., and O'Sullivan, J. A. "Analytical and Experimental Performance-Complexity Tradeoffs in ATR," *Proc. of 34th Conf. on Signals, Systems and Computers.* 1519-1523. New York: IEEE Press, 2000.

Diaconis, Persi and Freedman, David. "Asymptotics of Graphical Projection Pursuit," *The Annals of Statistics,* Vol 12, No 3: 793-815 (1984).

Duda, Richard O., Hart, Peter E., and Stork, David G. *Pattern Classification, (2nd ed.).* New York NY: John Wiley & Sons, Inc., 2001.

Dudgeon, D. E. *ATR performance modeling and estimation.* Lincoln Laboratories Tech. Rep. 1051, Lexington MA: Massachusetts Institute of Technology, 1998.

Egan J.P. *Signal Detection Theory and ROC Analysis.* New York: Academic Press, 1975.

Elman, J. L. "Finding Structure in Time," *Cognitive Science*, Vol 14: 179-211 (1990).

Fawcett, T. "Using Rule Sets to Maximize ROC Performance." *Proceedings of the IEEE International Conference on Data Mining (ICDM).* 131-138. New York: IEEE Press (2001).

Feraud, R. and Clerot, F. "A methodology to explain neural network classification," *Neural Networks*, Vol 15: 237-246 (2002).

Feuchter, C.A. *Air Force Analyst's Handbook: On Understanding the Nature of Analysis.* Kirtland NM: Office of Aerospace Studies, Jan 2000.

Frankel, Carl B. and Bedworth, Mark D. "Control, Estimation and Abstraction in Fusion Architectures: Lessons from Human Information Processing," *FUSION. Proc. of the Int. Conf. on Information Fusion 2000.* MoC5-3–5-10. R. Evans *et al.* (eds.), New York: IEEE Press, July 2000.

Fukunaga, K. *Introduction to Statistical Pattern Recognition*, *(2nd ed.).* New York: Academic Press, 1990.

Fukunaga, K., and Hayes, R. "Estimation of classifier performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 11, No 10: 1087-1101 (1989).

Fumera, G., Pillai, I., Roli, F. "A Two-Stage Classifier with Reject Option for Text Categorisation", *5th Int. Workshop on Statistical Techniques in Pattern Recognition (SPR 2004) Vol 3138.* 771-779. Lisbon Portugal: Springer, August 2004.

Fumera, G., Roli, F. "Analysis of error-reject trade-off in linearly combined multiple classifiers", *Pattern Recognition*, Vol 37 No.6: 1245-1265 (2004).

Fumera, G., Roli, F. and Giorgio, G. "Reject option with multiple thresholds," *Pattern Recognition*, Vol 33: 2099-2101 (2000).

Gat, N., Subramanian, S., Barhen, J. and Toomarian, N. "Spectral imaging applications: remote sensing, environmental monitoring, medicine, military operations, factory automation and manufacturing." *Proceedings of SPIE Vol. 2962.* 63-77. Bellingham WA: SPIE Press, 1997.

Gierull, Christoph H. and Sikaneta, Ishuwa, C. "Estimating the effective number of looks in interferometric SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, Vol 40, No 8: 1733-1742 (2002).

Giles, C.L., and Omlin, C., "Representation of Discrete States," *A Field Guide to Dynamical Neural Networks*. 83-102. Kolen, J.F. and Kremer, S.C. (ed.). New York: IEEE Press, 2001.

Greene, K.A., Feature Saliency In Artificial Neural Networks With Application To Modeling Workload, Dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH, 1998.

Greene K.A., Bauer, K.W., Kabrisky, M., Rogers, S.K. and Wilson, G.F. "Estimating pilot workload using Elman recurrent neural networks: a preliminary investigation," *Intelligent Engineering Systems through ANNs, Vol 7*. 703-708. New York: ASME Press, 1997.

Greene, K.A., K Bauer, K.W., Wilson, G.F., Russell, C.A., Rogers, S.K. and Kabrisky, M. "Selection of psychophysiological features for classifying air traffic controller workload in neural networks," *Smart Engineering System Design*, Vol 2: 315-330 (2000).

Hall, David L. and Llinas, James (eds.). *Handbook of Multisensor Data Fusion.* New York: CRC Press, 2001

Hall, David L. and Llinas, James. "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, Vol 85, No 1: 6-23, (January 1997).

Hall, Peter and Li, Ker-Chau. "On Almost Linearity of Low Dimensional Projections from High Dimensional Data," *The Annals of Statistics,* Vol 21, No 2: 867-889 (1993).

Hand, David J. and Till, Robert J. "A simple generalization of the area under the ROC curve for multiple class classification problems," *Machine Learning*, Vol 45: 171-186 (2001).

Haspert, Kent J. "Optimum ID Sensor Fusion for Multiple Target Types." Institute for Defense Analysis (IDA) Document D-2451, Virginia, March 2000.

Haspert, Kent J., Heagy, James F., and Sullivan, Roger J., "General Approach to Template-Based Target Recognition," Institute for Defense Analysis (IDA) Technical Report, Virginia, June 2004.

Hauter, Andrew, Chang, Kuo C. and Karp, Sherman. "Polarimetric fusion for synthetic aperture radar target classification," *Pattern Recognition*, Vol 30, No 5: 769-775 (1997).

Hebert, Adam J., "Building Battlespace Awareness," Air Force Magazine Online, Vol 87: No 9 (Sept 2004). http://www.afa.org/magazine/sept2004/0904Isr.asp

Hill, Justin M. *Evaluating the Performance of Multiple Classifier Systems: A Matrix Algebra Representation of Boolean Fusion Rules*, Thesis AFIT/GOR/ENC/03M-01, Air Force Institute of Technology, Wright-Patterson AFB OH, 2003.

Hogg, Robert V. and Ledolter, Johannes. *Engineering Statistics*. New York: Macmillan Publishing Company, 1987.

Hornik, K., Stinchcombe, M. and White, H. "Multilayer feedforward networks are universal approximators," *Neural Networks,* Vol 2: 359-368 (1989).

Hornik, K., Stinchcombe, M. and White, H. "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks,* Vol 3: 551-560 (1990).

Hughes, G.F., "On the mean accuracy of statistical pattern recognizers," *IEEE Transaction of Information Theory,* Vol IT-14: 55-63 (1968).

Ikeuchi, K., Pomeleau, D., Collins, R., Wheeler, M., Shakunaga, T., Yamazaki, T., and Ohba, K. "Adaptive Model Based ATR System." Technical Report WL-TR-97-1016, Wright-Patterson AFB OH, September 1996.

Jackson, Q. and Landgrebe, D. "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Trans. on Geoscience and Remote Sensing*, Vol 39, No 12: 2664-2679 (2001).

Jacques, D.R. "Modeling considerations for wide area search munitions effectiveness analysis," *Proceedings of the 2002 Winter Simulation Conference*. 878-886. New York: IEEE Press, December 2002.

Johnson, R.A. and Wichern D.W. *Applied Multi-Variate Statistical Analysis, 4th edition*. NJ: Prentice-Hall, Inc., 1998.

Klein, Lawrence A. *Sensor and Data Fusion: A Tool for Information Assessment and Decision Making*. Bellingham WA: SPIE Press, 2004.

Kokar, M. and Kim, K. H. "Review of Multisensor Data Fusion Architectures and Techniques." *Proc. of the 8th IEEE Int. Sym. on Intelligent Control.* 261-266. New York: IEEE Press, 1993.

Kolen, John F. and Kremer, Stefan, C. (ed.). *A Field Guide to Dynamical Recurrent Networks.* New York: IEEE Press, 2001.

Kremer, Stefan C. "On the Computational Power of Elman-Style Recurrent Networks." *IEEE Transactions on Neural Networks*, Vol 6, No. 4: 1000-1004 (1995).

Kumar, Shailesh, Ghosh, Joydeep, and Crawford, Melba M. "Hierarchical fusion of multiple classifiers for hyperspectral data analysis," *Pattern Analysis & Applications,* Vol. 5: 210-220 (2002).

Kuncheva, Ludmila I. *Combining Pattern Classifiers, Methods and Algorithms*. New Jersey: John Wiley & Sons, Inc., 2004.

Kwak, N. and Choi, C.H. "Input feature selection for classification problems," *IEEE Trans. on Neural Networks*, Vol. 13, No 1: 143-159 (2002).

Laine, T. I*., Selection of psychophysiological features across subjects for classifying workload using artificial neural networks*. Thesis AFIT/GOR/ENS/99M-09, Air Force Institute of Technology, Wright-Patterson AFB OH, 1999.

Laine, T.I. and Bauer, K.W. "Feature Selection Assessment and Comparison using Two Saliency Measures in an Elman Recurrent Neural Network," *Proc. of the Int'l Joint Conf. on Neural Networks (IJCNN).* 2807-2812. New York: IEEE Press, 2003.

Laine, T.I. and Bauer, K.W. "A mathematical framework for the optimization of rejection and ROC thresholds in the fusion of correlated sensor data," *Proceedings of SPIE, Vol 5434*. 37-48. Bellingham WA: SPIE Press, 2004a.

Laine, T.I. and Bauer, K.W. " A comparison of 2 neural fusion models using an optimization framework to assess rejection and ROC thresholds for correlated temporal data," *Intelligent Engineering Systems through ANNs, Vol 14*. 149-154. New York: ASME Press, 2004b.

Laine, T.I. and K.W. Bauer, "Improved Target Identification of Correlated Input Data using Recurrent Neural Networks and Feature Selection," *revision submitted* to *Military Operations Research* (2005).

Laine, T.I., Bauer, K.W., Lanning J.W., Russell, C.A. and Wilson G.F. "Selection of input features across subjects for classifying crewmember workload using artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics Part A,* Vol. 32, No. 6: 691-704 (2002).

Landgrebe, David. *"On information extraction principles for hyperspectral data,"* White Paper, School of Electrical & Computer Engineering, Purdue University, West Lafayette IN, 1997.

Landgrebe, David. *"Multispectral data analysis: a signal theory perspective,"* White Paper, School of Electrical & Computer Engineering, Purdue University, West Lafayette IN, 1998.

Landgrebe, David and Biehl, Larry. *An Introduction to MultiSpec$^©$ Ver 5.2001*, West Lafayette IN: Purdue Research Foundation, 2001.

Law, Averill M. and Kelton, W. David. *Simulation and Modeling Analysis (3rd ed.).* New York: McGraw-Hill, 2000.

Lazzerini, B. and Marcelloni, F. "Feature selection based on similarity," *Electronics Letters,* Vol 38, No. 3: 121-122 (2002).

Leap, Nathan J. *An Investigation of the Effects of Correlation, Autocorrelation, and Sample Size in Classifier Fusion*, Thesis AFIT/GOR/ENS/04-06, Air Force Institute of Technology, Wright-Patterson AFB OH, 2004.

Leap, N.J., Clemans, P.P., Bauer, K.W. and Oxley, M.E. " An investigation on the effects of correlation and autocorrelation on classifier fusion and optimal classifier ensembles," *Intelligent Engineering Systems through ANNs, Vol 14.* 627-632. New York: ASME Press, 2004.

Lee, Jong-Sen, Hoppel, Karl W., Mango, Stephan A., and Miller, Allen, R. "Intensity and phase statistics of multilook polarimetric and interferometric SAR imagery," *IEEE Transactions on Geoscience and Remote Sensing,* Vol 32, No 5: 1017-1028 (1994).

Liggins, Martin E. "An Evaluation of CFAR Effects on Adaptive Boolean Decision Fusion Performance for SAR/EO Change Detection," *Proceedings of SPIE, Vol. 4380.* 406-416. Bellingham WA: SPIE Press, 2001.

Lloyd, C.J. "Estimation of a convex ROC curve," *Statistics and Probability Letters*, Vol 59: 99-111 (2002).

Looney, Carl G. *Pattern Recognition using Neural Networks.* New York: Oxford University Press, 1997.

Lütkepohl, Helmut. *Introduction to Multiple Time Series Analysis (2nd ed.).* New York: Springer-Verlag, 1993.

Magnus, Amy L. and Oxley, Mark E. "Fusing and Filtering Arrogant Classifiers," *FUSION. Proc. of the Int. Conf. on Information Fusion 2002.* 388-395. New York: IEEE Press, July 2002.

Mahler, Ronald. "Random Set Theory for Target Tracking and Identification," *Handbook of Multisensor Data Fusion.* 14-1 – 14-33. Hall, D.L. and Llinas, J. (ed.). New York: CRC Press, 2001.

Mak, B. and Blanning, R.W. "An empirical measure of element contribution in neural networks," *IEEE Trans. on Sys., Man, and Cybernetics: Part C*, Vol 28: 561-564 (1998).

Metz, C.E. "ROC methodology in radiologic imaging," *Investigative Radiology*, Vol 21: 720-733 (1986).

Metz, C.E. "Some practical issues of experimental design and data analysis in radiological ROC studies," *Investigative Radiology*, Vol 24: 234-245 (1989).

Meyer, Gregory J., *Classification of Radar Targets using Invariant Features,* Dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH, 2003.

Mitchell, R. A. and Westerkamp, J.J. *A statistical feature based classifier for robust high range resolution radar target identification.* Technical Report, Air Force Research Laboratory, Sensors Directorate (AFRL/SN), Wright-Patterson AFB OH, 1998.

Milisavljevic, N., Bloch, I., Broek, S., Archeroy, M. "Improving mine recognition through processing and Dempster-Shafer fusion of ground penetrating radar data," *Pattern Recognition*, Vol 36: 1233-1250 (2003).

Mindrup, Frank M. *An Investigation of the Effects of Correlation and Autocorrelation on Classifier Fusion with Non-declarations*, Thesis AFIT/GOR/ENS/05-13, Air Force Institute of Technology, Wright-Patterson AFB OH, 2005.

Montgomery, Douglas C., *Design and Analysis of Experiments, Fourth Edition.* New York: John Wiley & Sons, Inc., 1997.

Moody, J., "Forecasting the economy with neural nets: a survey of challenges and solutions," *Neural Networks: tricks of the trade*. 348-371. Orr, G.B and Muller, K.R. (ed.), New York: Springer, 1998.

Morelli, Michael and DeSimone, Antony J. "Application of Dempster-Shafer theory of evidence to the correlation problem," *FUSION. Proc. of the Int. Conf. on Information Fusion 2002.* 759-762. New York: IEEE Press, 2002.

Mossman, Douglas. "Three-way ROCs," *Medical Decision Making*, Vol 19: 78-89 (1999).

Nasr, Hatem N. "Fundamentals of Automatic Target Recognition," *Short Course Notes, SPIE SC 158*, Bellingham WA: SPIE Press, April 2003.

Nelson, Barry L. and Yamnitsky. "Input modeling tools for complex problems," *Proceedings of the 1998 Winter Simulation Conference.* 105-112. D. Medeiros, E. Watson, J. Carson and M. Manivannan (eds.). New York: IEEE Press, 1998.

Nguyen, D. and Widrow, B. "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," *Proc. Int'l Joint Conf. on Neural Networks, Vol 3.* 21-26. New York: IEEE Press, 1990.

O'Brien, J. "Correlated probability fusion for multiple class discrimination," *Proceedings of Information Decision and Control.* 571-577. February 1999.

O'Brien, J. "An algorithm for the fusion of correlated probabilities," *Proceedings of the Int'l Conf. on Multisource-Multisensor Information Fusion, Fusion '98 Vol II.* 565-571. New York: IEEE Press, July 1998.

Oxley, M.E., and K. Bauer, "Classifier Fusion for Improved System Performance." *AFIT/ENS Working Document 02-02*, Wright-Patterson AFB OH, 2002.

304

Paquet, E., Rioux, M. and Arsenault, H. "Invariant pattern recognition for range images using the phase Fourier transform and a neural network," *Optical Engineering* Vol 34: 1178-1183 (1995).

Peterson, W.W., Birdsall, T.G. and Fox, T.G. "The theory of signal detectability," *Transactions of the IRE Professional Group on Information Theory*, PGIT-4: 171-212 (1954).

Piramuthu, S. "Feature selection for financial credit-risk evaluation decisions," *INFORMS Journal on Computing*, Vol 11, No 3: 258-266 (1999).

Priddy, K.L., Rogers, S.K., Ruck, D.W., Tarr, G.L. and Kabrisky, M. "Bayesian selection of important features for feed forward neural networks," *Neurocomputing*, Vol 5, No 2: 91-103 (1993).

Provost, Foster and Fawcett, Tom "Robust Classification for Imprecise Environments." *Machine Learning* Vol. 42 No. 3: 203-231 (2001).

Provost, Foster and Fawcett, Tom "Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions." *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD).* 43-48, 1997.

Provost, Foster, Fawcett, Tom and Kohavi, Ron "The Case against Accuracy Estimation for Comparing Induction Algorithms." *Proceedings of the International Conference on Machine Learning (ICML).* 445-453, 1998.

Ralston, James M. "Bayesian Sensor Fusion Minimum-Cost I.D. Declaration." Institute for Defense Analysis (IDA) Paper P-3441, Virginia: June 1999.

Robinson, Guner S. and Aboutalib, A. Omar "Trade-off Analysis of Multisensor Fusion Levels." *Proc. of the 2nd National Symposium on Sensors and Sensor Fusion*, GACIAC PR-89-01. 21-34. Chicago IL, 1990.

Ross, T. D., J.J. Bradley, L.J. Hudson, and M.P. O'Connor. "SAR ATR – So What's the Problem? - An MSTAR Perspective," *Proceedings of SPIE Vol. 3721.* 662-672. Bellingham WA: SPIE Press, 1999.

Ross, T. D., and J.C. Mossing. "The MSTAR Evaluation Methodology," *Proceedings of SPIE Vol. 3721.* 705-713. Bellingham WA: SPIE Press, 1999.

Ross, T. D., Westerkamp, L.A. Zelino, E. G. and Burns, T.L. "Extensibility and other model-based ATR evaluation concepts," *Proceedings of SPIE Vol. 3070.* 213-222. Bellingham WA: SPIE Press, 1997.

Ross, T.D., Westerkamp, L.A., Dilsavor, R.L. and Mossing, J.C. "Performance measures for summarizing confusion matrices – the AFRL COMPASE approach," *Proceedings of the SPIE Vol. 4727.* 310-321. Bellingham WA: SPIE Press, 2002.

Ross, T., Worrell, S., Velton, V., Mossing, J., and Bryant, M. "Standard SAR ATR evaluation experiments using the MSTAR public release data set," *Proceedings of SPIE Vol. 3370*. 566-573. Bellingham WA: SPIE Press, 1998.

Ruck, Dennis W., Rogers, Steven K., Kabrisky, M., Oxley, Mark E. and Suter, Bruce W. "The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function." *IEEE Transactions on Neural Networks*, Vol 1, No. 4: 296-298 (1990).

Sadowski, Charles, Jr., CID Senior Operations Analyst, ACC/DRSA, Langley AFB VA. Personal interview and personal correspondence. 8 July 2003, 31 August 2004, 28 October 2004.

Sadowski, Charles, Jr., "Measuring Combat Identification," presentation, 69th MORS Symposium, Annapolis MD, June 2001.

Schmeiser, Bruce. "Simulation Experiments," *Handbook of Operations Research and Management Science: Vol 2*, D. Heyman and M. Sobel (eds.), New York: North Holland, 1990.

Schroeder, Jim. "Automatic target detection and recognition using synthetic aperture radar imagery." *Proc. of the 2002 Workshop on the Applications of Radio Science (WARS02) Conference*, Wilkinson, *et al.* (eds.), National Committee for Radio Science, Australia, 2002.

Serpico, Sebastiano B. and Bruzzone, Lorenzo. "A new search algorithm for feature selection in hyperspectral remote sensing images." *IEEE Transactions on Geoscience and Remote Sensing*, Vol 39, No. 7: 1360-1367 (2001).

Shannon, C.E. "A Mathematical Theory of Communication." *The Bell System Technical Journal*, Vol. 27: 379-423, 623-656, July, October (1948).

Shipp, C.A and Kuncheva, L.I. "Relationships between combination methods and measures of diversity in combining classifiers." *Information Fusion*, Vol 3, No 2: 135-138 (2002).

Simone, G., Farina, A., Morabito, F.C., Serpico, S.B., and Bruzzone, L. "Image fusion techniques for remote sensing applications," *Information Fusion*, Vol 3: 3-15 (2002).

Song, Wheyming T. and Hsiao, Li-Ching. "Generation of autocorrelated random variables with a specified marginal distribution." *Proc. of the 1993 Winter Simulation Conference*. 374-377. Evans *et al*. (eds.), New York: IEEE Press, 1993.

Specht, D., "A General Regression Neural Network." *IEEE Transactions on Neural Networks*, Vol. 2, No. 6: 568-576 (1991).

Specht, D.F. "Probabilistic Neural Networks". *Neural Networks*, Vol. 3: 109-118, (1990).

Steinberg, Alan N., Bowman, Christopher L., and White, Franklin E. "Revisions to the JDL Data Fusion Model." *Proceedings of the SPIE Vol. 3719*. 430-441. Bellingham WA: SPIE Press, 1999.

Storm, Susan A. *An Investigation of the Effects of Correlation in Sensor Fusion.* Thesis AFIT/GOR/ENS/03-22, Air Force Institute of Technology, Wright-Patterson AFB OH, 2003.

Storm, S.A., Bauer, K.W. and Oxley, M.E. "An investigation of the effects of correlation in classifier fusion," *Intelligent Engineering Systems through ANNs, Vol 13*. 619-624. New York: ASME Press, 2003.

Sriver, Todd A. *Pattern Search Ranking and Selection Algorithms for Mixed-Variable Optimization of Stochastic Systems*, Dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH, 2004.

Strang, Gilbert. *Linear Algebra and its Applications.* Orlando FL: Harcourt Brace Jovanovich, 1988.

Suvorova, Sofia and Schroeder, Jim. "Automated target recognition using the Karhunen-Loeve transform with invariance," *Digital Signal Processing*, Vol 12: 295-306 (2002).

Swets, J.A., *Signal Detection and Recognition by Human Observers*, New York: John Wiley & Sons, Inc., 1964.

Swets, J.A., Dawes, R.M. and Monahan, J. "Better Decisions through Science," *Scientific American*, Vol 283: 82-87 (Oct 2000).

Swets, J.A., Dawes, R.M. and Monahan, J. "Psychological science can improve diagnostic decisions," *Psychological Science in the Public Interest*, Vol 1: 1-26, (May 2000).

Swets, J.A., and R.M. Pickett. *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory.* New York: Academic Press, 1982.

Tarr, G.L. *Multilayered Feedforward Neural Networks For Image Segmentation*, Dissertation, Air Force Institute of Technology, Wright-Patterson AFB OH, 1991.

Thomas, Jason E. *Multispectral Detection of Ground Targets in Highly Correlated Backgrounds.* Thesis AFIT/GEO/ENP/94D-06, Air Force Institute of Technology, Wright-Patterson AFB OH, 1994.

Utans, J., Moody, J., Rehfuss, S. and Siegelmann, H., "Input variable selection for neural networks: application to predicting the U.S. business cycle," *Proc. of the IEEE/IAFE 1995 CIFE.* 118-122. New York: IEEE Press, 1995.

Varner, Tom. "ACE TestJig Evaluation Metrics." *Jacobs Sverdrup Working Papers,* Dayton OH, Sept 2002.

Varshney, Pramod K. *Distributed Detection and Data Fusion.* New York: Springer, 1997.

Verikas, A. and Bacauskiene, M. "Feature selection with neural networks," *Pattern Recognition Letters*, Vol 23: 1323-1335 (2002).

Wackerly, Dennis, Mendenhall, William and Scheaffer, Richard. *Mathematical Statistics with Applications.* New York: Duxbury Press, 1996.

Wald, A. *Sequential Analysis.* New York: John Wiley & Sons, Inc., 1947.

Waltz, Edward and Llinas, James. *Multisensor Data Fusion.* Boston MA: Artech House, 1990.

Wan, Eric A. "Neural Network Classification: A Bayesian Interpretation." *IEEE Transactions on Neural Networks*, Vol 1, No. 4: 303-305 (1990).

Wang, S., Chen, P., and Lin, W. "Invariant Pattern Recognition by Moment Fourier Descriptor," *Pattern Recognition*, Vol 27: 1735-1742 (1994).

Wasserman, P., and Nostrand, V. *Advanced Methods in Neural Computing*. New York: John Wiley & Sons, Inc., 1993.

Willet, P., Swaszek, P.F. and Blud, R.S., "The good, bad and ugly: distributed detection of a known signal in dependent Gaussian noise." *IEEE Transactions on Signal Processing,* Vol 48: 3266-3279 (2000).

Williams, R., Gross, D., Palomino, A., Westerkamp, J. and Wardell, D. "1D HRR data analysis and ATR assessment," *Proceedings of SPIE Vol. 3370*. 588-599. Bellingham WA: SPIE Press, 1998.

Williams, R., Westerkamp, J., Gross, D., Palomino, A., Kaufman, T. and Fister, T. "Analysis of 1-D HRR moving target ATR," *Proceedings of SPIE Vol. 3721*. 413-424. Bellingham WA: SPIE Press, 1999.

Williams, R., Westerkamp, J., Gross, D. and Palomino, A., "1D High Range Resolution (HRR) Radar." *IEEE AES Systems Magazine,* 37-43. (April 2000).

Wilson, J. Holton and Keating, Barry. *Business Forecasting.* Boston MA: Irwin, Inc., 1994.

Wise, A.R., Fitzgerald, D., and Ross, T.D. "The Adaptive SAR ATR problem set (AdaptSAPS)," *Proceedings of SPIE Vol. 5427.* 366-375. Bellingham WA: SPIE Press, 2004.

Young, M., Dimalanta, A., Konno, D., Peli, T. and Zachery, K. N. "Shade Feature Level Fusion Results," *Proceedings of the MSS National Symposium on Sensor and Data Fusion*, Titan System Corp., June 2001.

Zhang, H. and Sun, G. "Feature selection using tabu search method," *Pattern Recognition,* Vol 35: 701-711 (2002).

Vita

Major Trevor I. Laine was born in San Diego, California. His family then moved back to Oregon, where he spent his initial school years. In 1988, he graduated from Woodrow Wilson High School in Portland, OR and then attended the University of Portland with an AFROTC scholarship. In 1992, he graduated with Bachelor of Science degree in Physics (*cum laude*) and received a commission in the USAF.

His first assignment was to the Space and Missile Systems Center (SMC), Los Angeles AFB, California where he was assigned to the Titan IV System Program Office. After spending three years as a project manager in charge of the Centaur upper-stage liquid rocket engines, Trevor served one year as a project officer in SMC's strategic planning office. Amidst this four-year tour and zest for beach life, he also earned a Master of Business Administration degree from Chapman University. In August of 1997 he was assigned to the Air Force Institute of Technology (AFIT). At AFIT, he earned a M.S. degree in Operations Research, and his thesis, *Selection of psychophysiological features across subjects for classifying mental workload using artificial neural networks*, was selected as the best in the Department of Operational Sciences and nominated for the Commandant's Award.

After graduation, he was assigned as a Weapons System Analyst for the Office of Aerospace Studies (OAS) at Kirtland AFB, New Mexico. After one year at OAS, he was promoted to Deputy Chief of the Analysis Division. In 2001, Trevor returned to AFIT's Department of Operational Sciences in pursuit of a Ph.D. After completion of his doctorate, Trevor will be assigned to AMC/A59, HQ AMC's Studies and Analysis Division, Scott AFB, IL.

	Form Approved OMB No. 074-0188

REPORT DOCUMENTATION PAGE

1. REPORT DATE (DD-MM-YYYY) 27/04/2005	2. REPORT TYPE Doctoral Dissertation	3. DATES COVERED (From – To) October 2001 - April 2005

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
OPTIMIZATION OF AUTOMATIC TARGET RECOGNITION WITH A REJECT OPTION USING FUSION AND CORRELATED SENSOR DATA	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER AFOSR grant #NMIPR045203616 ACC/DRSA
Laine, Trevor I., Major, USAF	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 641, WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DS/ENS/05-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM ATTN: Neal Glassman · 4015 Wilson Blvd · Mail Room 713 · Arlington, VA 22203-1954 · ACC/DRSA ATTN: Charles Sadowski Jr · 216 Hunting Ave, Room 106 · Langley AFB, VA 23665-2777	10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR/NM, ACC/DRSA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This dissertation research makes contributions toward the optimization of automatic target recognition (ATR) systems when a rejection option is included. First, a comprehensive review of the literature inclusive of ATR assessment, fusion, correlated sensor data, and classifier rejection is presented. An optimization framework for the fusion of multiple sensors is then developed. This framework identifies preferred fusion rules and sensors along with rejection and receiver operating characteristic (ROC) curve thresholds without the use of explicit misclassification costs as required by a Bayes' loss function. This optimization framework is the first to integrate both "vertical" warfighter output label analysis and "horizontal" engineering confusion matrix analysis. In addition, optimization is performed for the true positive rate, which incorporates the time required by classification systems. The mathematical programming framework is used to assess different fusion methods and to characterize correlation effects both within and across sensors. A synthetic classifier fusion-testing environment is developed by controlling the correlation levels of generated multivariate Gaussian data. This synthetic environment is used to demonstrate the utility of the optimization framework and to assess the performance of fusion algorithms as correlation varies. The mathematical programming framework is then applied to collected radar data. This radar fusion experiment optimizes Boolean and neural network fusion rules across four levels of sensor correlation. Comparisons are presented for the maximum true positive rate and the percentage of feasible thresholds to assess system robustness. Empirical evidence suggests ATR performance may improve by reducing the correlation within and across polarimetric radar sensors. Sensitivity analysis shows ATR performance is affected by the number of forced looks, prior probabilities, the maximum allowable rejection level and the acceptable error rates. Overall, the optimization framework presented in this research merges and extends current ROC and rejection analysis to include acceptable levels of error when classification costs are difficult to estimate. Further, this framework may be modified to address non-military pattern recognition and fusion efforts to include: medical diagnosis, automatic system prognosis, forecasting, robotics, and environmental monitoring.

15. SUBJECT TER

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. Kenneth W. Bauer, Jr., (AFIT/ENS)
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4328; e-mail: Kenneth.Bauer@afit.edu
U	U	U	UU	330	