RL-TR-91-116
Final Technical Report
July 1991

DISTRIBUTED SENSOR SYSTEM DECISION ANALYSIS USING TEAM STRATEGIES

University of Virginia

Howard C. Choe and Dimitri Kazakos

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

RL-TR-91-116 has been reviewed and is approved for publication.

APPROVED:

VINCENT C. VANNICOLA
Project Engineer

FOR THE COMMANDER:

JAMES W. YOUNGBERG, LtCol, USAF
Deputy Director
Surveillance Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1991	Final , eb 88 - Jan 89

4. TITLE AND SUBTITLE
DISTRIBUTED SENSOR SYSTEM DECISION ANALYSIS USING TEAM STRATEGIES

5. FUNDING NUMBERS
C - F30602-81-C-0169
PE - 62702F
PR - 4506
TA - 11
WU - PW

6. AUTHOR(S)
Howard C. Choe, Dimitri Kazakos

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Virginia
Department of Electrical Engineering
Charlottesville VA 22901

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (OCTS)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-91-116

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Vincent C. Vannicola/OCTS/(315) 330-4437

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
A distributed (or decentralized) multiple sensor system is considered under binary hypothesis environments. The system is deployed with a host sensor and multiple slave sensors. All sensors have their own independent decision makers (DM) which are capable of declaring local decisions based only on their own observation of the environment. The communication between the host sensor (HS) and the slave sensors (SS) is conditional upon the host sensor's command. Each communication that takes place involves a communication cost which plays an important role in approaches taken in this study. The conditional communication with cost initiates the team strategy in making the final decisions at the host sensor. The objectives are not only to apply the team strategy method in the decision making process, but also to minimize the expected system cost (or the probability or error in making decisions) by optimizing thresholds in the host sensor. The analytical expression of the expected system cost is numerically evaluated for Gaussian statistics over threshold locations in the host sensor to find an optimal threshold location for a given communication cost. The computer simulations of various sensor systems for Gaussian observations are also performed to understand the behavior of each system with respect to correct detections, false alarms, and target misses.

14. SUBJECT TERMS
Signal Processing, Sensor Fusion, Electronic Engineering, Communication Systems

15. NUMBER OF PAGES
150

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	U/L

Acknowledgement

I wish to express my appreciation to my thesis advisor, Dr. D. Kazakos, for his guidance throughout this research. I am also indebted greatly to my committee chairperson, Dr. S. G. Wilson, and Dr. P. Papantoni-Kazakos for their careful review of this thesis and invaluable comments. I would like to thank fellow CSL students for their friendships and support throughout my years at the University of Virginia. Finally, I thank my parents, sister, and brother for their prayers, and for always being there.

Table of Contents

List of Figures

List of Tables

List of Symbols Used in Chapter 2

HS	Host sensor
HDM	Host sensor's decision maker
SS	Slave sensor
SDM	Slave sensor's decision maker
TL or TL2	Lower threshold at HS
TU or TU2	Upper threshold at HS
T_{SS}	Decision threshold at SS
FT	Final threshold of system
U_{HS}	Local decision of HS
U_{SS}	Local decision of SS
U_F	Final decision of the system
C( )	The cost function of the system
y_{HS}	Observation received at HS
y_{SS}	Observation received at SS
$P_{e_{HS}}$	Probability of error incurred by the HS only
$P_{e_{Team}}$	Probability of error incurred by team decision
z	$= \begin{cases} 1 & \text{, if } TL \le y_{HS} \le TU \\ 0 & \text{, Otherwise} \end{cases}$
$c_{HS:SS}$	Communication cost constant
$\bar{C}$	Expected system cost
H_0	Environment without a target

H_1	Environment with a target
$\Lambda(y_{HS})$	Likelihood ratio at HS
$\Lambda(y_{SS})$	Likelihood ratio at SS
λ_t	Pre-calculated threshold for HS
$\lambda_{t_{SS}}$	Pre-calculated threshold for SS
$C_{\alpha\beta}$	Cost of deciding α given β
T_0	Ratio of a priori probabilities of environment
$f(U_{SS})$	Final threshold in function of U_{SS}
$Q(y)$	Q-function integrated from y to ∞
$f_{HS_0}(y_{HS})$	Gaussian PDF of y_{HS} given H_0
$f_{HS_1}(y_{HS})$	Gaussian PDF of y_{HS} given H_1
$f_{SS_0}(y_{SS})$	Gaussian PDF of y_{SS} given H_0
$f_{SS_1}(y_{SS})$	Gaussian PDF of y_{SS} given H_1
μ_{HS_0}	Mean received at HS given H_0
μ_{HS_1}	Mean received at HS given H_1
σ_{HS_0}	Standard deviation received at HS given H_0
σ_{HS_1}	Standard deviation received at HS given H_1
μ_{SS_0}	Mean received at SS given H_0
μ_{SS_1}	Mean received at SS given H_1
σ_{SS_0}	Standard deviation received at SS given H_0
σ_{SS_1}	Standard deviation received at SS given H_1

List of Symbols Used in Chapter 3

Those symbols do not appear in this section are the symbols which are commonly used in Chapter 2 and Chapter 3. Please refer to "List of Symbols used in Chapter 2" for the symbols not listed here.

SS1	Slave sensor 1
SS2	Slave sensor 2
SDM1	Slave sensor 2's decision maker
SDM2	Slave sensor 2's decision maker
TL or TL3	Lower threshold at HS
TU or TU3	Upper threshold at HS
T_{SS1}	Decision threshold at SS1
T_{SS2}	Decision threshold at SS2
U_{SS1}	Local decision of SS1
U_{SS2}	Local decision of SS2
y_{SS1}	Observation received at SS1
y_{SS2}	Observation received at SS2
$f(U_{SS1}, U_{SS2})$	Final threshold in function of U_{SS1} and U_{SS2}
$f_{SS1_0}(y_{SS1})$	Gaussian PDF of y_{SS1} given H_0
$f_{SS1_1}(y_{SS1})$	Gaussian PDF of y_{SS1} given H_1
$f_{SS2_0}(y_{SS2})$	Gaussian PDF of y_{SS2} given H_0
$f_{SS2_1}(y_{SS2})$	Gaussian PDF of y_{SS2} given H_1

μ_{SS1_0}	Mean received at SS1 given H_0
μ_{SS1_1}	Mean received at SS1 given H_1
σ_{SS1_0}	Standard deviation received at SS1 given H_0
σ_{SS1_1}	Standard deviation received at SS1 given H_1
μ_{SS2_0}	Mean received at SS2 given H_0
μ_{SS2_1}	Mean received at SS2 given H_1
σ_{SS2_0}	Standard deviation received at SS2 given H_0
σ_{SS2_1}	Standard deviation received at SS2 given H_1

List of Symbols Used in Chapter 4

For those symbols not appearing in this section, please refer to either "List of Symbols Used in Chapter 2" or "List of Symbols Used in Chapter 3" since those symbols are commonly used in Chapter 2, Chapter 3, and Chapter 4.

$P_{e_{T1}}$	Probability of error incurred by team decision with SS1 only
$P_{e_{T2}}$	Probability of error incurred by team decision with SS1 and SS2
TL1 or TL31	Lower threshold of HS for communicating with SS1 only
TL2 or TL32	Lower threshold of HS for communicating with SS1 and SS2
TU2 or TU32	Upper threshold of HS for communicating with SS1 and SS2
TU1 or TU31	Upper threshold of HS for communicating with SS1 only
z_{T1}	$= \begin{cases} 1 & \text{, if } TL1 \le y_{HS} \le TL2 \text{ ,or } TU2 \le y_{HS} \le TU1 \\ 0 & \text{, Otherwise} \end{cases}$
z_{T2}	$= \begin{cases} 1 & \text{, if } TL2 \le y_{HS} \le TU2 \\ 0 & \text{, Otherwise} \end{cases}$
c_{T1}	Communication cost constant for commnicating with SS1 only
c_{T2}	Communication cost constant for communicating with SS1 and SS2
$f(U_{SS1}, U_{SS2})$	Final threshold in function of U_{SS1} and U_{SS2}

CHAPTER 1

Introduction

1.1. Literature Review and Goals

The extension of classical detection theory to the case of distributed sensors is discussed in [1]; in particular, the problem of constructing decentralized Bayesian hypothesis testing rules is considered. In [2], the optimal data fusion structure is developed, when the global decision is obtained by weighting local decisions according to the reliability of detectors and comparing to a threshold. In that paper optimal fusion rules are derived when the decision rules per individual detector are known. Those rules are expressed in terms of the probability of false alarm and the probability of miss. The systems considered in [1] and [2] have a fusion center which always requires all the sensors to transmit their local decisions. But in certain applications, such continuous communication may not be desirable; such is the case in environments with adversaries. In [3], one of data fusion methods in distributed networks is to apply the Neyman-Pearson approach to find all of the optimal decision rules at each site (or detector). The optimal threshold for the system using those optimal decision rules found at each site is not stated. In [4], the problem of optimal data fusion in the sense of the Neyman-Pearson test is considered; uncertainty regions at the detectors are considered, but this information is used to enhance the decision at the data fusion center. A region where a definite decision cannot be made is called an uncertainty region. There are no communications between sensors when the observation falls in the confident region. Papastavrou and Athans [5] evaluated a two-sensor network, consisting of a primary sensor and a consulting sensor using team strategy method, with performance criterion of the probability of error. They also provide numerical results for varying quality of observations at different sensors and *a priori* probabilities. The relationship between the position of threshold in the primary sensor and the system probability of error is not clearly stated.

Through this study, we applied team decision strategies to three different sensor systems and an analysis of each system was performed. The three different systems are a two-sensor-system (2SS), a three-sensor-system (3SS), and a two/three-sensor-system (2/3SS). The main goals of this study are to identify the level of risk which prohibits communication between sensors, to obtain the analytical expression of optimal global decision rules for each system considered, to investigate the behavior of decision thresholds and system performance for a given communication cost (or risk), and to compare the performance of each system through numerical evaluations and system simulations.

1.2. Overview of Chapters

In Chapter 1, general concepts are discussed. The team strategy method in decision processes and the communication cost involved in the team strategy are described in this chapter. The binary hypothesis environment and assumptions made in deriving the expected system cost are also stated. A couple of examples, concerning the interpretation of the communication cost constant in real system, are also presented.

In Chapter 2, a two-sensor-system (2SS) similar to the system studied by Papastavrou and Athans [5] is considered. The model consists of a host sensor (HS) and a slave sensor (SS). The model used the team strategy method in making final decisions, depending upon the local decisions made by the host sensor. The expected system cost is expressed in general probabilistic terms. This expression is numerically evaluated based on the assumption of Gaussian observation.

Adding an additional sensor to the system, a three-sensor-system (3SS) is treated in Chapter 3. In this system both slave sensors return their binary information to the host sensor when a request of information is made by the host sensor. The analytical expression as well as the numerical evaluation of the expected system cost are also performed.

Chapter 4 contains an analysis of a two/three-sensor-system (2/3SS). This system can be considered as a combination of 2SS and 3SS since 2/3SS switches from 2SS to 3SS, or vice versa, depending upon the local decision of the host sensor. Most of the 2/3SS model criteria are the same as in the previous chapters. Numerical evaluation is also done for this system.

The results from the numerical evaluation of the expected cost of each system are presented in Chapter 5. The data are available in both tables and plots. The plots are attached at the end of Chapter 2, 3, and 4. Each system is compared to other systems based on the results. In the numerical evaluation of the expected system cost, $\bar{C}$, FORTRAN programs are written and these are attached in Appendix A, B, and C in order of 2SS, 3SS, and 2/3SS, respectively.

In Chapter 6, simulation results of the systems analyzed in Chapter 2, 3, and 4 are presented. Tables and plots are used to show the data from the simulation. A FORTRAN program is also written to carry out the simulation. The program is attached in Appendix E.

Finally, an overall summary of this study and the conclusion are written in Chapter 7.

1.3. Environment

A binary hypothesis environment, H_0 and H_1, is considered. H_0 indicates that there is no target present. H_1 indicates that a target is present.

1.4. Team Strategies

The sensor communications occur only when the host sensor declares lack of confidence in its local observation. When a slave sensor transmits only a binary decision to the host sensor, some information received at the slave sensor may be lost, but the risk of interception by adversaries is then reduced. Examples of the costs or the risks in real systems are given in section 1.6. The final threshold in the host sensor is evaluated using the binary decisions transmitted from the slave sensors and *a prior* probabilities of the hypothesis. The final threshold (FT), then, is compared against the observation at the host sensor to make the final decision. In other words, the final decision at the host sensor is declared by using its local analog data and the binary decisions transmitted from the slave sensors.

The team strategy allows collaboration of sensors in the distributed (multiple) sensor system. The differences between distributed sensor systems which use the team strategies and those that do not are;

(1) The host sensor in team strategies has an overall control of communication between the host sensor (HS) and the rest of the sensors, namely the slave sensors (SSs).

(2) All the sensors including HS and SSs are capable of making their own local decisions utilizing observations from their local environment.

(3) The host sensor carries multiple thresholds which divide the decision space into either three regions (2 thresholds) or five regions (4 thresholds). It uses them to recruit inputs from other sensors accordingly, which means that the communication schemes between the host sensor and the slave sensor are determined, depending upon the decisions of the HS.

(4) The systems do not have a central data fusion center. The host sensor is capable of making the final decision either based on its local observations only, or through communication with the slave sensors.

1.5. Assumptions

(1) The observations received at different sensors are mutually independent conditioned on each hypothesis.

(2) All the sensors used in the model are considered identical in performance.

(3) The influence of the number of observations available at each sensor is ignored.

1.6. Communication Cost Constant (CCC)

There may be many ways to interpret an application of the communication cost constant (CCC) in real systems. An example would be the communication between sensors in a hostile environment, where interception is possible. Then, the communication cost constant can be interpreted as the probability of interception, for example.

The other way to interpret the communication cost constant is that a limitation of bandwidth, a duration of time delay in communication, quality of information obtained by communication, etc.

This study can be applied with a modification when the environment is non-hostile and the cost is known. In this case, the communication cost constant can represent a physical value, such as a dollar cost, etc. For example, if there is an allocated asset or capital for the communication between each party, the asset (or budget) should be wisely used to obtain the information from the other party. If the asset is $100.00/month and the communication cost is $10.00/communication, there are only 10 communications per month allowed. Thus the system should use the communication capability when it is really required. On the other hand, if the communication cost constant is $1.00/communication, the system have 100 communications per month. This case the system can use the communication capability more frequently.

In this paper, the communication cost constant is interpreted as the probability of interception. The larger communication cost constant indicates greater risk in communication with other parties. Thus when the communication cost constant is null, the communication between parties, i.e., the host sensor and the slave sensors, are encouraged and desirable; however, as the communication cost constant increases, the exchange of information is restricted to the cases of "must communicate" only.

CHAPTER 2

Analysis of a Two-Sensor-System (2SS)

2.1. The Model and Configuration

2.1.1. Host Sensor (HS) and its Decision Maker (HDM)

The host sensor has two decision boundaries providing three decision regions. One of the boundaries is called a lower threshold (TL) and the other is called an upper threshold (TU). When an observation falls below TL, HDM will declare "No Target Detected". When an observation is between TL and TU, HDM declares "Not sure and communication necessary". Finally, when an observation falls above TU, HDM declares "Target Detected". Throughout this paper, the three decisions mentioned in the above will be denoted by a set $U_{HS} = \{0, ?, 1\}$, respectively. These decision regions are shown in Figure 2.1. These thresholds can be varied from one mission to another, depending on the specific requirements and constraints. The thresholds control decision accuracies and the frequency of communication between the host sensor and the slave sensors. The narrower the gap between TL and TU is, the less communication between HS and SS would occur. This is because the gap between the thresholds is directly related to the uncertainty decision region in the host sensor. The decision region between TL and TU may be called a dubious region or uncertainty region.

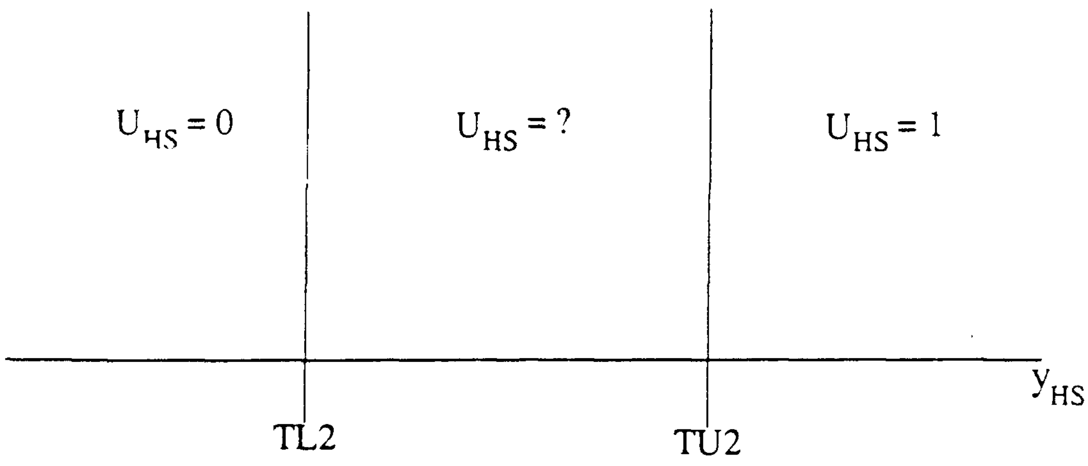

Figure 2.1 Decision Boundaries of HS for 2SS

2.1.2. Slave Sensor (SS) and its Decision Maker (SDM)

The slave sensors have a single decision threshold (T_{SS}) providing two decision regions. The slave sensor does not have an uncertainty region, meaning the SDMs are forced to make a decision either "0" or "1".

When an observation falls below T_{SS}, SDM declares "No Target Detected". When an observation falls above T_{SS}, SDM declares "Target Detected". In this paper these decisions are represented by a set $U_{SS} = \{0, 1\}$.

2.1.3. Host Sensor's Final Decision Threshold

When the communication between the host sensor and the slave sensors occurs, the analog data of the host sensor and the slave sensor's binary data are used to determine the final decision. The threshold for the final decision is evaluated utilizing the binary information from the slave sensors and *a priori* probabilities. Then this final threshold is compared against the observation received by the HS to determine whether there is a target or not. The final decision is denoted by U_F. FT can vary from one evaluation to another since U_{SS} provided from the SDMs may differ from one communication to the other.

2.1.4. The Overall Process of The Model

An observation, y_{HS}, which is received at the host sensor, is mutually independent from the observations received by other sensors. When the observation is greater than or equal to TU (TU2 in Figure 2.1) or is less than or equal to TL (TL2 in Figure 2.1), the host sensor's local decision, U_{HS}, 1 or 0, respectively, becomes the final decision, U_F. When y_{HS} is between TL2 and TU2, a request of assistance from the host sensor to the slave sensor is transmitted. The slave sensor returns the local decision, U_{SS}, to the host sensor's team processing unit upon the request. This communication process involves a communication cost constant (CCC). U_{SS} is also determined based upon an independent observation at the slave sensor. The slave sensor makes a binary decision since it only has one decision threshold. Refer to Figure 2.2.

2.2. Definition of the System Cost Function

The system cost is defined by the total system probability of error. The following is the cost function of the system, C(,,,,), which is represented by error probabilities of the individual sensor as well as the error caused by the team process.

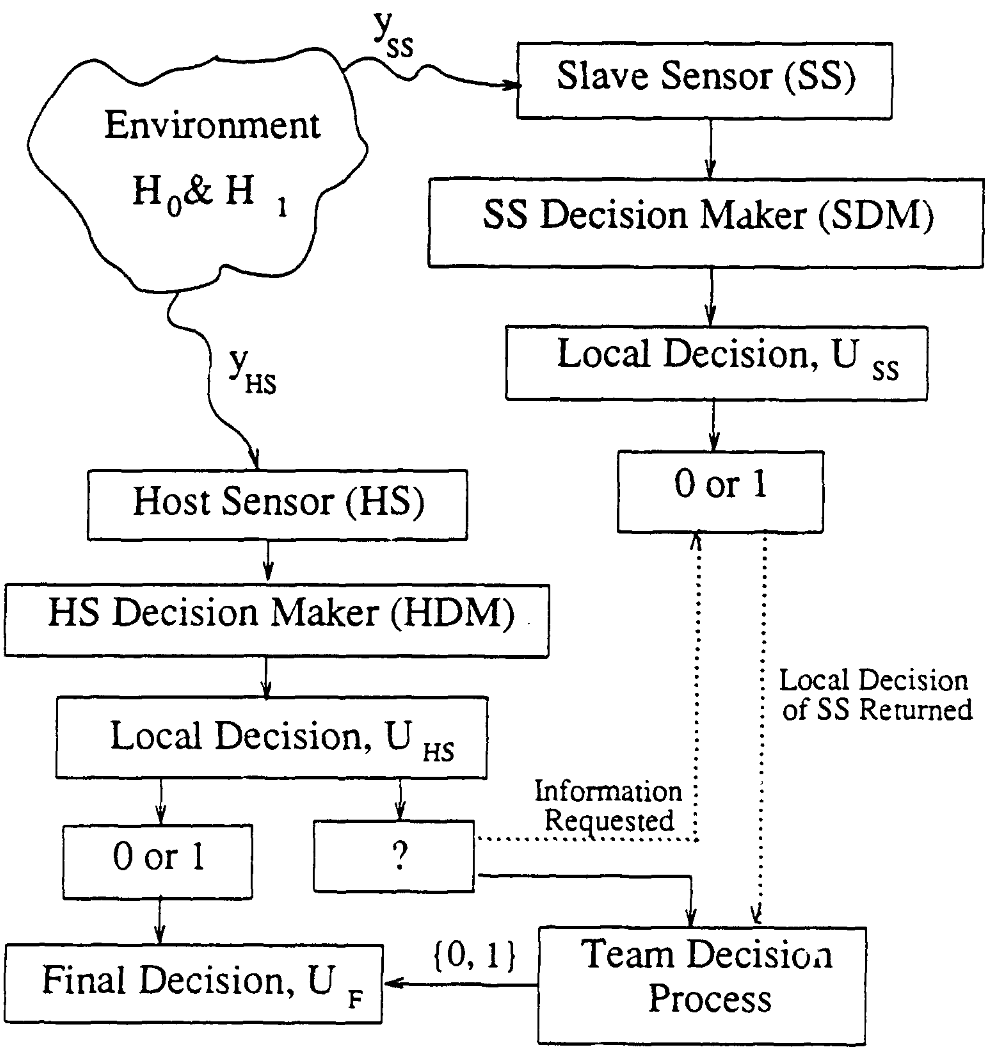

Figure 2.2 Model Configuration of 2SS

$$C(z, y_{HS}, TL, TU, c_{HS:SS}) = (1-z)\cdot P_{e_{HS}} + z\cdot(P_{e_{Team}} + c_{HS:SS}) \qquad (2.2.1)$$

In (2.2.1), z is a value that determines whether the communication should be made or not. z takes a binary number either 0 or 1. When the host sensor makes "?"

decision, z becomes 1. In case the host sensor makes a confident decision, z becomes 0. It is obvious from the expected system cost function that the team decision operation takes a role only when a communication channel is open, i.e. $z = 1$, between the host sensor and the slave sensor. When $z = 0$, meaning that there will be no communication between the host sensor and the slave sensor, the cost function becomes that of a centralized system of single sensor with a possibility of smaller error.

2.3. Evaluation of an System's Expected Cost, $\bar{C}$

Since the system cost function is defined, it is possible to evaluate an system's expected cost.

$$
\begin{aligned}
\bar{C} &= E\{C(z,TL,TU,c_{HS:SS})\} \\
&= C(0,TL,TU,c_{HS:SS}) \cdot P_r(z{=}0) + C(1,TL,TU,c_{HS:SS}) \cdot P_r(z{=}1) \\
&= C(0,TL,TU,c_{HS:SS}) \cdot \{1-P_r(z{=}1)\} + C(1,TL,TU,cc) \cdot P_r(z{=}1) \\
&= C(0,TL,TU,c_{HS:SS}) + \{C(1,TL,TU,c_{HS:SS}) - C(0,TL,TU,cc)\} \cdot P_r(z{=}1)
\end{aligned} \tag{2.3.1}
$$

By evaluating C(0,,,,) and C(1,,,,), and using (2.2.1); the following is obtained;

$$C(0,TL,TU,c_{HS:SS}) = P_{e_{HS}} \tag{2.3.2}$$

$$C(1,TL,TU,c_{HS:SS}) = P_{e_{Team}} + c_{HS:SS} \tag{2.3.3}$$

Therefore the expression of equation becomes as follows:

$$\bar{C} = P_{e_{HS}} + (P_{e_{Team}} + c_{HS:SS} - P_{e_{HS}}) \cdot P_r(z{=}1) \tag{2.3.4}$$

This equation is further developed in detail such as;

$$P_{e_{HS}} = P_r(\text{false local decision at HS})$$

$$= P_r(\text{Decide } H_1 \mid H_0) \cdot P_r(H_0) + P_r(\text{Decide } H_0 \mid H_1) \cdot P_r(H_1)$$

$$= P_r(y_{HS} \geq TU \mid H_0) \cdot P_r(H_0) + P_r(y_{HS} \leq TL \mid H_1) \cdot P_r(H_1) \qquad (2.3.5)$$

$$P_r(z=1) = P_r(\text{uncertainty in decision; communication channel open})$$

$$= P_r(TL < y_{HS} < TU)$$

$$= P_r(TL < y_{HS} < TU \mid H_0) \cdot P_r(H_0) + P_r(TL < y_{HS} < TU \mid H_1) \cdot P_r(H_1) \qquad (2.3.6)$$

$$P_{e_{Team}} = P_r(\text{error resulted by communication using team strategy}) = P_r(E)$$

$$= P_r(E \mid y_{HS} \in [TL,TU]) \cdot P_r(y_{HS} \in [TL,TU])$$

$$= P_r(E,\ y_{HS} \in [TL,TU] \mid H_0) \cdot P_r(H_0) + P_r(E,\ y_{HS} \in [TL,TU] \mid H_1) \cdot P_r(H_1)$$

$$= P_r(E,\ y_{HS} \in [TL,TU] \mid H_0, U_{SS}) \cdot P_r(U_{SS} \mid H_0) \cdot P_r(H_0)$$

$$+ P_r(E,\ y_{HS} \in [TL,TU] \mid H_1, U_{SS}) \cdot P_r(U_{SS} \mid H_1) \cdot P_r(H_1) \qquad (2.3.7)$$

where $P_r(H_0)$ and $P_r(H_1)$ are *a priori* probabilities of the environment H_0 and H_1 respectively. U_{SS} is the local decision determined by the slave sensor.

In the (2.3.7) it is quite reasonable that a decision of the slave sensor, the binary data, takes a part since the communication between the host sensor and the slave sensor is established as a team effort. This is shown in the equation by giving the probability terms conditioned on the decision of the slave sensor. To evaluate

$$P_r(E,\ y_{HS} \in [TL,TU] \mid U_{SS}, H_i), \qquad i=0,1 \qquad (2.3.8)$$

it is necessary to compute the likelihood ratio, $\Lambda(y_{HS}, U_{SS})$. This probability is the probability of error induced in the host sensor due to the communication with the slave sensor (refer to (2.3.7)). Thus, the probability of error in the slave sensor will contribute to the probability of error evaluated in the host sensor.

2.4. The Likelihood Ratio Test

In evaluating the Likelihood Ratio (LR), it is assumed that the individual observations received at each sensors are independent form each other. Thus, their local decisions are also independent. In other words, the local decision of the slave sensor is statistically independent from (or not coupled with) either the local decision or the observation of the host sensor. Then the LR of this system can be written as

$$\Lambda(y_{HS}, U_{SS}) = \frac{P_r(y_{HS}, U_{SS} \mid H_1)}{P_r(y_{HS}, U_{SS} \mid H_0)} \tag{2.4.1}$$

and the above equation is re-written as follows:

$$\Lambda(y_{HS}, U_{SS}) = \frac{P_r(y_{HS} \mid H_1)}{P_r(y_{HS} \mid H_0)} \cdot \frac{P_r(U_{SS} \mid H_1)}{P_r(U_{SS} \mid H_0)} \cdot \frac{P_r(H_1)}{P_r(H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \tag{2.4.2}$$

$$\text{where,} \quad \lambda_t = \frac{C_{10} - C_{00}}{C_{01} - C_{11}} \text{ : a pre–calculated threshold, and} \tag{2.4.3}$$

$$C_{\alpha\beta} \text{ : a cost of deciding } \alpha \text{ given } \beta$$

For the most of cases it is assumed that a cost of making a false decision and cost of missing target are the same, i.e.

$$C_{01} = C_{10},$$

and this also applies to a cost of making a correct decision, for example,

$$C_{00} = C_{11}.$$

These conditions will give the pre-calculated threshold $\lambda_t = 1$. Re-writing (2.4.2) into (2.4.4), which plays an important role in evaluating (2.3.8) is obtained.

$$\frac{P_r(y_{HS} \mid H_1)}{P_r(y_{HS} \mid H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \cdot \frac{P_r(H_0)}{P_r(H_1)} \cdot \frac{P_r(U_{SS} \mid H_0)}{P_r(U_{SS} \mid H_1)} \tag{2.4.4}$$

and now defining the following,

$$g(y_{HS}) = \frac{P_r(y_{HS} \mid H_1)}{P_r(y_{HS} \mid H_0)}, \tag{2.4.5}$$

$$T_0 = \frac{P_r(H_0)}{P_r(H_1)}, \text{ and} \tag{2.4.6}$$

$$f(U_{SS}) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS} \mid H_0)}{P_r(U_{SS} \mid H_1)} \tag{2.4.7}$$

then (2.4.4) becomes as

$$g(y_{HS}) \underset{U_F=0}{\overset{U_F=1}{\gtrless}} f(U_{SS}) \tag{2.4.8}$$

$g(y_{HS})$ is the decision-statistic and $f(U_{SS})$ is depends on the slave sensor's decision. The function $f(U_{SS})$ represents the final threshold (FT) in the host sensor after a communication is exchanged. As is seen in (2.4.8), the function $f(U_{SS})$ takes two different values (thresholds) depending on the decision of the slave sensor, U_{SS}. Since the slave sensor is forced to make a decision based only upon its observation, U_{SS} is going to be either "0" or "1". More explicit expression of the function $f(U_{SS})$ is

$$f(U_{SS}=0) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS}=0 \mid H_0)}{P_r(U_{SS}=0 \mid H_1)}, \tag{2.4.9}$$

$$f(U_{SS}=1) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS}=1 \mid H_0)}{P_r(U_{SS}=1 \mid H_1)} \tag{2.4.10}$$

Then, the final decision, U_F, rules can be written as

$$U_F = \begin{cases} 1 & , \text{ if } y_{HS} \geq TU \\ 1 & , \text{ if } g(y_{HS}) \geq f(U_{SS}) \\ 0 & , \text{ if } g(y_{HS}) < f(U_{SS}) \\ 0 & , \text{ if } y_{HS} < TL \end{cases}$$

All equations needed to evaluate (2.3.8) which is from (2.3.7) are obtained. By evaluating (2.3.8) further, the following is derived:

$$\begin{aligned} &P_r(E, y_{HS} \in [TL, TU]) \\ &= P_r(H_0) \cdot \sum_{U_{SS}=0}^{U_{SS}=1} P_r(E, y_{HS} \in [TL, TU] \mid U_{SS}, H_0) \cdot P_r(U_{SS} \mid H_0) \\ &\quad + P_r(H_1) \cdot \sum_{U_{SS}=0}^{U_{SS}=1} P_r(E, y_{HS} \in [TL, TU] \mid U_{SS}, H_1) \cdot P_r(U_{SS} \mid H_1) \\ &= P_r(H_0) \cdot \sum_{U_{SS}=0}^{U_{SS}=1} P_r\{g(y_{HS}) \overset{U_F=1}{>} f(U_{SS}) \text{ and } y_{HS} \in [TL, TU] \mid U_{SS}, H_0\} \cdot P_r(U_{SS} \mid H_0) \\ &\quad + P_r(H_1) \cdot \sum_{U_{SS}=0}^{U_{SS}=1} P_r\{g(y_{HS}) \underset{U_F=0}{<} f(U_{SS}) \text{ and } y_{HS} \in [TL, TU] \mid U_{SS}, H_1\} \cdot P_r(U_{SS} \mid H_1) \end{aligned} \quad (2.4.11)$$

Thus, when (2.3.5), (2.3.6), (2.3.7), and (2.4.11) are substitute into (2.3.4), the general expression of the expected system cost is obtained.

2.5. $\bar{C}$ under Gaussian Assumption

The Gaussian distribution are applied to the probabilistic expression of $\bar{C}$ so that numerical method can be used to evaluate the $\bar{C}$ for various thresholds in the host sensor that effects system performances.

2.5.1. The Q(y)-function

In evaluating the probabilities which are involved in the analytical expression of the expected system cost, an integration of Gaussian probability density function is required. We define the Q(y) function to be

$$Q(y) = \int_{y}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}} dz \tag{2.5.1.1}$$

2.5.2. Probability Density Functions

The probability density functions under H_0 and H_1 at each sensors are written below. For the symbols used in the expression, please refer to the beginning of the thesis under "Symbols used in Chapter 2".

2.5.2.1. Gaussian PDFs at the Host Sensor

$$f_{HS_0}(y_{HS}) = \frac{1}{\sqrt{2\pi}\,\sigma_{HS_0}} e^{\frac{-(y_{HS}-\mu_{HS_0})^2}{2\sigma^2_{HS_0}}} \tag{2.5.2.1.1}$$

$$f_{HS_1}(y_{HS}) = \frac{1}{\sqrt{2\pi}\,\sigma_{HS_1}} e^{\frac{-(y_{HS}-\mu_{HS_1})^2}{2\sigma^2_{HS_1}}} \tag{2.5.2.1.2}$$

2.5.2.2. Gaussian PDFs at the Slave Sensor

$$f_{SS_0}(y_{SS}) = \frac{1}{\sqrt{2\pi}\,\sigma_{SS_0}} e^{\frac{-(y_{SS}-\mu_{SS_0})^2}{2\sigma^2_{SS_0}}} \tag{2.5.2.2.1}$$

$$f_{SS_1}(y_{SS}) = \frac{1}{\sqrt{2\pi}\,\sigma_{SS_0}} e^{\frac{-(y_{SS}-\mu_{SS_1})^2}{2\sigma^2_{SS_1}}} \tag{2.5.2.2.2}$$

2.5.3. Decision Boundary of Slave Sensor (SS)

A LRT is used to find SDM's optimal decision threshold, T_{SS}.

$$\Lambda_{SS}(y_{SS}) = \frac{P_r(H_0)}{P_r(H_1)} \cdot \frac{C_{10}-C_{00}}{C_{01}-C_{11}} = \lambda_{t_{SS}} \cdot \frac{P_r(H_0)}{P_r(H_1)} \tag{2.5.3.1}$$

and the LR can also represented as

$$\Lambda_{SS}(y_{SS}) = \frac{P_r(y_{SS} \mid H_1)}{P_r(y_{SS} \mid H_0)} = \frac{\frac{1}{\sqrt{2\pi}\,\sigma_{SS_1}} \cdot \exp\left\{-\frac{(y_{SS}-\mu_{SS_1})^2}{2\sigma_{SS_1}}\right\}}{\frac{1}{\sqrt{2\pi}\,\sigma_{SS_0}} \cdot \exp\left\{-\frac{(y_{SS}-\mu_{SS_0})^2}{2\sigma_{SS_0}}\right\}} \tag{2.5.3.2}$$

Equating (2.5.3.1) and (2.5.3.2) and taking the natural logarithm on both sides of the equation, we obtain

$$\log_e\left\{\lambda_{t_{SS}} \cdot \frac{P_r(H_0)}{P_r(H_1)}\right\}$$

$$= \log_e \frac{\sigma_{SS_0}}{\sigma_{SS_1}} + \frac{(\sigma^2_{SS_1}-\sigma^2_{SS_0})y_{SS}^2 + 2(\mu_{SS_1}\cdot\sigma^2_{SS_0} - \mu_{SS_0}\cdot\sigma^2_{SS_1})y_{SS} + \mu^2_{SS_0}\cdot\sigma^2_{SS_1} - \mu^2_{SS_1}\cdot\sigma^2_{SS_0}}{2\sigma^2_{SS_0}\cdot\sigma^2_{SS_1}}$$

By re-arranging the terms, the above equation can be written as (2.5.3.3).

$$(\sigma^2_{SS_1}-\sigma^2_{SS_0})y_{SS}^2 + 2(\mu_{SS_1}\cdot\sigma^2_{SS_0} - \mu_{SS_0}\cdot\sigma^2_{SS_1})y_{SS}$$
$$+\left[\mu^2_{SS_0}\cdot\sigma^2_{SS_1} - \mu^2_{SS_1}\cdot\sigma^2_{SS_0} - 2\sigma^2_{SS_0}\cdot\sigma^2_{SS_1}\log_e\left\{\lambda_{t_{SS}} \cdot \frac{P_r(H_0)}{P_r(H_1)}\right\}\right] = 0 \tag{2.5.3.3}$$

Solving (2.5.3.3) for y_{SS}, the optimal threshold for the slave sensor, T_{SS} is found.

$$T_{SS} = \frac{-(\mu_{SS_1}\cdot\sigma^2_{SS_0} - \mu_{SS_0}\cdot\sigma^2_{SS_1})}{(\sigma^2_{SS_1}-\sigma^2_{SS_0})} \pm \frac{1}{(\sigma^2_{SS_1}-\sigma^2_{SS_0})}$$

$$\cdot\sqrt{(\mu_{SS_1}\cdot\sigma^2_{SS_0} - \mu_{SS_0}\cdot\sigma^2_{SS_1})^2 - (\sigma^2_{SS_1}-\sigma^2_{SS_0})\left[\mu^2_{SS_0}\cdot\sigma^2_{SS_1} - \mu^2_{SS_1}\cdot\sigma^2_{SS_0} - 2\sigma^2_{SS_0}\cdot\sigma^2_{SS_1}\cdot\log_e\left\{\lambda_{t_{SS}} \cdot \frac{P_r(H_0)}{P_r(H_1)}\right\}\right]} \tag{2.5.3.4}$$

In (2.5.3.4) we require $\sigma^2_{SS_0}$ not equal $\sigma^2_{SS_1}$. For the case $\sigma_{SS_0} = \sigma_{SS_1} = \sigma$, (2.5.3.3) becomes

$$2\sigma^4 \cdot \log_e\left\{\lambda_{t_{SS}} \cdot \frac{P_r(H_0)}{P_r(H_1)}\right\} = 2\sigma^2(\mu_{SS_1} - \mu_{SS_0})y_{SS} - (\mu_{SS_1}^2 - \mu_{SS_0}^2)$$

Again, solving for y_{SS},

$$T_{SS} = \frac{\mu_{SS_0} + \mu_{SS_1}}{2} + \frac{\sigma^2}{\mu_{SS_1} - \mu_{SS_0}} \log_e\left\{\lambda_{t_{SS}} \cdot \frac{P_r(H_0)}{P_r(H_1)}\right\} \tag{2.5.3.5}$$

For the numerical evaluation performed later in this chapter, the threshold for the slave sensor is evaluated by using (2.5.3.5). The decision at the slave sensor is carried out as below:

$$U_{SS} = \begin{cases} 0 \ , & \text{if } y_{SS} < T_{SS} \\ 1 \ , & \text{if } y_{SS} \geq T_{SS} \end{cases} \tag{2.5.3.6}$$

2.5.4. Calculation of $\bar{C}$ for Gaussian Model

Let's represent the equations derived in the previous sections using Gaussian-distributed data.

$$\begin{aligned} P_{e_{HS}} &= P_r(y_{HS} \geq TU \mid H_0) \cdot P_r(H_0) + P_r(y_{HS} \leq TL \mid H_1) \cdot P_r(H_1) \\ &= P_r(H_0) \cdot \int_{TU}^{\infty} f_{HS_0}(y_{HS})\, dy_{HS} + P_r(H_1) \cdot \int_{-\infty}^{TL} f_{HS_1}(y_{HS}) dy_{HS} \\ &= P_r(H_0) \cdot Q\left[\frac{TU - \mu_{HS_0}}{\sigma_{HS_0}}\right] + P_r(H_1) \cdot \left\{1 - Q\left[\frac{TL - \mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}, \end{aligned} \tag{2.5.4.1}$$

$$\begin{aligned} P_r(z{=}1) &= P_r(TL < y_{HS} < TU \mid H_0) \cdot P_r(H_0) + P_r(TL < y_{HS} < TU \mid H_1) \cdot P_r(H_1) \\ &= P_r(H_0) \cdot \int_{TL}^{TU} f_{HS_0}(y_{HS})\, dy_{HS} + P_r(H_1) \cdot \int_{TL}^{TU} f_{HS_1}(y_{HS})\, dy_{HS} \end{aligned}$$

$$= P_r(H_0) \cdot \left\{ Q\left[\frac{TL - \mu_{HS_0}}{\sigma_{HS_0}}\right] - Q\left[\frac{TU - \mu_{HS_0}}{\sigma_{HS_0}}\right]\right\}$$

$$+ P_r(H_1) \cdot \left\{ Q\left[\frac{TL - \mu_{HS_1}}{\sigma_{HS_1}}\right] - Q\left[\frac{TU - \mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}, \qquad (2.5.4.2)$$

and

$$P_{e_{Team}} = P_r(H_0) \cdot \sum_{U_{ss}=0}^{U_{ss}=1} P_r\{g(y_{HS}) > f(U_{SS}) \text{ and } y_{HS} \in [TL, TU] \mid U_{SS}, H_0\} \cdot P_r(U_{SS} \mid H_0)$$

$$+ P_r(H_1) \cdot \sum_{U_{ss}=0}^{U_{ss}=1} P_r\{g(y_{HS}) < f(U_{SS}) \text{ and } y_{HS} \in [TL, TU] \mid U_{SS}, H_1\} \cdot P_r(U_{SS} \mid H_1)$$

$$= P_r(H_0) \cdot \int_{f(U_{SS}=0)}^{\infty} f_{HS_0}(y_{HS})\, dy_{HS} \cdot \int_{-\infty}^{T_{SS}} f_{SS_0}(y_{SS})\, dy_{SS}$$

$$+ P_r(H_0) \cdot \int_{f(U_{SS}=1)}^{\infty} f_{HS_0}(y_{HS})\, dy_{HS} \cdot \int_{T_{SS}}^{\infty} f_{SS_0}(y_{SS})\, dy_{SS}$$

$$+ P_r(H_1) \cdot \int_{-\infty}^{f(U_{SS}=0)} f_{HS_1}(y_{HS})\, dy_{HS} \cdot \int_{-\infty}^{T_{SS}} f_{SS_1}(y_{SS})\, dy_{SS}$$

$$+ P_r(H_1) \cdot \int_{-\infty}^{f(U_{SS}=1)} f_{HS_1}(y_{HS})\, dy_{HS} \cdot \int_{T_{SS}}^{\infty} f_{SS_1}(y_{SS})\, dy_{SS}$$

$$= P_r(H_0) \cdot Q\left[\frac{f(U_{SS}=0) - \mu_{HS_0}}{\sigma_{HS_0}}\right] \cdot Q\left\{1 - \left[\frac{T_{SS} - \mu_{SS_0}}{\sigma_{SS_0}}\right]\right\}$$

$$+ P_r(H_0) \cdot Q\left[\frac{f(U_{SS}=1) - \mu_{HS_0}}{\sigma_{HS_0}}\right] \cdot Q\left[\frac{T_{SS} - \mu_{SS_0}}{\sigma_{SS_0}}\right]$$

$$+P_r(H_1)\cdot\left\{1-Q\left[\frac{f(U_{SS}=0)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}\cdot\left\{1-Q\left[\frac{T_{SS}-\mu_{SS_1}}{\sigma_{SS_1}}\right]\right\}$$

$$+P_r(H_1)\cdot\left\{1-Q\left[\frac{f(U_{SS}=1)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}\cdot Q\left[\frac{T_{SS}-\mu_{SS_1}}{\sigma_{SS_1}}\right] \qquad (2.5.4.3)$$

Substituting (2.5.4.1), (2.5.4.2), and (2.4.5.3) into (2.3.4), the expected system cost is obtained.

2.6. Numerical Evaluation of $\bar{C}$

In the previous section $\bar{C}$ is represented in terms of Q(y)-function, which makes possible to evaluate $\bar{C}$ numerically. The purpose of the numerical evaluation is to determine the expected system cost at the various thresholds position in the host sensor, meaning the position of TL and TU, so that the optimal thresholds for the system can be realized with different communication cost constants. At the optimal thresholds the expected system cost is minimum.

The *a priori* probabilities of the environment are considered equiprobable, $P_r(H_0) = P_r(H_1) = 0.5$. For the statistics of the observations we take $\sigma_{HS_0} = \sigma_{HS_1} = \sigma_{SS_0} = \sigma_{SS_1} = \sigma = 1$. The mean values of the observations at each sensor are $\mu_{HS_0} = \mu_{SS_0} = -1$ and $\mu_{HS_1} = \mu_{SS_1} = 1$. The communication cost constant is held a constant value until all the expected system costs are evaluated at the desired threshold positions. The thresholds, TL and TU. are varied with a relationship of TU = -TL. This threshold relationship is selected because of the symmetric nature of. Gaussian PDF and its observations. The results from the numerical evaluation of $\bar{C}$ are plotted in Figure 2.3, Figure ?.4, Figure 2.5, and Figure 2.6. For the tabulated data of Figure 2.5, please refer to Table 5.1 in Chapter 5. The computer program is written for (2.3.4), substituted with $P_{e_{HS}}$, $P_r(z=1)$, and $P_{e_{Team}}$, which are expressed in (2.5.4.1), (2.5.4.2), and (2.5.4.3), respectively. The program is listed in Appendix A.

2.6.1. Comments on Numerical Evaluation

The system expected costs are evaluated over the threshold positions on the host sensor's observation space for a given communication cost constant. The following figures are plotted from the results obtained through the numerical evaluation of $\bar{C}$.

Figure 2.3 shows the expected system cost as the threshold is departing from the origin (position 0.0) for each communication cost constant, CCC1. TU moves in the positive direction and TL moves in the negative direction. When CCC1 $\geq$ 0.5, in the curve of the expected system cost vs. HS threshold position, the expected system cost never gets smaller than the cost at threshold position 0.0. The dotted curve indicates the maximum CCC1 which has a minimum other than at the threshold position of 0.0. Figure 2.4 is the enlarged version of Figure 2.3, where the minima of the curves are shown clearly.

Figure 2.5 is a plot of extracted information from Figure 2.3. It shows the behavior of the minimum expected system cost due to the change of the communication cost constant.

Figure 2.6 involves all the vital information obtained in this evaluation. It represents the 2SS's minimum expected system co:t vs. the optimum threshold position and the communication cost constant.

More detailed observation of these data is performed in Chapter 5 where the systems are compared.

Two-Sensor-System

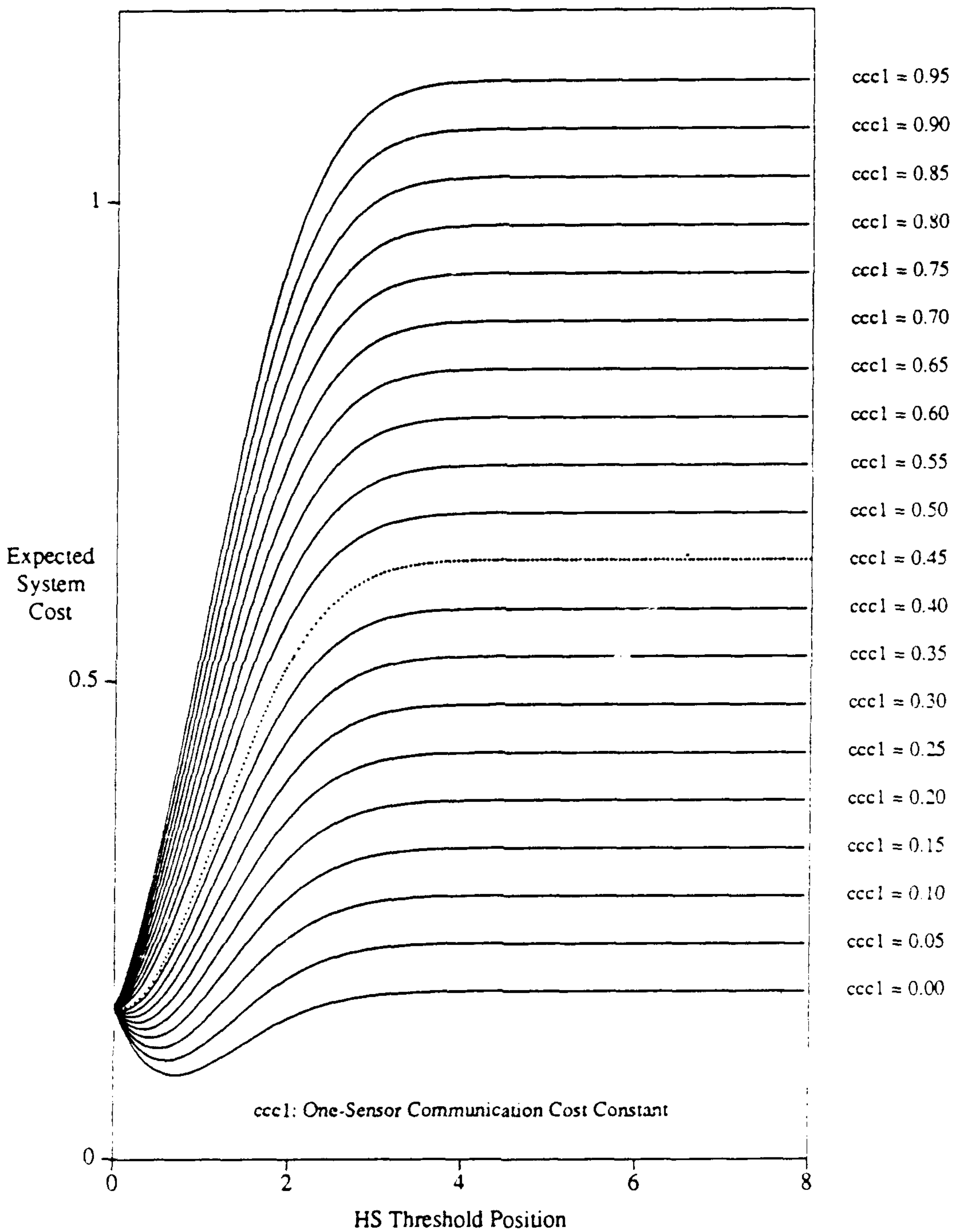

Figure 2.3 Expected System Cost vs. HS Threshold Position

Two-Sensor-System

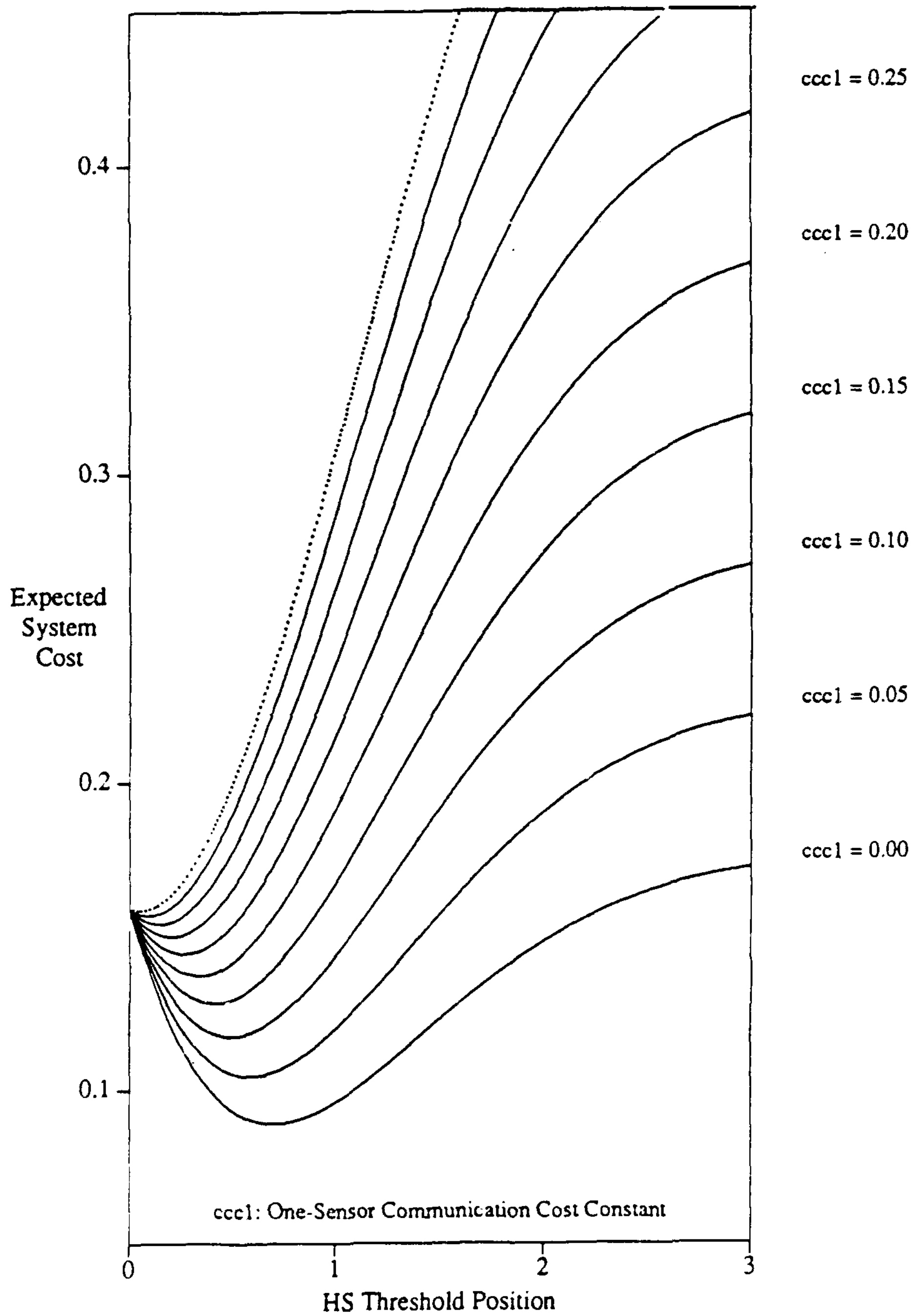

Figure 2.4 Expected System Cost vs. HS Threshold Position (Enlarged Version of Figure 2.3)

Two-Sensor-System

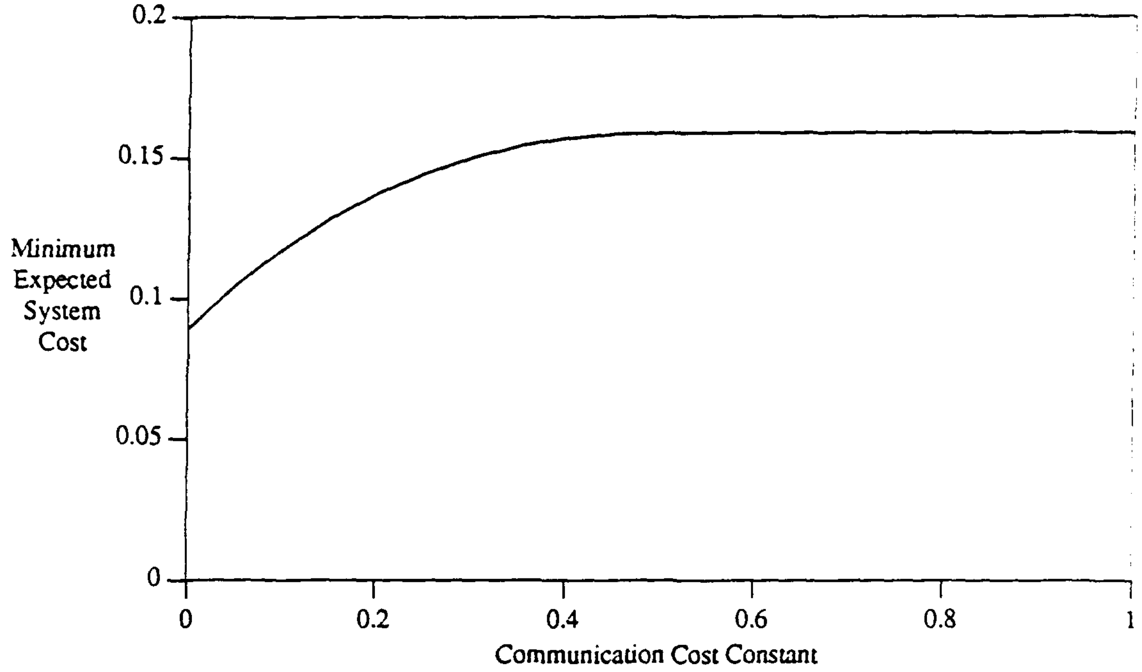

Figure 2.5 Min. Expected System Cost vs. Communication Cost Constant

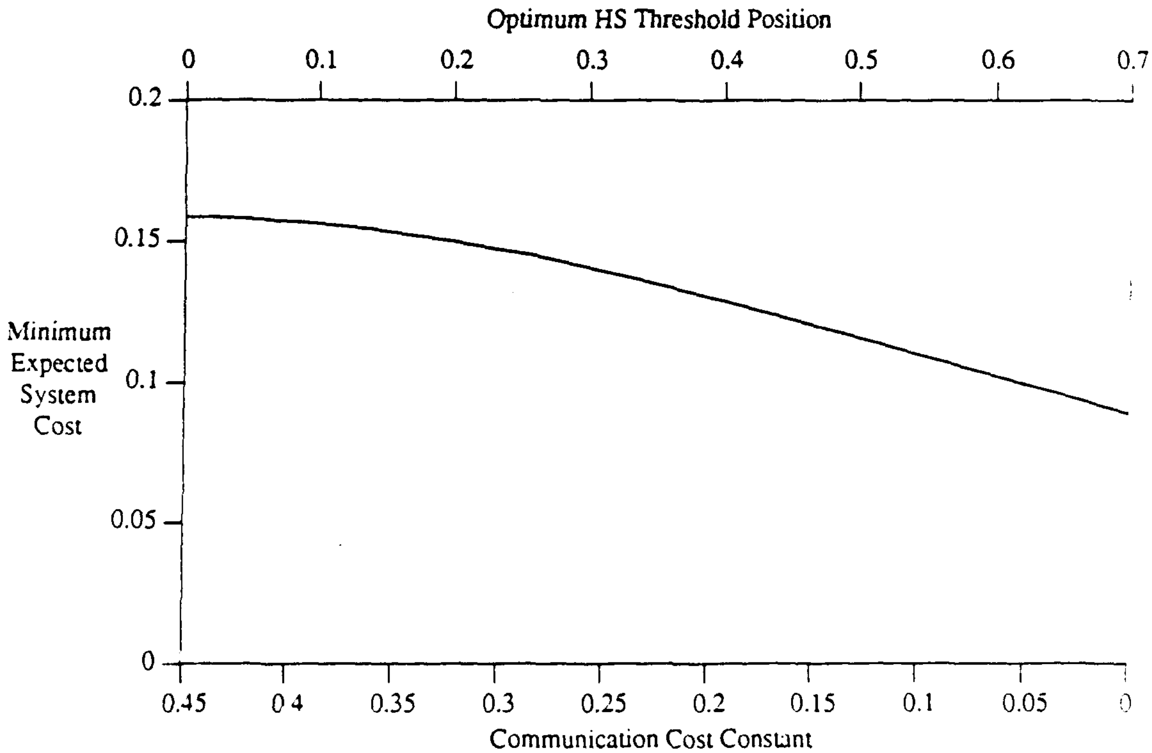

Figure 2.6 Summary of Data

CHAPTER 3

Analysis of a Three-Sensor-System (3SS)

3.1. The Model and Configuration

The environment and other elements in modeling this system are closely related to those in Chapter 2, the two-sensor-system. The only difference in this chapter is that the host sensor communicates with two slave sensors, instead of one, when the host sensor makes an uncertain decision. The system cost is also defined the same way as in (2.2.1) of Chapter 2. Thus, the expected system cost expression is the same as (2.3.4). All the expressions of terms in (2.3.4) are directly applied for this system.

3.1.1. Host Sensor's Decision Boundaries

The design of thresholds in the host sensor in the three-sensor-system is very similarly done as in the two-sensor-system (refer to Figure 3.1). The thresholds divide the observation space into three decision regions, No Target (0), No Decision (?), and Target Detected (1).

3.1.2. The Overall Process of The Model

The host sensor's confident local decision, either 0 or 1, will become the final decision of the system. In case the decision of the host sensor is dubious (y_{HS} falls between TL and TU, or TL3 and TU3 in Figure 3.1), the host sensor will request

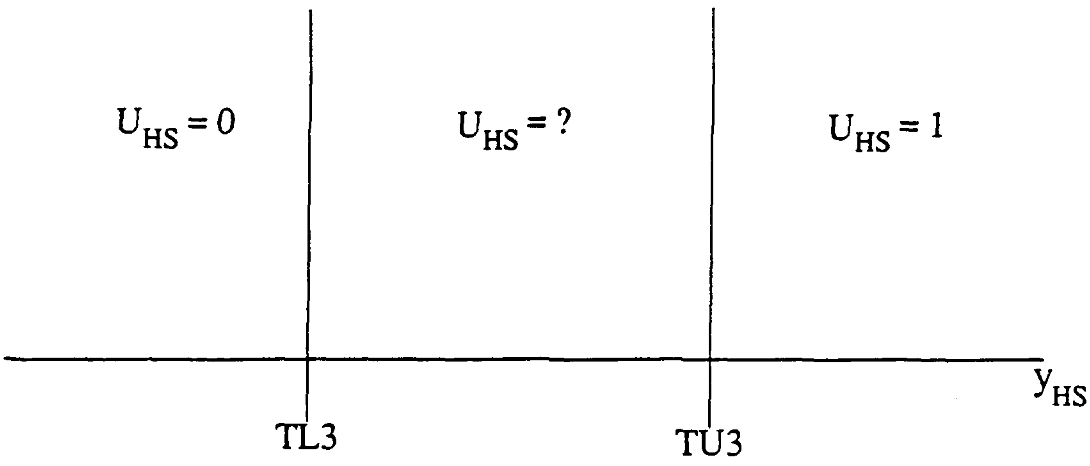

Figure 3.1 The Decision Boundaries of HS for 3-Sensor-System

binary informations, U_{SS1} and U_{SS2}, from both of the slave sensors, which are also independently generated according to their observation, y_{SS1} and y_{SS2}, at the slave sensor 1 (SS1) and the slave sensor 2 (SS2). These communication process also involves a communication cost constant as in 2SS. The illustration of the process is in Figure 3.2.

3.2. Evaluation of Error Caused by Team Strategies

Pe_{Team} of 3SS has the same probabilistic expression as that of 2SS except now the expression is conditioned on two slave sensors, not one. The expression is shown below.

$Pe_{Team} = P_r$(error resulted by communication using team strategy) $= P_r(E)$

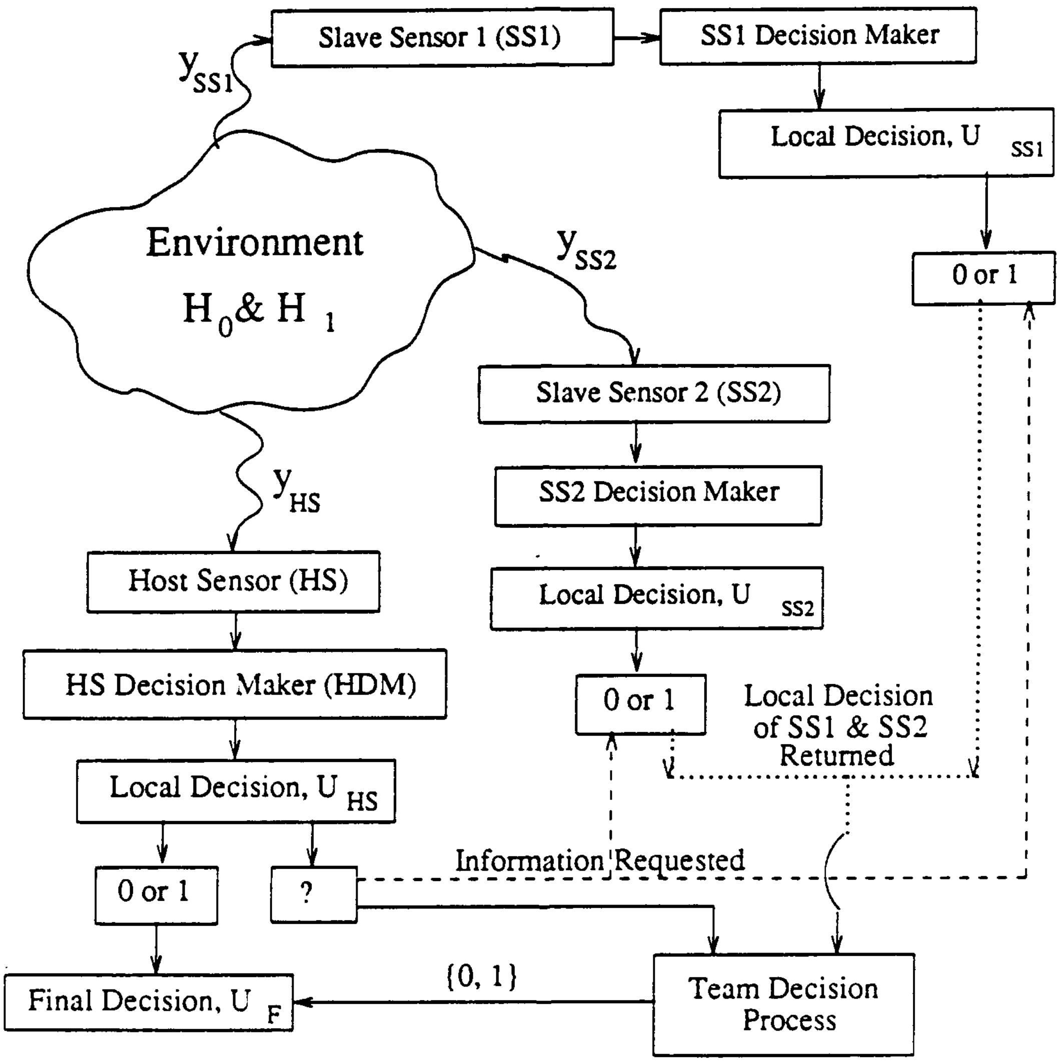

Figure 3.2 Model Configuration of 3SS

$$= P_r(E \mid y_{HS} \in [TL, TU]) \cdot P_r(y_{HS} \in [TL, TU])$$

$$= P_r(E, y_{HS} \in [TL, TU] \mid H_0) \cdot P_r(H_0) + P_r(E, y_{HS} \in [TL, TU] \mid H_1) \cdot P_r(H_1)$$

$$= P_r(E, y_{HS} \in [TL, TU] \mid H_0, U_{SS1}, U_{SS2}) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(U_{SS2} \mid H_0) \cdot P_r(H_0)$$

$$+ P_r(E, y_{HS} \in [TL,TU] \mid H_1, U_{SS1}, U_{SS2}) \cdot P_r(U_{SS1} \mid H_1) \cdot P_r(U_{SS2} \mid H_1) \cdot P_r(H_1) \quad (3.2.1)$$

3.3. The Likelihood Ratio Test

In evaluating the Likelihood Ratio Test (LRT), it is assumed that the individual observations received at each sensor are independent from each other. Thus, their local decisions are also independent from other sensors. In other words, the local decision of the slave sensors is statistically independent from (or not coupled with) either the local decision or the observation of the host sensor.

$$\Lambda(y_{HS}, U_{SS1}, U_{SS2}) = \frac{P_r(y_{HS}, U_{SS1}, U_{SS2} \mid H_1)}{P_r(y_{HS}, U_{SS1}, U_{SS2} \mid H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \quad (3.3.1)$$

Using the above assumption, (3.3.1) can be written as following.

$$\frac{P_r(y_{HS} \mid H_1) \cdot P_r(U_{SS1} \mid H_1) \cdot P_r(U_{SS2} \mid H_1) \cdot P_r(H_1)}{P_r(y_{HS} \mid H_0) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(U_{SS2} \mid H_0) \cdot P_r(H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \quad (3.3.2)$$

$$\frac{P_r(y_{HS} \mid H_1)}{P_r(y_{HS} \mid H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \cdot \frac{P_r(H_0)}{P_r(H_1)} \cdot \frac{P_r(U_{SS1} \mid H_0)}{P_r(U_{SS1} \mid H_1)} \cdot \frac{P_r(U_{SS2} \mid H_0)}{P_r(U_{SS2} \mid H_1)}$$

$$f(U_{SS1}, U_{SS2}) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1} \mid H_0)}{P_r(U_{SS1} \mid H_1)} \cdot \frac{P_r(U_{SS2} \mid H_0)}{P_r(U_{SS2} \mid H_1)} \quad (3.3.3)$$

$$g(y_{HS}) \underset{U_F=0}{\overset{U_F=1}{\gtrless}} f(U_{SS1}, U_{SS2}) \quad (3.3.4)$$

T_0 is a ratio of *a priori* probabilities. The function $f(U_{SS1}, U_{SS2})$ represents the final threshold in the host sensor after communication between the host sensor and the slave sensors. As it is seen in (3.3.3), the function $f(U_{SS1}, U_{SS2})$ can have four different values (thresholds) depending on the decision of the slave sensors, U_{SS1} and U_{SS2}, since the decisions of SS, U_{SS1}, U_{SS2} are always either "0" or "1". This gives more explicit expression of the function $f(U_{SS1}, U_{SS2})$ which is listed below.

$$f(U_{SS1}=0, U_{SS2}=0) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1}=0 \mid H_0)}{P_r(U_{SS1}=0 \mid H_1)} \cdot \frac{P_r(U_{SS2}=0 \mid H_0)}{P_r(U_{SS2}=0 \mid H_1)} \tag{3.3.5}$$

$$f(U_{SS1}=0, U_{SS2}=1) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1}=0 \mid H_0)}{P_r(U_{SS1}=0 \mid H_1)} \cdot \frac{P_r(U_{SS2}=1 \mid H_0)}{P_r(U_{SS2}=1 \mid H_1)} \tag{3.3.6}$$

$$f(U_{SS1}=1, U_{SS2}=0) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1}=1 \mid H_0)}{P_r(U_{SS1}=1 \mid H_1)} \cdot \frac{P_r(U_{SS2}=0 \mid H_0)}{P_r(U_{SS2}=0 \mid H_1)} \tag{3.3.7}$$

$$f(U_{SS1}=1, U_{SS2}=1) = \lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1}=1 \mid H_0)}{P_r(U_{SS1}=1 \mid H_1)} \cdot \frac{P_r(U_{SS2}=1 \mid H_0)}{P_r(U_{SS2}=1 \mid H_1)} \tag{3.3.8}$$

Using (3.3.4), we can write the final decision, U_F, rules of the system as

$$U_F = \begin{cases} 1 & , \text{ if } y_{HS} \geq TU \\ 1 & , \text{ if } g(y_{HS}) \geq f(U_{SS1}, U_{SS2}) \\ 0 & , \text{ if } g(y_{HS}) < f(U_{SS1}, U_{SS2}) \\ 0 & , \text{ if } y_{HS} < TL \end{cases}$$

Then, it is possible to express $P_{e_{Team}}$ as shown below.

$$P_r(E, y_{HS} \in [TL, TU])$$

$$= P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r(E, y_{HS} \in [TL, TU] \mid U_{SS1}, U_{SS2}, H_0) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(U_{SS2} \mid H_0)$$

$$+ P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r(E, y_{HS} \in [TL, TU] \mid U_{SS1}, U_{SS2}, H_1) \cdot P_r(U_{SS2} \mid H_1) \cdot P_r(U_{SS2} \mid H_1)$$

$$= P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r\{g(y_{HS}) \overset{U_F=1}{>}$$

$$f(U_{SS1}, U_{SS2}) \text{ and } y_{HS} \in [TL, TU] \mid U_{SS1}, U_{SS2}, H_0\} \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(U_{SS2} \mid H_0)$$

$$+ P_r(H_1) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r\{g(y_{HS}) \overset{U_F=0}{>}$$

$$f(U_{SS1}, U_{SS2}) \text{ and } y_{HS} \in [TL, TU] \mid U_{SS1}, U_{SS2}, H_1\} \cdot P_r(U_{SS1} \mid H_1) \cdot P_r(U_{SS2} \mid H_1) \quad (3.3.9)$$

3.4. Calculation of $\overline{C}$ for Gaussian Models

As in Chapter 2, the Gaussian distribution function is used to give examples in expressing $\overline{C}$ so that it can be evaluated numerically.

3.4.1. Gaussian Probability Density Function

The probability density functions are shown below. They show PDF of "0" and "1" at HS, SS1, and SS2.

$$f_{HS_0}(y_{HS}) = \frac{1}{\sqrt{2\pi}\,\sigma_{HS_0}} e^{\frac{-(y_{HS}-\mu_{HS_0})^2}{2\sigma^2_{HS_0}}} \quad (3.4.1.1)$$

$$f_{HS_1}(y_{HS}) = \frac{1}{\sqrt{2\pi}\,\sigma_{HS_1}} e^{\frac{-(y_{HS}-\mu_{HS_1})^2}{2\sigma^2_{HS_1}}} \quad (3.4.1.2)$$

$$f_{SS1_0}(y_{SS1}) = \frac{1}{\sqrt{2\pi}\,\sigma_{SS1_0}} e^{\frac{-(y_{SS1}-\mu_{SS1_0})^2}{2\sigma^2_{SS1_0}}} \quad (3.4.1.3)$$

$$f_{SS1_1}(y_{SS1}) = \frac{1}{\sqrt{2\pi}\,\sigma_{SS1_1}} e^{\frac{-(y_{SS1}-\mu_{SS1_1})^2}{2\sigma^2_{SS1_1}}} \quad (3.4.1.4)$$

$$f_{SS2_0}(y_{SS2}) = \frac{1}{\sqrt{2\pi}\,\sigma_{SS2_0}} e^{\frac{-(y_{SS2}-\mu_{SS2_0})^2}{2\sigma^2_{SS2_0}}} \quad (3.4.1.5)$$

$$f_{SS2_1}(y_{SS2}) = \frac{1}{\sqrt{2\pi}\,\sigma_{SS2_1}} e^{\frac{-(y_{SS2}-\mu_{SS2_1})^2}{2\sigma_{SS2_1}^2}} \tag{3.4.1.6}$$

3.4.2. Decision Boundary of SS1 & SS2

An equation of the decision boundary for the SSs was derived, (2.5.3.5), in Chapter 2. The analytical expression of threshold in SS1 and SS2 follows the same as in Chapter 2.

$$T_{SS1} = \frac{\mu_{SS1_0} + \mu_{SS1_1}}{2} + \frac{\sigma^2}{\mu_{SS1_1} - \mu_{SS1_0}} \cdot \log_e\left\{ \lambda_{t_{SS_1}} \cdot \frac{P_r(H_0)}{P_r(H_1)} \right\} \tag{3.4.2.1}$$

$$T_{SS2} = \frac{\mu_{SS2_0} + \mu_{SS2_1}}{2} + \frac{\sigma^2}{\mu_{SS2_1} - \mu_{SS2_0}} \cdot \log_e\left\{ \lambda_{t_{SS_2}} \cdot \frac{P_r(H_0)}{P_r(H_1)} \right\} \tag{3.4.2.2}$$

Then, the decisions at the slave sensors are stated as below:

For SS1,

$$U_{SS1} = \begin{cases} 0 \;, & \text{if } y_{SS1} < T_{SS1} \\ 1 \;, & \text{if } y_{SS1} \geq T_{SS1} \end{cases} \tag{3.4.2.3}$$

and for SS2,

$$U_{SS2} = \begin{cases} 0 \;, & \text{if } y_{SS2} < T_{SS2} \\ 1 \;, & \text{if } y_{SS2} \geq T_{SS2} \end{cases} \tag{3.4.2.4}$$

Since the probability of error incurred by communication and the final decision boundaries are different from Chapter 2, they are written in this section. The Gaussian expression of the FTs are written:

$$f(U_{SS1}=0,\, U_{SS2}=0) = \lambda_t \cdot T_0 \cdot \frac{1-Q\left[\dfrac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]}{1-Q\left[\dfrac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]} \cdot \frac{1-Q\left[\dfrac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]}{1-Q\left[\dfrac{T_{SS1}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]} \tag{3.4.2.5}$$

$$f(U_{SS1}=0,\, U_{SS2}=1) = \lambda_t \cdot T_0 \cdot \frac{1-Q\left[\dfrac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]}{Q\left[\dfrac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]} \cdot \frac{1-Q\left[\dfrac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]}{Q\left[\dfrac{T_{SS1}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]} \tag{3.4.2.6}$$

$$f(U_{SS1}=1,\, U_{SS2}=0) = \lambda_t \cdot T_0 \cdot \frac{Q\left[\dfrac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]}{1-Q\left[\dfrac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]} \cdot \frac{Q\left[\dfrac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]}{1-Q\left[\dfrac{T_{SS1}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]} \tag{3.4.2.7}$$

$$f(U_{SS1}=1,\, U_{SS2}=1) = \lambda_t \cdot T_0 \cdot \frac{Q\left[\dfrac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]}{Q\left[\dfrac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]} \cdot \frac{Q\left[\dfrac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]}{Q\left[\dfrac{T_{SS1}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]} \tag{3.4.2.8}$$

The corresponding probability of error in the team process is, then,

$$P_{e_{Team}} = P_r(\text{error resulted by communicating with all SSs using team strategy})$$

$$= P_r(H_0) \cdot \left\{ \int_{-\infty}^{T_{SS1}} f_{SS1_0}(y_{SS1})dy_{SS1} \cdot \int_{-\infty}^{T_{SS2}} f_{SS2_0}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=0,\, U_{SS2}=0)}^{\infty} f_{HS_0}(y_{HS})dy_{HS} \right.$$

$$+ \int_{-\infty}^{T_{SS1}} f_{SS1_0}(y_{SS1})dy_{SS1} \cdot \int_{T_{SS2}}^{\infty} f_{SS2_0}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=0,\, U_{SS2}=1)}^{\infty} f_{HS_0}(y_{HS})dy_{HS}$$

$$+\int_{T_{SS1}}^{\infty} f_{SS1_0}(y_{SS1})dy_{SS1} \cdot \int_{-\infty}^{T_{SS2}} f_{SS2_0}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=1,\,U_{SS2}=0)}^{\infty} f_{HS_0}(y_{HS})dy_{HS}$$

$$\left. +\int_{T_{SS1}}^{\infty} f_{SS1_0}(y_{SS1})dy_{SS1} \cdot \int_{T_{SS2}}^{\infty} f_{SS2_0}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=1,\,U_{SS2}=1)}^{\infty} f_{HS_0}(y_{HS})dy_{HS} \right\}$$

$$+P_r(H_1) \cdot \left\{ \int_{-\infty}^{T_{SS1}} f_{SS1_1}(y_{SS1})dy_{SS1} \cdot \int_{-\infty}^{T_{SS1}} f_{SS2_1}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=0,\,U_{SS2}=0)}^{\infty} f_{HS_1}(y_{HS})dy_{HS} \right.$$

$$+\int_{-\infty}^{T_{SS1}} f_{SS1_1}(y_{SS1})dy_{SS1} \cdot \int_{T_{SS2}}^{\infty} f_{SS2_1}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=0,\,U_{SS2}=1)}^{\infty} f_{HS_1}(y_{HS})dy_{HS}$$

$$+\int_{T_{SS1}}^{\infty} f_{SS1_1}(y_{SS1})dy_{SS1} \cdot \int_{-\infty}^{T_{SS2}} f_{SS2_1}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=1,\,U_{SS2}=0)}^{\infty} f_{HS_1}(y_{HS})dy_{HS}$$

$$\left. +\int_{T_{SS1}}^{\infty} f_{SS1_1}(y_{SS1})dy_{SS1} \cdot \int_{T_{SS2}}^{\infty} f_{SS2_1}(y_{SS2})dy_{SS2} \cdot \int_{f(U_{SS1}=1,\,U_{SS2}=1)}^{\infty} f_{HS_1}(y_{HS})dy_{HS} \right\}$$

$$=P_r(H_0) \cdot \left[\left[\left\{ 1-Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\right\} \cdot \left\{ 1-Q\left[\frac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\right\} \cdot Q\left[\frac{f(U_{SS1}=0,\,U_{SS2}=0)-\mu_{HS_0}}{\sigma_{HS_0}}\right] \right.\right.$$

$$+\left\{ 1-Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\right\} \cdot Q\left[\frac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right] \cdot Q\left[\frac{f(U_{SS1}=0,\,U_{SS2}=1)-\mu_{HS_0}}{\sigma_{HS_0}}\right]$$

$$+Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right] \cdot \left\{ 1-Q\left[\frac{T_{SS1}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\right\} \cdot Q\left[\frac{f(U_{SS1}=1,\,U_{SS2}=0)-\mu_{HS_0}}{\sigma_{HS_0}}\right]$$

$$
\begin{aligned}
&+Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\cdot Q\left[\frac{f(U_{SS1}=1,U_{SS2}=1)-\mu_{HS_0}}{\sigma_{HS_0}}\right]\Bigg]\\
&+P_r(H_1)\cdot\Bigg[\Bigg\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\Bigg\}\cdot\Bigg\{1-Q\left[\frac{T_{SS2}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\Bigg\}\cdot\Bigg\{1-Q\left[\frac{f(U_{SS1}=0,U_{SS2}=0)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\Bigg\}\\
&+\Bigg\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\Bigg\}\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\cdot\Bigg\{1-Q\left[\frac{f(U_{SS1}=0,U_{SS2}=1)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\Bigg\}\\
&+Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\cdot\Bigg\{1-Q\left[\frac{T_{SS1}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\Bigg\}\cdot\Bigg\{1-Q\left[\frac{f(U_{SS1}=1,U_{SS2}=0)-\mu_{HS_0}}{\sigma_{HS_1}}\right]\Bigg\}\\
&+Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\cdot\Bigg\{1-Q\left[\frac{f(U_{SS1}=1,U_{SS2}=1)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\Bigg\}\Bigg] \qquad (3.4.2.9)
\end{aligned}
$$

Thus, $\overline{C}$ of 3SS in the Gaussian case is obtained by substituting (3.4.2.9), (2.5.4.1), and (2.5.4.2) into (2.3.4). A program listing is attached in Appendix B.

3.4.3. Numerical Evaluation of $\overline{C}$

The method of performing the numerical evaluation for $\overline{C}$ is quite similar to those done in Chapter 2. The parameter values used in this section are as follows; All the standard deviations, σ, are set to 1.0, the mean of "0" observation is -1.0, and the mean of "1" observation is 1.0. The communication cost constant is varied from 0.0 to 1.0 in steps of 0.05. The thresholds in the host sensor are moved away from the origin. TL moves to the negative direction and TU moves to the positive direction with the relationship of TU = -TL.

3.4.4. Comments on Numerical Evaluation

The results of the numerical evaluation are plotted in Figure 3.3, Figure 3.4 and Figure 3.5. Figure 3.3 shows that the curves of the system expected cost over the threshold positions with different communication cost constant (CCC1). When CCC1 is greater or equal to 0.55, the curves are monotonically increasing, giving minima at the threshold position of 0.0. Figure 3.4 is an enlarged version of Figure 3.3 which shows the minima of the curves with CCC1 less than 0.55 clearly. Figure 3.5 can be interpreted that the minimum expected system cost increases as CCC1 increases; however, near CCC1 = 0.5, the minimum expected system cost tends to be flattening since the thresholds (TL and TU) are collapsed into the threshold position of 0.0. This is shown in Figure 3.6 where we plot the minimum expected system cost vs. communication cost constant and vs. the optimal threshold position. More discussion of the results are carried in Chapter 5.

Three-Sensor-System

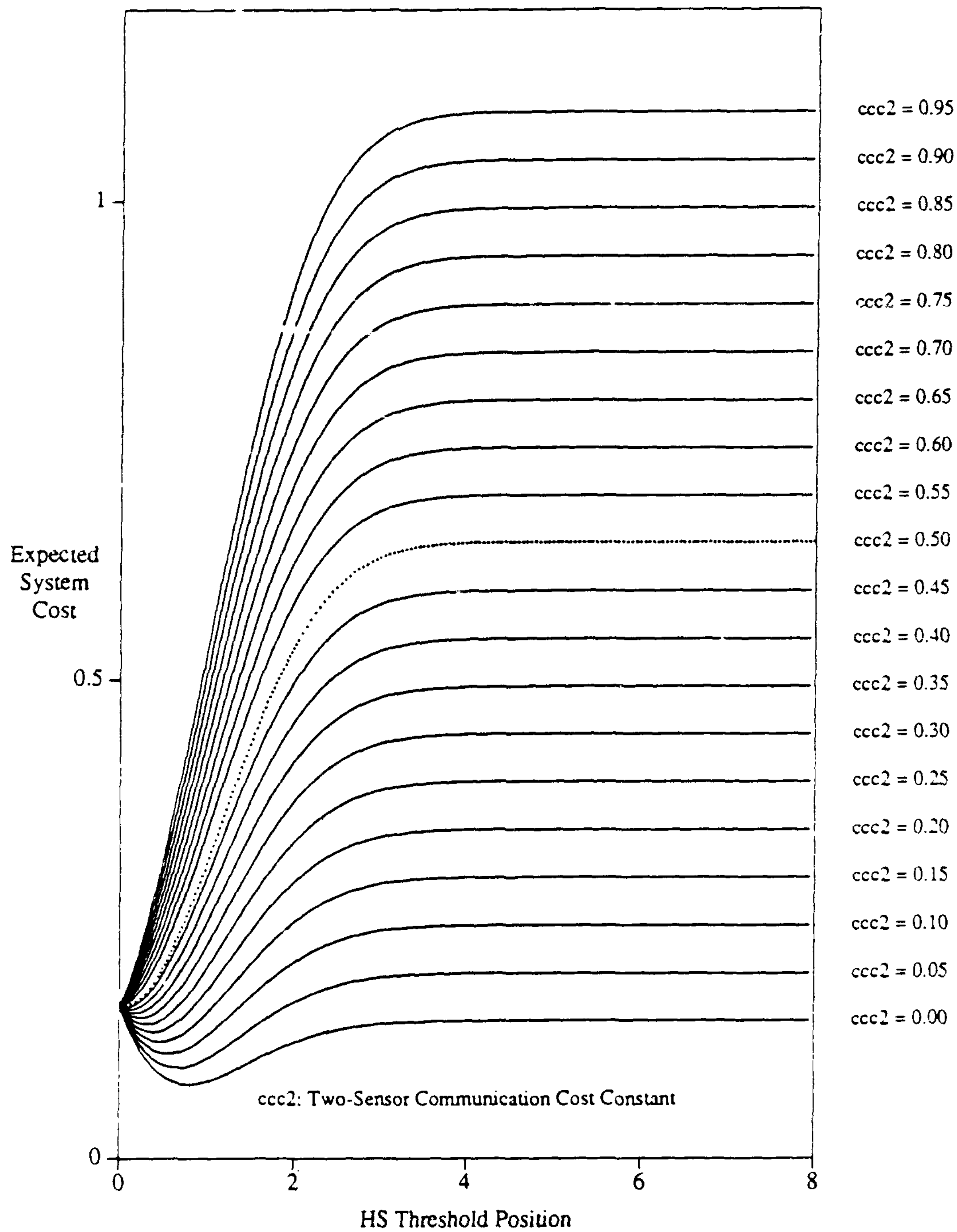

Figure 3.3 Expected System Cost vs. HS Threshold Position

Three-Sensor-System

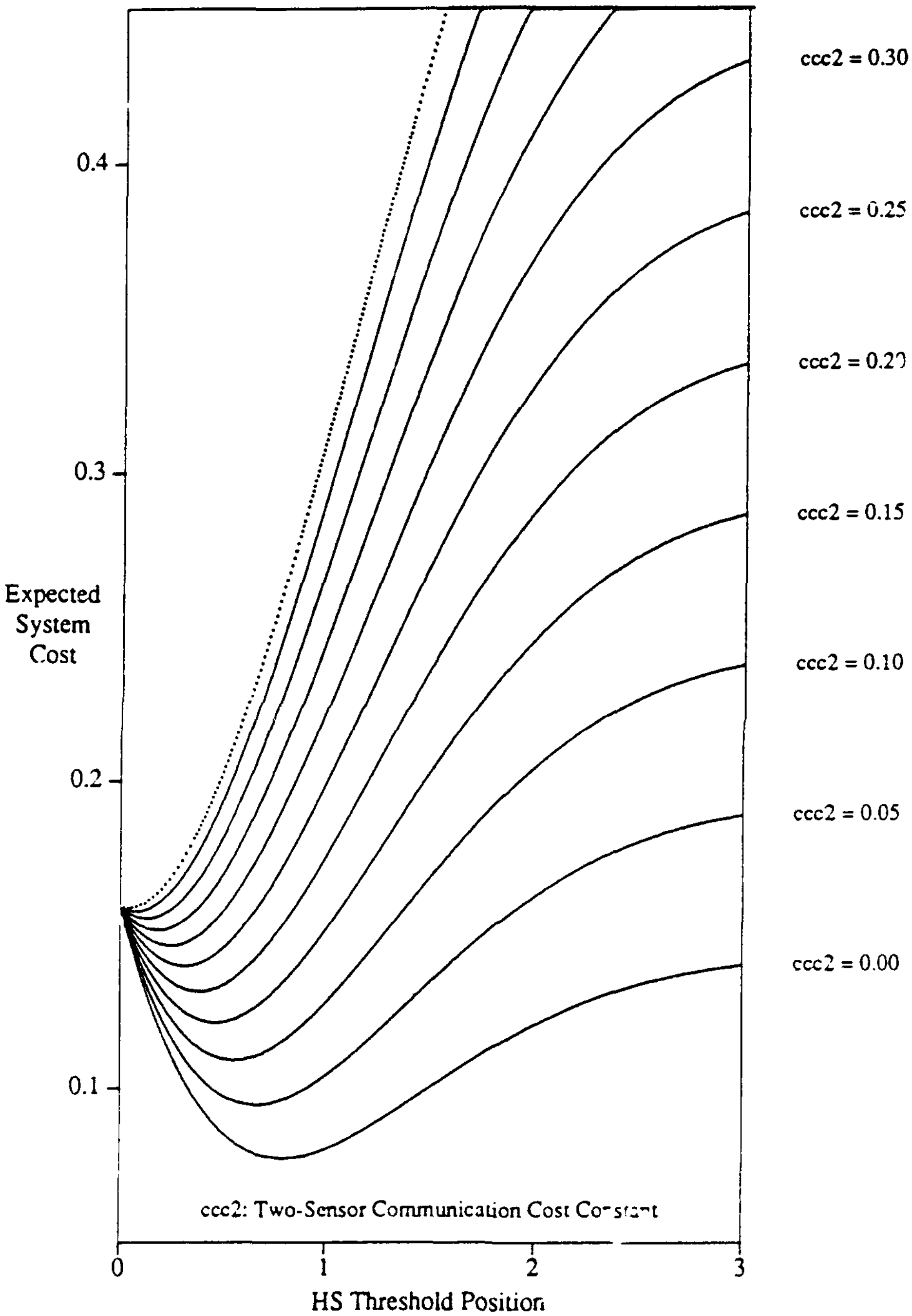

Figure 3.4 Expected System Cost vs. HS Threshold Position
(Enlarged Version of Figure 3.3)

Three-Sensor-System

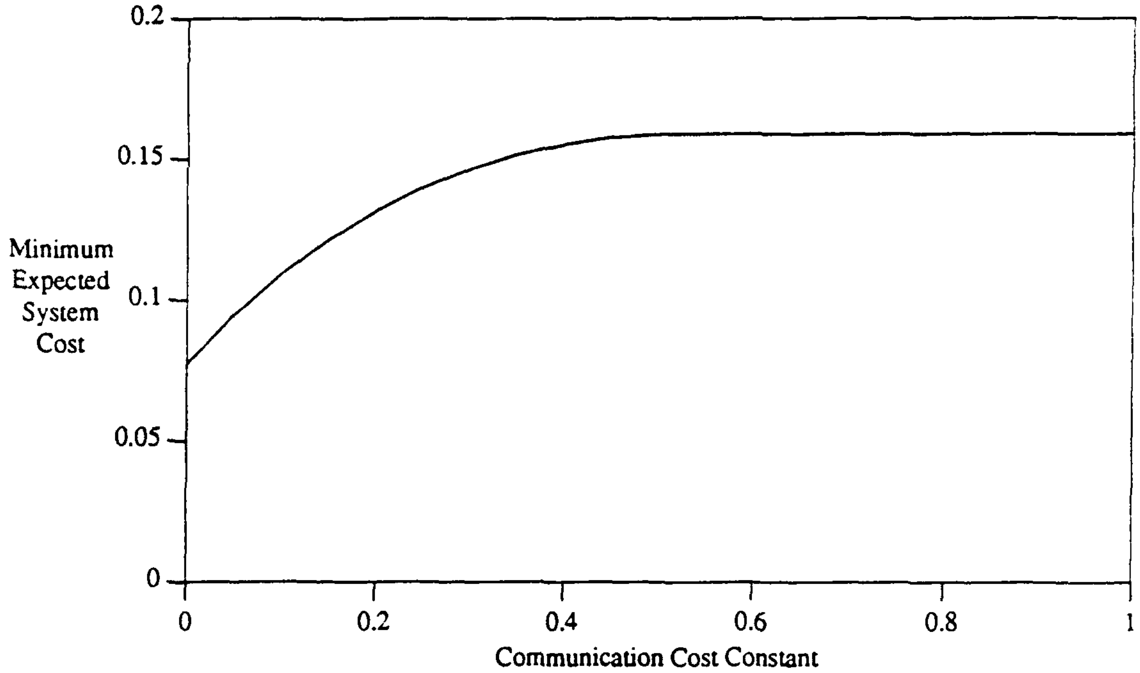

Figure 3.5 Min. Expected System Cost vs. Communication Cost Constant

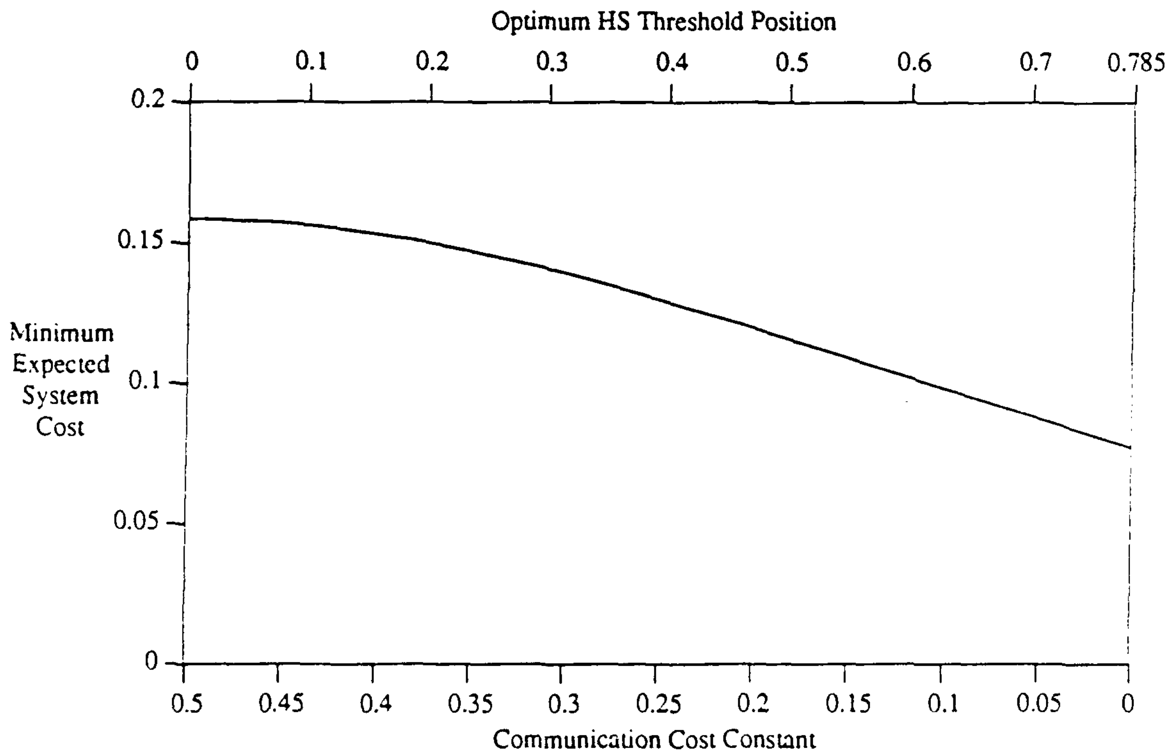

Figure 3.6 Summary of Data

CHAPTER 4

Analysis of a Two/Three-Sensor-System (2/3SS)

4.1. The Model and Configuration

The difference between this chapter and the previous chapters is that here the host sensor chooses a communication scheme, based on quality of its own decision. For example, when the host sensor's observation, y_{HS}, falls in a certain region of uncertainty, it communicates with only one slave sensor. It communicates with two slave sensors when the observation falls in the other region of uncertainty. Contrary to the previous chapters, two different communication cost constants are considered; one for communicating with one slave sensor, and another for communicating with two slave sensors.

4.1.1. The Host Sensor's Thresholds and Decision Regions

In the host sensor's observation space, there are four thresholds that divide the space into four decision regions. (Actually, there are five decision regions but two out of five regions yield the same decision.) When y_{HS} falls below TL1 (TL31 in Figure 4.1) or above TU1 (TU31 in Figure 4.1), the host sensor decides 0 or 1, respectively. When the observation falls between TL1 and TL2 (TL31 and TL32 in Figure 4.1) or between TU2 and TU1 (TU32 and TU31 in Figure 4.1), the host sensor's decision, U_{HS}, becomes uncertain (?1). In case of the observation lies between TL2 and TU2 (TL32 and TU32 in Figure 4.1), the host sensor makes a dubious decision (?2). Thus there are level of confidence in making uncertain decision. In ?1 decision region, the probability of making a correct decision is much greater than the probability of making a false decision; then, a minimum help from the slave sensors is needed. In ?2 decision region, the probability of making a correct decision and the probability of making a false detection are compatible; thus, this situation requires more information to make a correct decision.

When the host sensor makes a binary decision (either 0 or 1), it becomes the final decision of the system. In case the decision of the host sensor is ?1, the host sensor requests information only from one of the slave sensors, say slave sensor 1 (SS1). On the other hand, when the the host sensor determines ?2, it asks an assistance from both of the slave sensors, slave sensor 1 (SS1) and slave sensor 2 (SS2).

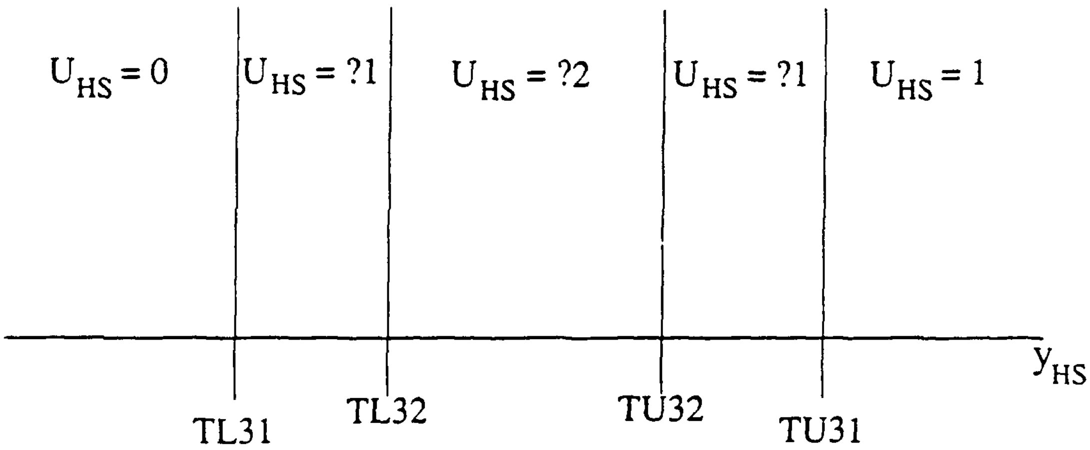

Figure 4.1 Decision Boundaries of HS for 2/3SS

This process is illustrated in Figure 4.2.

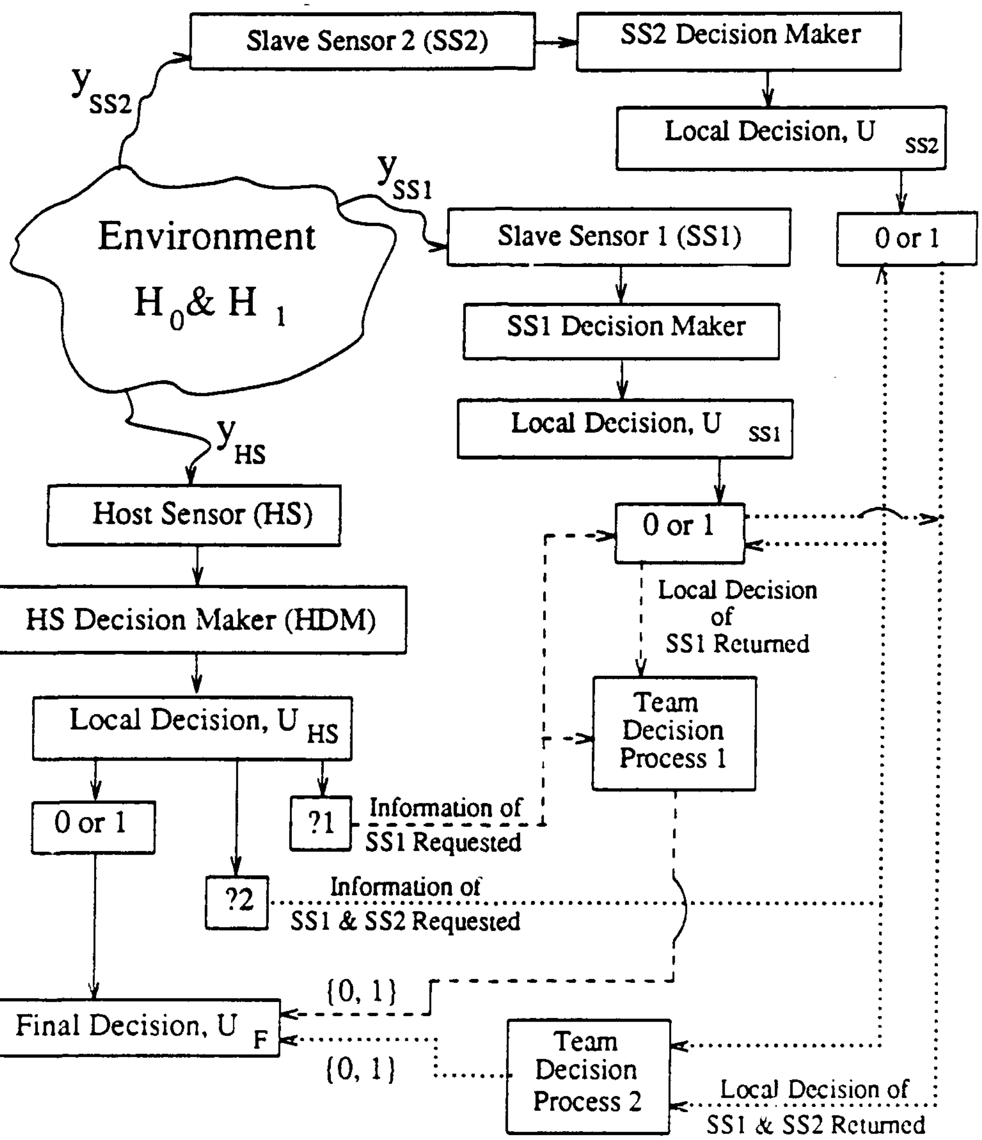

Figure 4.2 Model Configuration of 2/3SS

4.2. Definition of the System Cost Function

The following is the cost function of the system, C(,,,,), which is represented by error probabilities of the individual sensors. For descriptions of symbols used in this chapter, please refer to the beginning of this paper under "Symbols used in Chapter 4".

$$
\begin{aligned}
C(f) &= C(z_{T1}, z_{T2}, TL1, TL2, TU2, TU1, c_{T1}, c_{T2}) \\
&= (1 - z_{T1}) \cdot (1 - z_{T2}) \cdot P_{e_{HS}} \\
&\quad + z_{T1} \cdot (1 - z_{T2}) \cdot (P_{e_{T1}} + c_{T1}) \\
&\quad\quad + z_{T2} \cdot (1 - z_{T1}) \cdot (P_{e_{T2}} + c_{T2})
\end{aligned} \tag{4.2.1}
$$

As shown in the above equation, the communication schemes are dependent upon the values of z_{T1} and z_{T2}. z_{T1} *and* z_{T2} *take binary numbers* depending on the type of host sensor's uncertain decision. When U_{HS} = ?1, z_{T1} becomes 1. When U_{HS} = ?2, z_{T2} becomes 1. This is shown in the table below.

Communication Scheme		
	z_{T1}	z_{T2}
No Communication	0	0
Communication with SS1	1	0
Communication with SS1 & SS2	0	1

Table 4.1 Communication Scheme of 2/3SS

4.3. Evaluation of an Expected System's Total Cost, $\bar{C}$

Let's evaluate the expected cost of the system.

$$\bar{C} = E\{C(f)\}$$

$$= C(f)|_{z_{T1}=0,\, z_{T2}=0} \cdot P_r(z_{T1}=0) \cdot P_r(z_{T2}=0)$$

$$+ C(f)|_{z_{T1}=1,\, z_{T2}=0} \cdot P_r(z_{T1}=1) \cdot P_r(z_{T2}=0)$$

$$+ C(f)|_{z_{T1}=1,\, z_{T2}=1} \cdot P_r(z_{T1}=1) \cdot P_r(z_{T2}=1)$$

$$= P_{e_{HS}} \cdot P_r(z_{T1}=0) \cdot P_r(z_{T2}=0)$$

$$+ (P_{e_{T1}} + c_{T1}) \cdot P_r(z_{T1}=1) \cdot P_r(z_{T2}=0)$$

$$+ (P_{e_{T2}} + c_{T2}) \cdot P_r(z_{T1}=0) \cdot P_r(z_{T2}=1)$$

$$= P_{e_{HS}} + (P_{e_{T1}} + c_{T1} - P_{e_{HS}}) \cdot P_r(z_{T1}=1) + (P_{e_{T2}} + c_{T2} - P_{e_{HS}}) \cdot P_r(z_{T2}=1)$$

$$+ (P_{e_{HS}} - P_{e_{T1}} - P_{e_{T2}} - c_{T1} - c_{T2}) \cdot P_r(z_{T1}=1) \cdot P_r(z_{T2}=1) \qquad (4.3.1)$$

The terms, $P_{e_{HS}}$, $P_r(z_{T1}=1)$, and $P_r(z_{T2}=1)$, are written in generalized probabilistic expressions:

$$P_{e_{HS}} = P_r(\text{false local decision at HS})$$

$$= P_r(\text{Decide } H_1 \mid H_0) \cdot P_r(H_0) + P_r(\text{Decide } H_0 \mid H_1) \cdot P_r(H_1)$$

$$= P_r(y_{HS} \geq TU1 \mid H_0) \cdot P_r(H_0) + P_r(y_{HS} \leq TL1 \mid H_1) \cdot P_r(H_1) \qquad (4.3.2)$$

$$P_r(z_{T1}=1) = P_r(\text{uncertain decision} = ?1\text{; communication channel open only with SS1})$$

$$= P_r(TL1 < y_{HS} < TL2) + P_r(TL2 < y_{HS}\ TU1)$$

$$= P_r(H_0) \cdot \{P_r(TL1 < y_{HS} < TL2 \mid H_0) \cdot P_r(H_0) + P_r(TL2 < y_{HS} < TU1 \mid H_0)\}$$

$$+ P_r(H_1) \cdot \{P_r(TL1 < y_{HS} < TL2 \mid H_1) \cdot P_r(H_0) + P_r(TL2 < y_{HS} < TU1 \mid H_1)\} \qquad (4.3.3)$$

$$P_r(z_{T2}=1) = P_r(\text{uncertain decision} = ?2\text{; communication channel open with SS1 \& SS2})$$

$$= P_r(TL2 < y_{HS} < TU2)$$

$$= P_r(TL2 < y_{HS} < TU2 \mid H_0) \cdot P_r(H_0) + P_r(TL2 < y_{HS} < TU2 \mid H_1) \cdot P_r(H_1) \qquad (4.3.4)$$

There are two different costs (probability of error) incurred in communication, $P_{e_{T1}}$ and $P_{e_{T2}}$, since the system has two different modes of communication, communicating with one slave sensor (SS1), or with two slave sensors (SS1 and SS2), respectively.

$$
\begin{aligned}
P_{e_{T1}} &= P_r(\text{error resulting after communication with SS1 only}) = P_r(E1) \\
&= P_r(E1 \mid y_{HS} \in [TL1,TL2] \text{ or } [TU2,TU1]) \cdot P_r(y_{HS} \in [TL1,TL2] \text{ or } [TU2,TU1]) \\
&= P_r(E1,\ y_{HS} \in [TL1,TL2] \text{ or } [TU2,TU1] \mid H_0) \cdot P_r(H_0) \\
&\quad + P_r(E1,\ y_{HS} \in [TL1,TL2] \text{ or } [TU2,TU1] \mid H_1) \cdot P_r(H_1) \\
&= P_r(E1,\ y_{HS} \in [TL1,TL2] \text{ or } [TU2,TU1] \mid H_0, U_{SS1}) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(H_0) \\
&\quad + P_r(E1,\ y_{HS} \in [TL1,TL2] \text{ or } [TU2,TU1] \mid H_1, U_{SS1}) \cdot P_r(U_{SS1} \mid H_1) \cdot P_r(H_1)
\end{aligned} \tag{4.3.5}
$$

$$
\begin{aligned}
P_{e_{T2}} &= P_r(\textit{Error resulting after communication with SS1 \& SS2}) = P_r(E2) \\
&= P_r(E2 \mid y_{HS} \in [TL2,TU2]) \cdot P_r(y_{HS} \in [TL2,TU2]) \\
&= P_r(E2,\ y_{HS} \in [TL2,TU2] \mid H_0) \cdot P_r(H_0) + P_r(E2,\ y_{HS} \in [TL2,TU2] \mid H_1) \cdot P_r(H_1) \\
&= P_r(E2,\ y_{HS} \in [TL2,TU2] \mid H_0, U_{SS1}, U_{SS2}) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(U_{SS2} \mid H_0) \cdot P_r(H_0) \\
&\quad + P_r(E2,\ y_{HS} \in [TL2,TU2] \mid H_1, U_{SS1}, U_{SS2}) \cdot P_r(U_{SS1} \mid H_1) \cdot P_r(U_{SS2} \mid H_1) \cdot P_r(H_1)
\end{aligned} \tag{4.3.6}
$$

4.4. The Likelihood Ratio Test

In this chapter, it is necessary to evaluate two kinds of LRT, since the LRT for the different communication schemes differs. These evaluations closely follow those derived in Chapters 2 and 3.

4.4.1. LRT for Communicating with SS1 Only

Let LR of this case be

$$\Lambda_{T1}(y_{HS}, U_{SS1}) = \frac{P_r(y_{HS}, U_{SS1} \mid H_1)}{P_r(y_{HS}, U_{SS1} \mid H_0)} \tag{4.4.1.1}$$

Since the observations received at different sensors are mutually independent, (4.4.1.1) can be written as

$$\Lambda_{T1}(y_{HS}, U_{SS1}) = \frac{P_r(y_{HS} \mid H_1) \cdot P_r(U_{SS1} \mid H_1) \cdot P_r(H_1)}{P_r(y_{HS} \mid H_0) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \tag{4.4.1.2}$$

Thus,

$$\frac{P_r(y_{HS} \mid H_1)}{P_r(y_{HS} \mid H_0)} \underset{U_F=0}{\overset{U_F=1}{\gtrless}} \lambda_t \cdot \frac{P_r(H_0)}{P_r(H_1)} \cdot \frac{P_r(U_{SS1} \mid H_0)}{P_r(U_{SS1} \mid H_1)} \tag{4.4.1.3}$$

Recalling the definitions made in Chapter 2, (2.4.5), (2.4.6), and (2.4.7), then (4.4.1.3) can be written as below, provided that we substitute $g(y_{HS}) = g_{T1}(y_{HS})$ and $f(U_{SS}) = f(U_{SS1})$.

$$g_{T1}(y_{HS}) \underset{U_F=0}{\overset{U_F=1}{\gtrless}} f(U_{SS1}) \tag{4.4.1.4}$$

The function $f(U_{SS1})$ represents the final threshold at the host sensor after communication with one slave sensor, SS1. $f(U_{SS1})$ can be two different values (thresholds) depending on the decision of the slave sensor, U_{SS1}. More explicit expression of the function $f(U_{SS1})$ is listed below.

$$f(U_{SS1}=0)=\lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1}=0 \mid H_0)}{P_r(U_{SS1}=0 \mid H_1)} \tag{4.4.1.5}$$

$$f(U_{SS1}=1)=\lambda_t \cdot T_0 \cdot \frac{P_r(U_{SS1}=1 \mid H_0)}{P_r(U_{SS1}=1 \mid H_1)} \tag{4.4.1.6}$$

Then, the probability of error caused by the team process with SS1 only can be expressed in probabilistic terms as below. .

$$\begin{aligned}
&P_r(E1,\, y_{HS} \in [TL1, TL2] \text{ or } [TU2, TU1]) \\
&= P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} P_r(E1,\, y_{HS} \in [TL1, TL2] \text{ or } [TU2, TU1] \mid U_{SS1}, H_0) \cdot P_r(U_{SS1} \mid H_0) \\
&\quad + P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} P_r(E1,\, y_{HS} \in [TL1, TL2] \text{ or } [TU2, TU1] \mid U_{SS1}, H_1) \cdot P_r(U_{SS1} \mid H_1) \\
&= P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} P_r\{g(y_{HS}) > \\
&\qquad f(U_{SS1}) \text{ and } y_{HS} \in [TL1, TL2] \text{ or } [TU2, TU1] \mid U_{SS1}, H_0\} \cdot P_r(U_{SS1} \mid H_0) \\
&\quad + P_r(H_1) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} P_r\{g(y_{HS}) > \\
&\qquad f(U_{SS1}) \text{ and } y_{HS} \in [TL1, TL2] \text{ or } [TU2, TU1] \mid U_{SS1}, H_1\} \cdot P_r(U_{SS1} \mid H_1)
\end{aligned} \tag{4.4.1.7}$$

4.4.2. LRT for Communicating with SS1 and SS2

This section is very similar to the section 3.3 of Chapter 3. The LRT when communicating with two slave sensors had been derived in Chapter 3. Adapting (3.3.1) and (3.3.2), we obtain $\Lambda_{T2}(y_{HS}, U_{SS1}, U_{SS2}) = \Lambda(y_{HS}, U_{SS1}, U_{SS2})$. The (3.3.3) can be directly applicable in this section as well. By replacing $g(y_{HS})$ in (3.3.4) with $g_{T2}(y_{HS})$, we have the description of the two-helper LRT as

$$g_{T2}(y_{HS}) \underset{U_F=0}{\overset{U_F=1}{\gtrless}} f(U_{SS1},U_{SS2}) \tag{4.4.2.1}$$

From (4.4.1.4) and (4.4.2.1), the final decision, U_F, rule of the system can be written as

$$U_F = \begin{cases} 1 & , \text{ if } y_{HS} \geq TU1 \\ 1 & , \text{ if } g_{T2}(y_{HS}) \geq f(U_{SS1}, U_{SS2}) \\ 1 & , \text{ if } g_{T1}(y_{HS}) \geq f(U_{SS1}) \\ 0 & , \text{ if } g_{T1}(y_{HS}) < f(U_{SS1}) \\ 0 & , \text{ if } g_{T2}(y_{HS}) < f(U_{SS1}, U_{SS2}) \\ 0 & , \text{ if } y_{HS} < TL1 \end{cases}$$

An explicit expression of the final threshold, $f(U_{SS1},U_{SS2})$, is dependent upon the decision on the slave sensors as mentioned in Chapter 3. The explicit expressions are given in section 3.3, (3.3.5), (3.3.6), (3.3.7), and (3.3.8). Then, it is possible to express the probability of error caused by using data from two slave sensors. This is shown in (4.4.2.2).

$$\begin{aligned} &P_r(E2, y_{HS} \in [TL2, TU2]) \\ &= P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r(E2, y_{HS} \in [TL2, TU2] \mid U_{SS1}, U_{SS2}, H_0) \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(U_{SS1} \mid H_0) \\ &+ P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r(E2, y_{HS} \in [TL2, TU2] \mid U_{SS1}, U_{SS2}, H_1) \cdot P_r(U_{SS2} \mid H_1) \cdot P_r(U_{SS2} \mid H_1) \\ &= P_r(H_0) \cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r\{g(y_{HS}) > \\ &\qquad f(U_{SS1},U_{SS2}) \text{ and } y_{HS} \in [TL2, TU2] \mid U_{SS1}, U_{SS2}, H_0\} \cdot P_r(U_{SS1} \mid H_0) \cdot P_r(us2 \mid H_0) \end{aligned}$$

$$+ P_r(H_1)\cdot \sum_{U_{SS1}=0}^{U_{SS1}=1} \sum_{U_{SS2}=0}^{U_{SS2}=1} P_r\{g(y_{HS}) >$$

$$f(U_{SS1},U_{SS2}) \text{ and } y_{HS} \in [TL2, TU2] \mid U_{SS1}, U_{SS2}, H_1\} \cdot P_r(U_{SS1} \mid H_1)\cdot P_r(U_{SS2} \mid H_1) \quad (4.4.2.2)$$

4.5. Calculation of $\overline{C}$ under Gaussian Models

We again assume the probability density function on the observation of the host sensor and the slave sensors given in Section 3.4.1 of Chapter 3. The decision boundary for the local decision on the slave sensors is also given in Section 3.4.2 of Chapter 3, namely set the binary decision threshold at 0.

$$P_{e_{HS}} = P_r(H_0)\cdot \int_{TU1}^{\infty} f_{HS_0}(y_{HS})dy_{HS} + P_r(H_1)\cdot \int_{-\infty}^{TL1} f_{HS_1}(y_{HS})dy_{HS}$$

$$= P_r(H_0)\cdot Q\left[\frac{TU1-\mu_{HS_0}}{\sigma_{HS_0}}\right] + P_r(H_1)\cdot\left\{1-Q\left[\frac{TL1-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\} \quad (4.5.1)$$

$$P_r(z_{TI}=1) = P_r(H_0)\cdot\left\{\int_{TL1}^{TL2} f_{HS_0}(y_{HS})dy_{HS} + \int_{TU2}^{TU1} f_{HS_0}(y_{HS})dy_{HS}\right\}$$

$$+ P_r(H_1)\cdot\left\{\int_{TL1}^{TL2} f_{HS_1}(y_{HS})dy_{HS} + \int_{TU2}^{TU1} f_{HS_1}(y_{HS})dy_{HS}\right\}$$

$$= P_r(H_0)\cdot\left\{Q\left[\frac{TL1-\mu_{HS_0}}{\sigma_{HS_0}}\right] - Q\left[\frac{TL2-\mu_{HS_0}}{\sigma_{HS_0}}\right] + Q\left[\frac{TU2-\mu_{HS_0}}{\sigma_{HS_0}}\right] - Q\left[\frac{TU1-\mu_{HS_0}}{\sigma_{HS_0}}\right]\right\}$$

$$+ P_r(H_1)\cdot\left\{Q\left[\frac{TL1-\mu_{HS_1}}{\sigma_{HS_1}}\right] - Q\left[\frac{TL2-\mu_{HS_1}}{\sigma_{HS_1}}\right] + Q\left[\frac{TU2-\mu_{HS_1}}{\sigma_{HS_1}}\right] - Q\left[\frac{TU1-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\} \quad (4.5.2)$$

$$P_r(z_{T2}=1) = P_r(H_0)\cdot\int_{TL2}^{TU2} f_{HS_0}(y_{HS})dy_{HS} + P_r(H_1)\cdot\int_{TL2}^{TU2} f_{HS_0}(y_{HS})dy_{HS}$$

$$= P_r(H_0)\cdot\left\{Q\left[\frac{TL2-\mu_{HS_0}}{\sigma_{HS_0}}\right] - Q\left[\frac{TU2-\mu_{HS_0}}{\sigma_{HS_0}}\right]\right\}$$

$$+ P_r(H_1)\cdot\left\{Q\left[\frac{TL2-\mu_{HS_1}}{\sigma_{HS_1}}\right] - Q\left[\frac{TU2-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\} \qquad (4.5.3)$$

$$P_{e_{T1}} = P_r(H_0)\cdot\left\{\int_{f(U_{SS1}=0)}^{\infty} f_{HS_0}(y_{HS})\,dy_{HS}\cdot\int_{-\infty}^{T_{SS1}} f_{SS1_0}(y_{SS1})\,dy_{SS1}\right.$$

$$\left.+\int_{f(U_{SS1}=1)}^{\infty} f_{HS_0}(y_{HS})\,dy_{HS}\cdot\int_{T_{SS1}}^{\infty} f_{SS1_0}(y_{SS1})\,dy_{SS1}\right\}$$

$$+ P_r(H_1)\cdot\left\{\int_{-\infty}^{f(U_{SS1}=0)} f_{HS_1}(y_{HS})\,dy_{HS}\cdot\int_{-\infty}^{T_{SS1}} f_{SS1_1}(y_{SS1})\,dy_{SS1}\right.$$

$$\left.+\int_{-\infty}^{f(U_{SS1}=1)} f_{HS_1}(y_{HS})\,dy_{HS}\cdot\int_{T_{SS1}}^{\infty} f_{SS1_1}(y_{SS1})\,dy_{SS1}\right\}$$

$$= P_r(H_0)\cdot\left[Q\left(\frac{f(U_{SS1}=0)-\mu_{HS_0}}{\sigma_{HS_0}}\right)\cdot\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\right\}\right.$$

$$\left.+Q\left(\frac{f(U_{SS1}=1)-\mu_{HS_0}}{\sigma_{HS_0}}\right)\cdot Q\left(\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right)\right]$$

$$+\,P_r(H_1)\cdot\left[\left\{1-Q\left[\frac{f(U_{SS1}{=}0)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}\cdot\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\right\}\right.$$

$$\left.+\left\{1-Q\left[\frac{f(U_{SS1}{=}1)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}\cdot Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\right] \tag{4.5.4}$$

$$P_{e_{T2}} = P_r(H_0)\cdot\left[\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\right\}\cdot\left\{1-Q\left[\frac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\right\}\cdot Q\left[\frac{f(U_{SS1}{=}0,\,U_{SS2}{=}0)-\mu_{HS_0}}{\sigma_{HS_0}}\right]\right.$$

$$+\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\right\}\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\cdot Q\left[\frac{f(U_{SS1}{=}0,\,U_{SS2}{=}1)-\mu_{HS_0}}{\sigma_{HS_0}}\right]$$

$$+\,Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\cdot\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\right\}\cdot Q\left[\frac{f(U_{SS1}{=}1,\,U_{SS2}{=}0)-\mu_{HS_0}}{\sigma_{HS_0}}\right]$$

$$\left.+\,Q\left[\frac{T_{SS1}-\mu_{SS1_0}}{\sigma_{SS1_0}}\right]\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_0}}{\sigma_{SS2_0}}\right]\cdot Q\left[\frac{f(U_{SS1}{=}1,\,U_{SS2}{=}1)-\mu_{HS_0}}{\sigma_{HS_0}}\right]\right]$$

$$+\,P_r(H_1)\cdot\left[\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\right\}\cdot\left\{1-Q\left[\frac{T_{SS2}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\right\}\cdot\left\{1-Q\left[\frac{f(U_{SS1}{=}0,\,U_{SS2}{=}0)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}\right.$$

$$+\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\right\}\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\cdot\left\{1-Q\left[\frac{f(U_{SS1}{=}0,\,U_{SS2}{=}1)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}$$

$$+\,Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\cdot\left\{1-Q\left[\frac{T_{SS1}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\right\}\cdot\left\{1-Q\left[\frac{f(U_{SS1}{=}1,\,U_{SS2}{=}0)-\mu_{HS_0}}{\sigma_{HS_1}}\right]\right\}$$

$$+Q\left[\frac{T_{SS1}-\mu_{SS1_1}}{\sigma_{SS1_1}}\right]\cdot Q\left[\frac{T_{SS2}-\mu_{SS2_1}}{\sigma_{SS2_1}}\right]\cdot\left\{1-Q\left[\frac{f(U_{SS1}=1,\,U_{SS2}=1)-\mu_{HS_1}}{\sigma_{HS_1}}\right]\right\}\right]\right] \qquad (4.5.5)$$

4.5.1. Numerical Evaluation of $\bar{C}$

The same method is used as in the previous chapters in evaluating $\bar{C}$ numerically. The difference is that the host sensor in this system has 4 thresholds, unlike 2SS and 3SS. The thresholds are varied with a relationship of TU31 = -TL31, TU32 = -TL32, and TU32 = $\frac{1}{2}$TU31. This threshold relationship is selected arbitrarily. For the threshold configuration, refer back to Figure 4.1. The program written for this evaluation is attached under Appendix C. As shown in Figure 4.3, depending on the communication cost constants, CCC1 and CCC2, individual curves are obtained. The dotted curve which is generated using CCC1 = 0.325 and CCC2 = 0.65 is the last curve with a minimum other than at the threshold position of 0.0. Figure 4.4 is an enlarged version of Figure 4.3. In Figure 4.3 and 4.4, the curves have ripples, unlike the set of curves shown in the previous chapters. This phenomenon is induced from the arbitrary choice of thresholds, giving a suboptimal threshold locations, and from the changes of the system's communication scheme from one to another. Figure 4.5 shows the minimum expected system cost holds at a constant beyond the communication cost constant of 0.65. This is because that the optimum thresholds at the host sensor, TU31, TU32, TL31, and TL32, eventually become zero. This is more clearly represented in Figure 4.6.

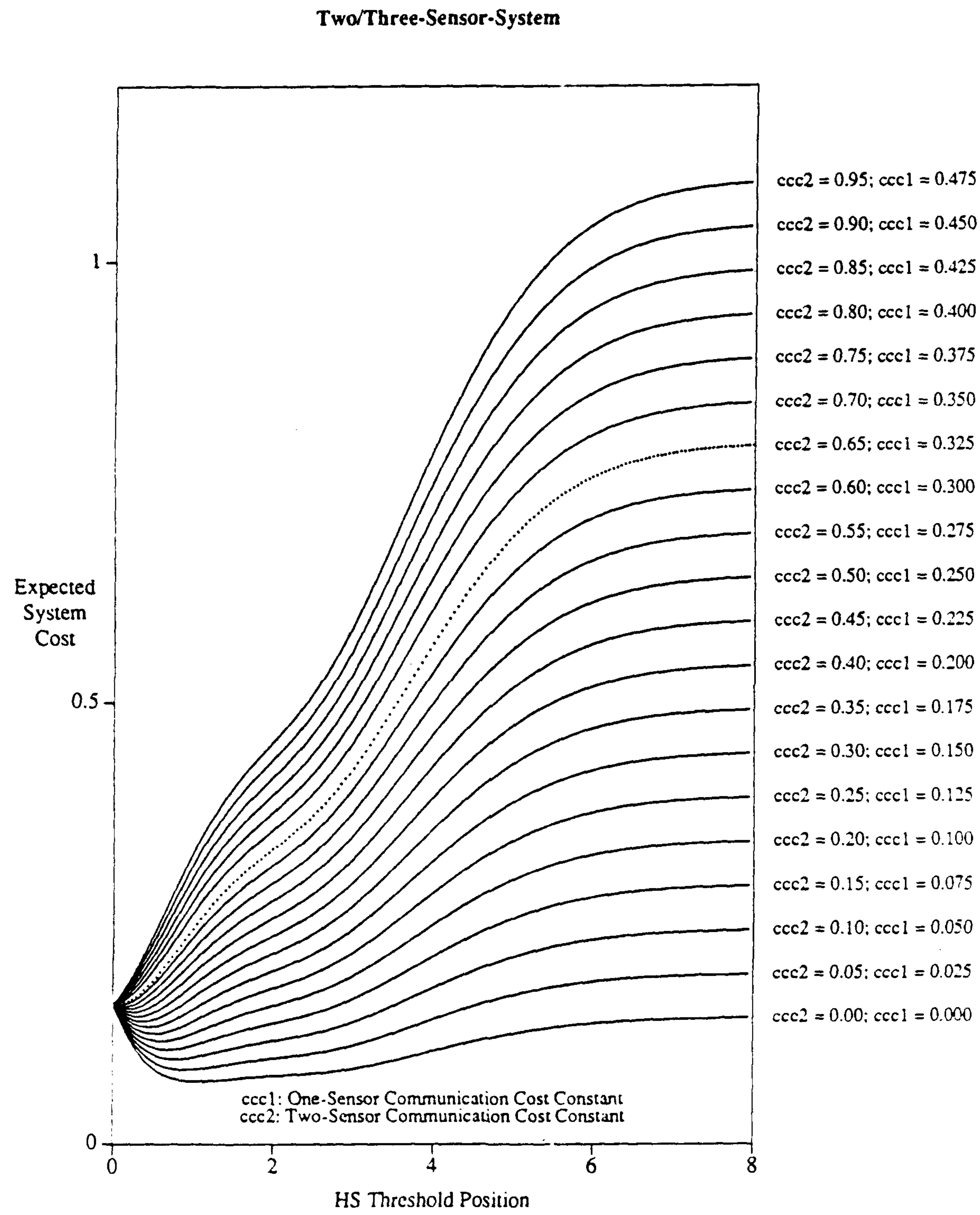

Figure 4.3 Expected System Cost vs. HS Threshold Position

Two/Three-Sensor-System

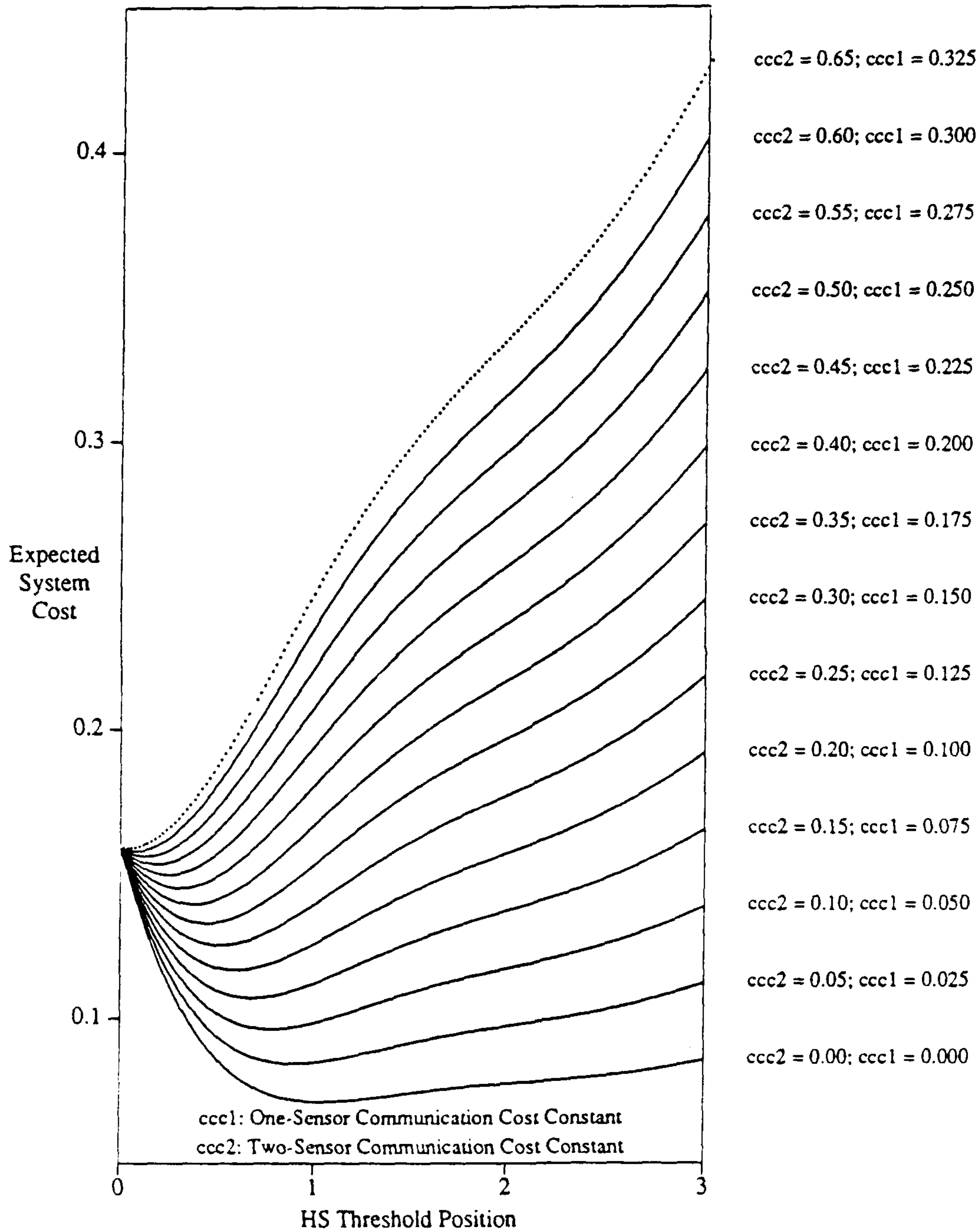

Figure 4.4 Expected System Cost vs. HS Threshold Position
Enlarged Version of Figure 4.3

Two/Three-Sensor-System

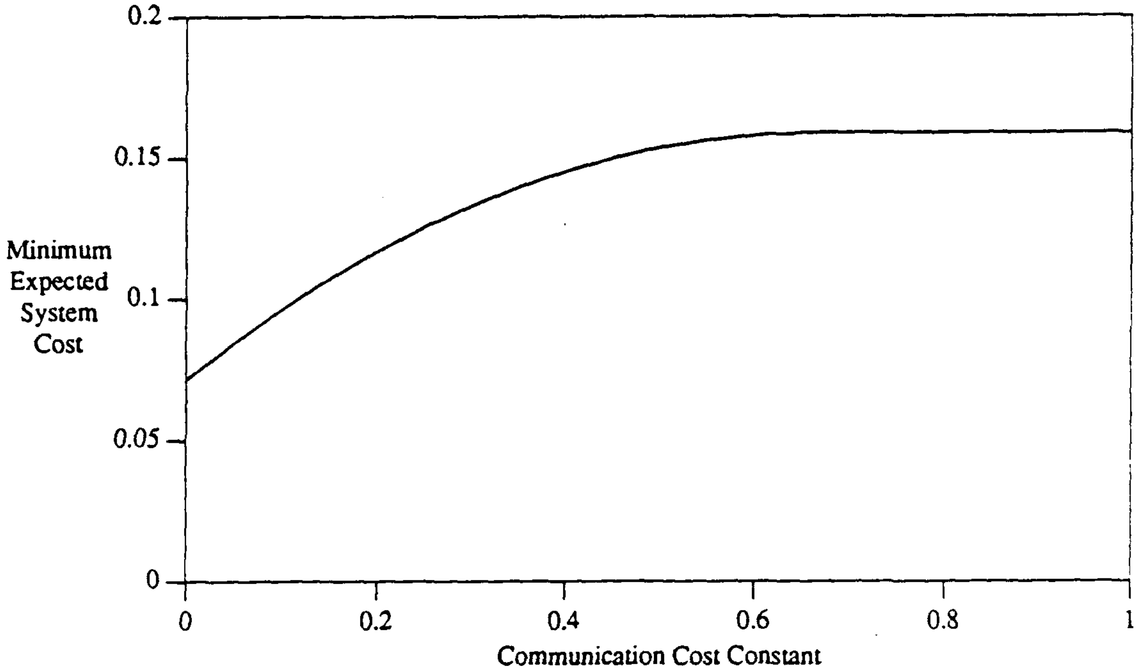

Figure 4.5 Min. Expected System Cost vs. Communication Cost Constant

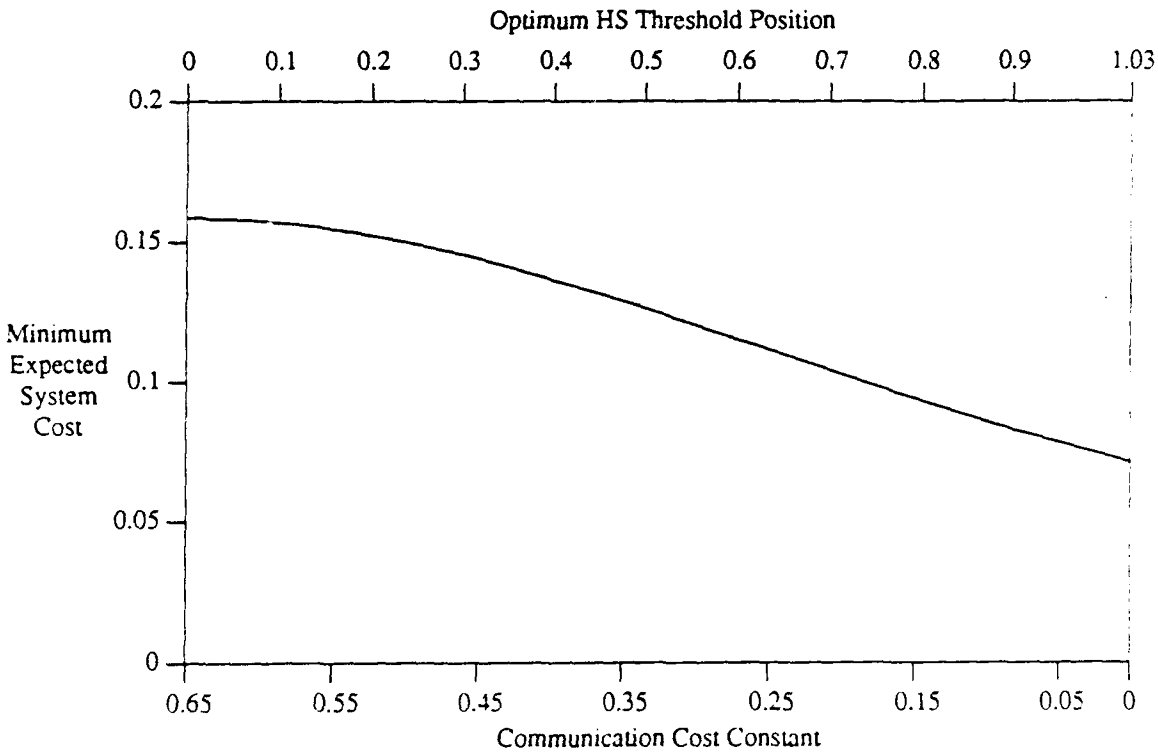

Figure 4.6 Summary of Data

CHAPTER 5

Comparison of $\bar{C}$ of 2SS, 3SS, and 2/3SS

5.1. Comparison of $\bar{C}$

In this section numerically-evaluated expected system costs in Chapter 2, Chapter 3, and Chapter 4 are compared against each other. The comparison are made based upon the data obtained using Gaussian models for the different sensor systems.

The system expected costs are evaluated over the various threshold locations on the host sensor's observation space and different communication cost constant incurred in communication between the host sensor and the slave sensors. Data are collected from the results obtained through the $\bar{C}$ expressed in terms of Q(y)-functions. These informations are plotted and attached at the end of Chapter 2, Chapter 3, and Chapter 4. The summarized data are tabulated in Table 5.1, Table 5.2, and Table 5.3 in following sections.

5.1.1. $\bar{C}$ of 2SS

Figure 2.3 shows that the total expected system costs are evaluated as the thresholds, TL and TU, are departing from the origin with various communication cost constant, CCC1. As in Figure 2.3 or 2.4, some of the curves have minima other than at the threshold position of 0.0, some don't. It is roughly seen that a minimum of curve occurs at the threshold position of 0.0 when CCC1 $\geq$ 0.5. If CCC1 $\geq$ 0.5, the communications between sensors are prohibited and the final decision is made by the host sensor alone. The exact value of the communication cost constant that may not give minimum (other than the threshold position of 0.0) is included between 0.45 and 0.50 leaning more toward to 0.45. The dotted curve indicates that the communication cost constant is 0.45. As the thresholds move away from 0.0, the cost is increasing beyond the optimal threshold position. It begins to stop increasing near the threshold position of 4.0.

Figure 2.5 is a plot of extracted information from Figure 2.3. It shows the behavior of the minimum expected system cost due to the change of the communication cost constant. As the communication cost constant becomes greater, the minimum expected system cost increases; however, the cost starts saturating at CCC of 0.45. The percentage change in the expected system cost when the communication cost constants are varied from 1.0 to 0.0 is 43.7 %. This shows that the communication cost constant takes a very important role in the system.

The relationship among the minimum expected cost, the optimal threshold position, and the communication cost constant is shown in Figure 2.6. The numerical tabulated data are given in Table 5.1. In Table 5.1, when the minimum expected system cost is 0.1587, this means there are no communications between sensors. This number, thus, represents the cost of making a final decision by the host sensor only.

5.1.2. $\bar{C}$ of 3SS

As in the previous section, Figure 3.3 shows that the expected system cost vs. the threshold position in the host sensor's observation space. The dotted curve indi-

Communication Cost Constant CCC	Optimum Threshold Position TU	Minimum Expected System Cost
0.00	0.700	0.0891
0.05	0.585	0.1045
0.10	0.490	0.1175
0.15	0.405	0.1283
0.20	0.330	0.1372
0.25	0.265	0.1444
0.30	0.200	0.1500
0.35	0.140	0.1542
0.40	0.085	0.1570
0.45	0.035	0.1584
0.50	0.000	0.1587
0.55	0.000	0.1587
0.60	0.000	0.1587
0.65	0.000	0.1587
0.70	0.000	0.1587
0.75	0.000	0.1587
0.80	0.000	0.1587
0.85	0.000	0.1587
0.90	0.000	0.1587
0.95	0.000	0.1587
1.00	0.000	0.1587

Table 5.1 Tabulated Data of 2SS

cates that the curves with CCC2 $\geq$ 0.55 have minima at the threshold position of 0.0.

The information in Figure 3.3 are summarized in Figure 3.5 and Figure 3.6. The numerical tabulated data of these figures are listed in Table 5.2. From the table there is 51.5 % difference in the expected system cost when communication cost is varied from 1.0 to 0.0.

Communication Cost Constant CCC	Optimum Threshold Position TU	Minimum Expected System Cost
0.00	0.785	0.0773
0.05	0.655	0.0947
0.10	0.550	0.1092
0.15	0.460	0.1214
0.20	0.380	0.1315
0.25	0.310	0.1400
0.30	0.240	0.1465
0.35	0.180	0.1516
0.40	0.125	0.1553
0.45	0.070	0.1576
0.50	0.015	0.1586
0.55	0.000	0.1587
0.60	0.000	0.1587
0.65	0.000	0.1587
0.70	0.000	0.1587
0.75	0.000	0.1587
0.80	0.000	0.1587
0.85	0.000	0.1587
0.90	0.000	0.1587
0.95	0.000	0.1587
1.00	0.000	0.1587

Table 5.2 Tabulated Data of 3SS

5.1.3. $\bar{C}$ of 2/3SS

The numerical tabulated data of Figure 4.3, Figure 4.4, and Figure 4.5 is in Table 5.3. In Figure 4.3, it is noted that the dotted curve occurs when CCC1 = 0.325 and CCC2 = 0.65. The percentage change in the expected system cost when CCC1 changes from 0.5 to 0.0, meaning CCC2 changes from 1.0 to 0.0, is 55.3 %. It is clearly shown in Figure 4.3 that the curves are leveling off near the threshold position of 8.0.

Communication Cost Constant with Two Sensor CCC2	Communication Cost Constant with One Sensors CCC1	Optimum Inner Threshold Position TU32	Optimum Outer Threshold Position TU31	Minimum Expected System Cost
0.00	0.000	0.5150	1.030	0.0712
0.05	0.025	0.4425	0.885	0.0845
0.10	0.050	0.3850	0.770	0.0964
0.15	0.075	0.3375	0.675	0.1072
0.20	0.010	0.2925	0.585	0.1169
0.25	0.125	0.2550	0.510	0.1255
0.30	0.150	0.2175	0.435	0.1330
0.35	0.175	0.1825	0.365	0.1396
0.40	0.200	0.1525	0.305	0.1452
0.45	0.225	0.1200	0.240	0.1498
0.50	0.250	0.0900	0.180	0.1534
0.55	0.275	0.0625	0.125	0.1561
0.60	0.300	0.0350	0.070	0.1578
0.65	0.325	0.0075	0.015	0.1586
0.70	0.350	0.0000	0.000	0.1587
0.75	0.375	0.0000	0.000	0.1587
0.80	0.400	0.0000	0.000	0.1587
0.85	0.425	0.0000	0.000	0.1587
0.90	0.450	0.0000	0.000	0.1587
0.95	0.475	0.0000	0.000	0.1587
1.00	0.500	0.0000	0.000	0.1587

Table 5.3 Tabulated Data of 2/3SS

5.2. Comparison of Systems

Since each system's numerical evaluation results are collected, and 5.1.3, it is possible to carry out the performance comparison of these systems. Mainly the systems' expected cost and the optimal threshold position at different communication cost constant are considered for the comparison. The method used to compare the systems in this section is that, first, the 2SS is compared with the rest of systems, 3SS and 2/3SS. Secondly, the 2SS is compared to 2/3SS. For the convenience, the communication cost constants of 0.0 and 0.45 are chosen to be the bases of comparison. The communication cost constant of 0.0 is selected since it means that there is no risk in communication between sensors, in other words, the communication between the host sensor and the slave sensors is encouraged. The communication cost constant of 0.45 are chosen because it is the largest communication cost constant of 2SS which gives an optimal threshold position other than 0.0.

Using the table presented in the previous sections, at the communication cost constant of 0.0, the expected system cost of 3SS is 13.24 % less than that of 2SS. Comparing 2SS to 2/3SS, 2/3SS outperforms 2SS by 20.10 % in the expected system cost. In comparing with 2/3SS, the outer threshold location is selected for the comparison. 2/3SS has 47.14 % larger width (or size) of the dubious decision region in systems observation space. This paragraph is summarized in Table 5.4.

	Type of Sensor System		
CCC1 = 0.0 CCC2 = 0.0	2SS	3SS	2/3SS
Improvement in Expected System Cost	0.0 %	13.24 %	20.10 %
Improvement in Optimal Threshold Position	0.0 %	12.14 %	47.14 %

CCC1 = Communication Cost Constant of communicating with one sensor
CCC2 = Communication Cost Constant of communicating with two sensors

Table 5.4 Comparison of 2SS to the Others with CCC=0.0

In aspects of the optimal threshold position, 2/3SS has a wider uncertain decision region than 3SS by 31.21 %. With CCC1 = CCC2 = 0.0, 2/3SS performs about 7.90 % better than 3SS in the expected system cost. This is because 2/3SS requests information from the slave sensors more frequent than 3SS since 2/3SS has a wider uncertain decision region. This information are contained in Table 5.5.

Now we consider system improvements in the expected system cost and in optimal threshold location with the communication cost constant of 0.45 is considered. In 2/3SS this communication cost constant is used when the host sensor communicates with two slave sensors; when the host sensor communicates with only one sensor, the communication cost constant in this case is a half of the prior case, 0.225. In aspects of the expected system cost, the difference of system cost between 2SS and 3SS is 0.5 % in favour of 3SS. For the optimal threshold location, 3SS has a wider uncertainty region by 100 %. In comparison of the 2SS to 2/3SS, 2/3SS performs better in the expected system cost by 5.43 %. These are listed in Table 5.6.

	Type of Sensor System	
CCC1 = 0.0 CCC2 = 0.0	3SS	2/3SS
Improvement in Expected System Cost	0.0 %	7.90 %
Improvement in Optimal Threshold Position	0.0 %	31.21 %

CCC1 = Communication Cost Constant of communicating with one sensor
CCC2 = Communication Cost Constant of communicating with two sensors

Table 5.5 Comparison of 2SS to 2/3SS with CCC=0.0

	Type of Sensor System		
CCC1 = 2.25 CCC2 = 4.5	2SS	3SS	2/3SS
Improvement in Expected System Cost	0.0 %	0.5 %	5.43 %
Improvement in Optimal Threshold Position	0.0 %	100 %	584.71 %

CCC1 = Communication Cost Constant of communicating with one sensor
CCC2 = Communication Cost Constant of communicating with two sensors

Table 5.6 Comparison of 2SS to the Others with CCC $\neq$ 0.0

2/3SS performs about 4.95 % better than 3SS in the expected system cost. In aspects of the optimal threshold position, 2/3SS has a wider uncertain decision region than 3SS by 242.86 %. This information is contained in Table 5.7.

It is noted that the width of the threshold location is shrinking relatively faster for 2SS and 3SS than 2/3SS as the communication cost constant increases. As far as

	Type of Sensor System	
CCC1 = 2.25 CCC2 = 4.5	3SS	2/3SS
Improvement in Expected System Cost	0.0 %	4.95 %
Improvement in Optimal Threshold Position	0.0 %	242.86 %

CCC1 = Communication Cost Constant of communicating with one sensor
CCC2 = Communication Cost Constant of communicating with two sensors

Table 5.7 Comparison of 2SS to 2/3SS with CCC $\neq$ 0.0

the expected system cost is concerned, there is not a great difference as in the position of optimal threshold. Moreover, at a higher communication cost constant say 0.45 (refer to Table 5.6), there are insignificant differences in the expected system cost among the systems.

It is interesting to observe the relationship between optimal thresholds and $P_r(U_{HS} = ?)$ since the probability of an observation landing in the dubious region is closely related to the optimal thresholds location in the host sensor. $P_r(U_{HS} = ?)$ represents that the probability of the host sensor's observation falls in the uncertainty region, $TL \leq y_{HS} \leq TU$, inducing the host sensor's local decision to be "?". This relationship is tabulated in Table 5.8, Table 5.9, and Table 5.10. It is obvious, without looking at the tables, that $P_r(U_{HS} = ?)$ decreases as the optimum threshold approaches to zero. When the tables are plotted (See Figure 5.1, 5.2, and 5.3), a linear relationship is found between the optimal threshold positions and the probabil-

Optimal Threshold Position	$P_r(U_{HS} = ?)$
0.700	0.3375
0.585	0.2826
0.490	0.2369
0.405	0.1959
0.330	0.1597
0.265	0.1282
0.200	0.0968
0.140	0.0678
0.085	0.0411
0.035	0.0169
0.000	0.0000

Table 5.8 $P_r(U_{HS} = ?)$ given the Optimal Thresholds of 2SS

ity of U_{HS} = ?. The program which evaluates the probability of communication with given the optimal thresholds is attached in Appendix D.

Optimal Threshold Position	$P_r(U_{HS} = ?)$
0.785	0.3778
0.655	0.3161
0.550	0.2658
0.460	0.2225
0.380	0.1838
0.310	0.1500
0.240	0.1161
0.180	0.0871
0.125	0.0605
0.070	0.0339
0.015	0.0073
0.000	0.0000

Table 5.9 $P_r(U_{HS} = ?)$ given the Optimal Thresholds of 3SS

Optimal Threshold Position	$P_r(U_{HS} = ?)$
1.030	0.4908
0.885	0.4245
0.770	0.3707
0.675	0.3256
0.585	0.2826
0.510	0.2465
0.435	0.2104
0.365	0.1766
0.305	0.1476
0.240	0.1161
0.180	0.0871
0.125	0.0605
0.070	0.0339
0.015	0.0073
0.000	0.0000

Table 5.10 $P_r(U_{HS} = ?)$ given the Optimal Thresholds of 2/3SS

Two-Sensor-System

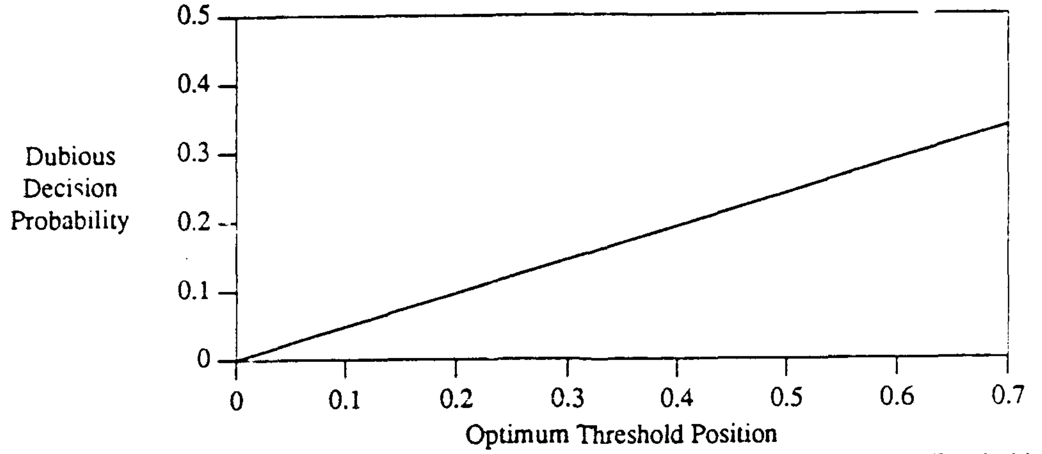

Figure 5.1 Dubious Decision Probability at HS vs. HS Optimum Threshold

Three-Sensor-System

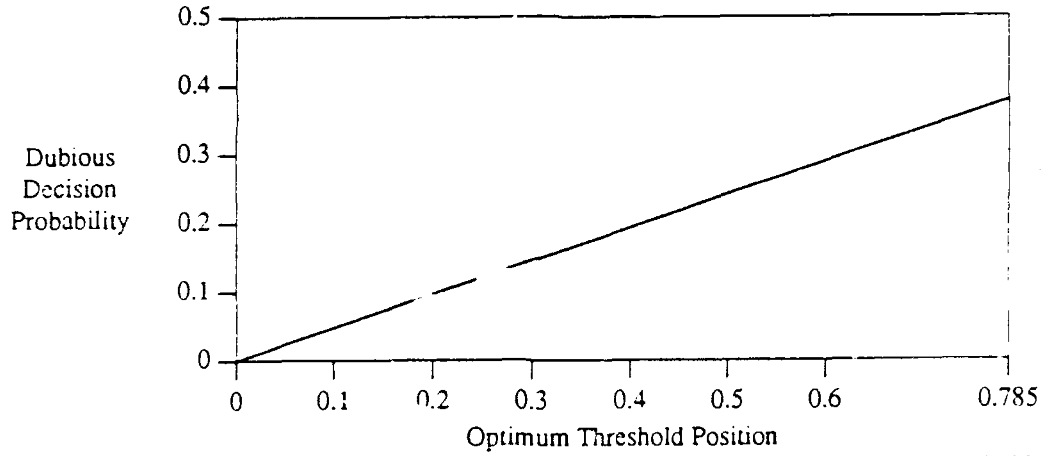

Figure 5.2 Dubious Decision Probability at HS vs. HS Optimum Threshold

Two/Three-Sensor-System

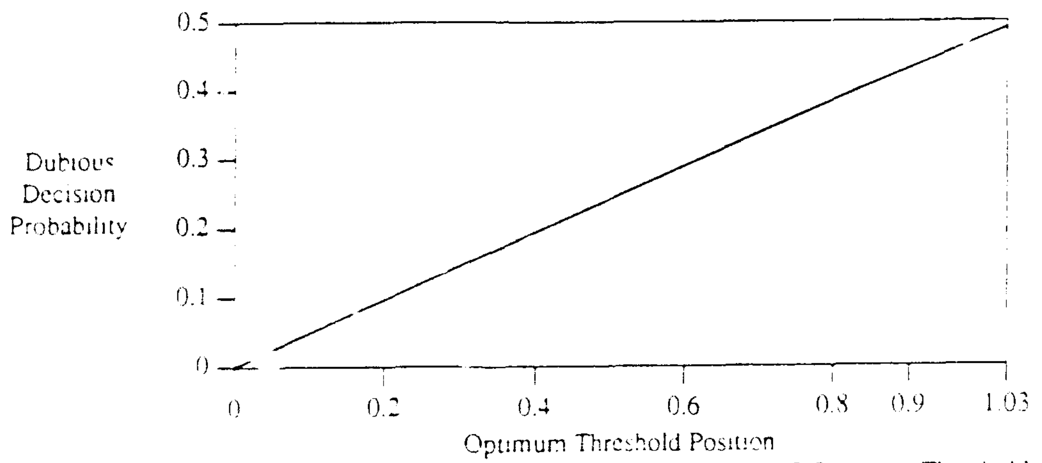

Figure 5.3 Dubious Decision Probability at HS vs. HS Optimum Threshold

CHAPTER 6

System Simulations

6.1. Simulation Method

Simulation of the systems evaluated in Chapter 2, 3, and 4 are performed to understand how these systems behave in a realistic environment. The same assumptions as those made in the beginning of this work (Refer Chapter 1) are also used in the simulation. One additional system is simulated in addition to three systems with which we have dealt. This system consists of one sensor that has a single threshold and no slave sensors.

To the signal, either -1 or 1, Gaussian noise is added at the host sensor and the slave sensors. In each system, different slave sensors receive independent observation. However, in all systems, each host sensor receives the same observation so that the performance of each system can be compared easily. In the simulations, different communication constants were used and the number of iterations performed was 10,000. The iterations can be interpreted as the number of observations taken by the host sensor and the slave sensors. In our Gaussian random number generation routine, 10,000 iterations provide with well distributed Gaussian random numbers. The outputs of the different system are compared in terms of the percentage of correct detections (CD), false alarms (FA), and target misses (TM). The total detection error is, then, FA + TM. These are listed in Table 6.1 and Table 6.2.

Simulation of Systems						
	Type of Sensor Systems					
	1SS			2SS		
CCC	CD (%)	FA (%)	TM (%)	CD (%)	FA (%)	TM (%)
0.00	84.17	7.64	8.18	90.38	4.60	5.01
0.05	83.71	8.09	8.82	89.02	5.45	5.52
0.10	83.90	7.96	8.13	88.76	5.51	5.72
0.15	83.94	8.00	8.05	88.85	5.60	5.54
0.20	83.89	8.03	8.07	88.26	5.83	5.90
0.25	83.32	8.51	8.16	87.24	6.54	6.21
0.30	83.52	8.31	8.16	86.31	6.88	6.80
0.35	84.73	7.66	7.60	86.73	6.72	6.45
0.40	84.02	8.11	7.86	84.95	7.69	7.35
0.45	84.03	8.01	7.95	84.70	7.63	7.66
0.50	84.08	7.98	7.93	84.08	7.98	7.93
0.55	84.45	7.94	7.60	84.45	7.94	7.60
0.60	83.88	8.17	7.94	83.88	8.17	7.94
0.65	84.16	7.97	7.86	84.16	7.97	7.86
0.70	84.81	7.75	7.43	84.81	7.75	7.43

CCC = communication cost constant

Table 6.1 Simulation Results for 1SS & 2SS

Simulation of Systems						
	Type of Sensor Systems					
	3SS			2/3SS		
CCC	CD (%)	FA (%)	TM (%)	CD (%)	FA (%)	TM (%)
0.00	90.36	2.07	7.56	89.93	3.03	7.03
0.05	89.44	3.02	7.53	90.16	3.46	6.37
0.10	88.95	3.34	7.70	89.72	3.53	6.74
0.15	89.38	3.80	6.81	90.13	3.87	5.99
0.20	88.76	4.28	6.95	89.91	4.15	5.93
0.25	87.80	5.23	6.96	89.32	4.35	6.32
0.30	86.88	5.73	7.38	88.57	5.12	6.30
0.35	87.19	5.87	6.93	89.16	4.78	6.05
0.40	85.76	6.86	7.37	87.79	5.62	6.58
0.45	84.96	7.28	7.75	87.62	5.73	6.64
0.50	84.43	7.74	7.82	86.60	6.50	6.89
0.55	84.45	7.94	7.60	86.29	6.77	6.93
0.60	83.88	8.16	7.94	85.34	7.32	7.33
0.65	84.16	7.97	7.86	84.48	7.76	7.75
0.70	84.81	7.75	7.43	84.81	7.75	7.43

CCC = communication cost constant

Table 6.2 Simulation Results for 3SS & 2/3SS

6.2. Simulation Results and Discussion

The simulation results are quite reasonable. In general, the results show that 2/3SS performs the best and followed by 3SS, 2SS, and 1SS in declining performance. As shown in the previous chapter the optimal threshold of 2SS collapses to 0.0 when the communication cost constant is 0.5. This is also shown in Table 6.1 as the CD of 2SS equals to that of 1SS when CCC becomes 0.50. Also, CD of 3SS and 2/3SS become that of 1SS when CCC is equal to 0.55 and 0.70, respectively. This is because the host sensors in the different systems receive the same observations.

There are about 6 % difference in CD between 1SS and other systems; however, the difference among the 2SS, 3SS, and 2/3SS is rather insignificant when CCC is 0.0. This is because there are no influence of communication cost constant to each system. As the CCC increases, the differences in CD among systems become noticeable even though the largest difference are about 3.5 %. For 1SS CD, FA, and TM are virtually remain constant over all the CCCs were used since the performance of 1SS is independent from CCC. CD in 2SS decreases as CCC increases. FA and MT are increasing as CCC increases. In 3SS and 2/3SS FA is about 3.5 and 2.3 times less than TM, respectively, when CCC = 0.0. As CCC increases the ratio of FA and MT approaches to 1.0. These informations are tabulated in Table 6.1 and Table 6.2. The plotted version of these data are in Figure 6.1, Figure 6.2, and Figure 6.3. The program for this simulation is attached in Appendix E.

Correct Detections (%)

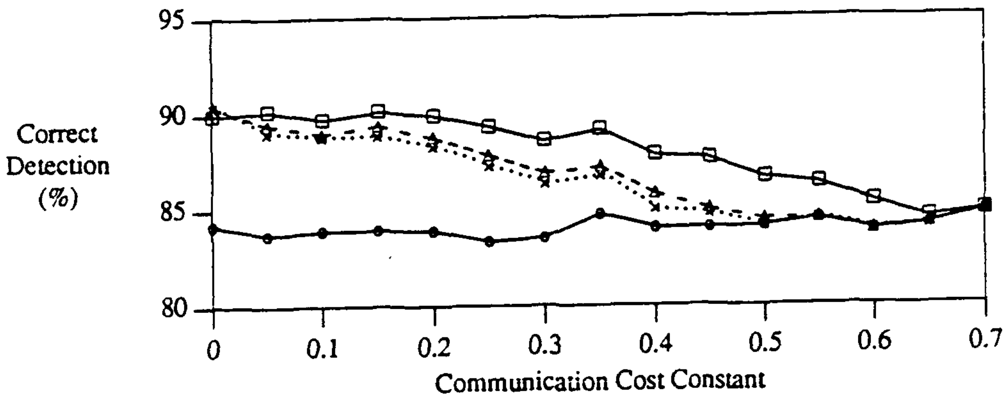

Figure 6.1 CD vs. CCC

False Alarms (%)

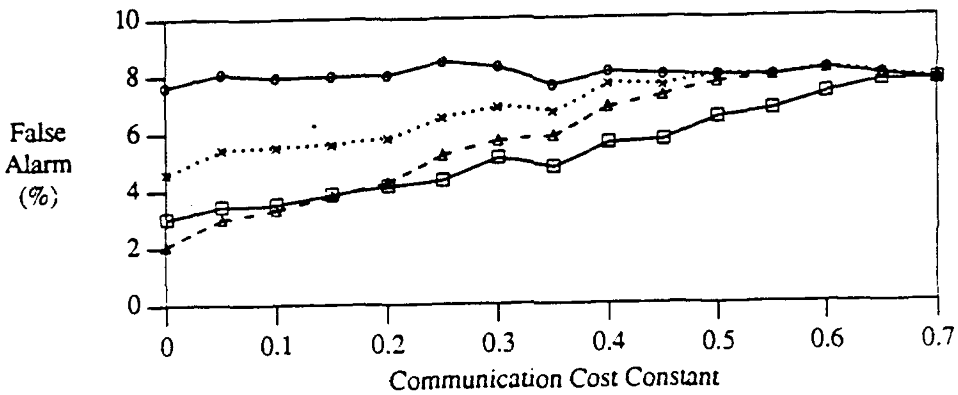

Figure 6.2 FA vs. CCC

Target Misses (%)

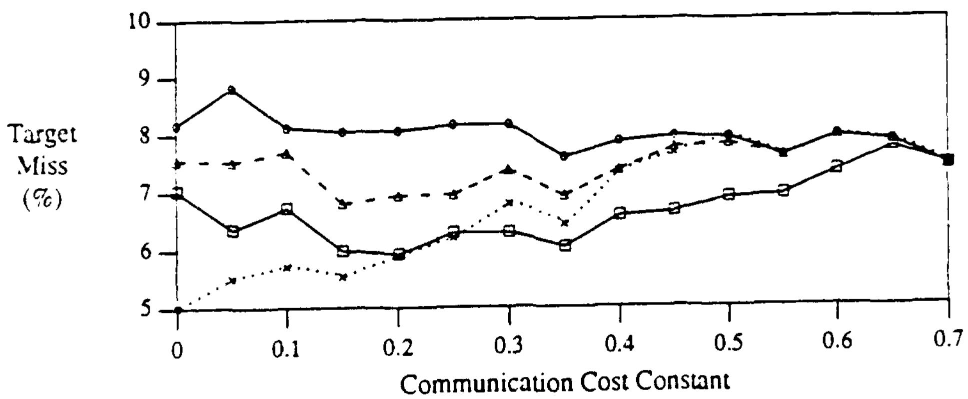

Figure 6.3 TM vs. CCC

O----> 1SS; × ----> 2SS; Δ ----> 3SS; □ ----> 2/3SS

CHAPTER 7

Conclusion

One objective of this study was to characterize team strategy decision methods in terms of analytical derivation, numerical evaluations, and system simulations. The optimization of the system in terms of minimization of either the expected system cost or the probability of error in decision is another objective.

The team strategy is applied to three different systems, and the performance of each system is characterized. Cost functions for each system are defined. From the cost function, the expected system cost, $\bar{C}$, is derived. The $\bar{C}$ is represented in general probabilistic terms as well as for Gaussian statistics using Q(y) functions. The numerical evaluations are performed for Gaussian models. The numerical evaluation shows, subject to communication cost, that 2/3SS is the most efficient. The next most desirable system is 3SS and the least is 2SS. Simulation of the three systems was also carried out. The simulation results confirm the above order of desirability.

The communication cost constant plays an important role in the global decisions of team strategies. Changes in the communication cost constant influence the frequency of communication between sensors. Selection of optimum thresholds in the host sensor is heavily dependent upon the communication cost constant since the frequency of the communication allowance determines the optimal threshold positions.

The simulation results show that there is a communication cost constant which makes all the systems perform the same. The communication cost constant in this situation is 0.7. Thus when the communication between sensors becomes very risky or expensive, meaning a high probability of interception, the sensors avoid communication. This fact is shown by comparing the system to 1SS because 1SS has no communication capability, i.e, there are no slave sensors involved. Refer to Figure 6.1, 6.2, and 6.3.

As the communication cost increases, the system with three sensors apparently make better chances of detection than the system with two sensors. When the communication cost is large so that communications between sensors are prohibited, then the performance of 2SS, 3SS, and 2/3SS is compatible to that of 1SS.

APPENDIX A

Program for Cost Evaluation of 2-Sensor-System

This appendix contains a FORTRAN program listing which evaluates (2.3.4) substituted with (2.5.4.1), (2.5.4.2), and (2.5.4.3) numerically. It consists of a main routine called "TWOSENSYS" and two subroutines, "QfunE" and "FindMin". Program TWOSENSYS (TWO SENsor SYStem) is responsible for iterations (generation of Gaussian observations), main calculations, calling subroutines, writing outputs to files, etc. Subroutine QfunE evaluates Q(y)-function (refer to (2.5.1.1)) when limits of integration are provided. Subroutine FindMin sorts a minimum in output data. This routine is used to find an optimal threshold in HS where the expected system cost is minimum.

The program computes expected system costs over threshold positions in HS for a given communication cost constant. Descriptions of variables used and comments are embedded in the program.

Figure 2.3, Figure 2.4, Figure 2.5, and Figure 2.6 are plotted version of outputs from this program. The tabulated data are contained in Table 5.1.

```
      PROGRAM TWOSENSYS

c***********************************************************************

c Author:      Howard C. Choe
c Organization: Department of Electrical Engineering
c              The University of Virginia, Charlottesville

c This program numerically evaluates expected costs of a
c two-sensor-system which uses team strategies for Gaussian statistics.

c Variable Description
c   p0    : a priori probability of "No target exists", H0
c   p1    : a priori probability of "Target exists", H1
c For Host Sensor
c   mh0   : mean value of H0 received by HS
c   mh1   : mean value of H1 received by HS
c   sh0   : standard deviation of H0 received by HS
c   sh1   : standard deviation of H1 received by HS
c For Slave Sensor
c   ms0   : mean value of H0 received by SS
c   ms1   : mean value of H1 received by SS
c   ss0   : standard deviation of H0 received by SS
c   ss1   : standard deviation of H1 received by SS
c Thresholds
c   For Host Sensor
c     TL  : lower threshold
c     TU  : upper threshold
c   For Slave Sensor
c     Tss : LRT optimal threshold
c Pre-cost constant
c   c00   : deciding H0 given H0
c   c10   : deciding H1 given H0
c   c11   : deciding H1 given H1
c   c01   : deciding H0 given H1
c Team effort cost
c   ccc   : communication cost constant

c***********************************************************************

c Declaration

      REAL p0, p1
      REAL mh0, mh1, ms0, ms1
      REAL sh0, sh1, ss0, ss1
```

```
      REAL c00, c10, c11, c01
      REAL Tss

      REAL ccc(21), cfmin(21), opthr(21)
      REAL TU(1001), TL(1001)
      REAL pehs(1001), pzcom(1001), cf(1001)
      REAL cbar(21,1001)

      DATA ccc/0.00, 0.05, 0.10, 0.15, 0.20,
     *         0.25, 0.30, 0.35, 0.40, 0.45,
     *         0.50, 0.55, 0.60, 0.65, 0.70,
     *         0.75, 0.80, 0.85, 0.90, 0.95, 1.00/

c The input data
c  a priori probability of the binary hypothesis environment
      p0 = 0.5
      p1 = 0.5

c  The statistics of the received information
      mh0 = -1.0
      mh1 =  1.0
      ms0 = -1.0
      ms1 =  1.0

      sigma = 1.0

c  Assign all standard deviation to the same value
      sh0 = sigma
      sh1 = sigma
      ss0 = sigma
      ss1 = sigma

c  pre-cost values
      c00 = 0.0
      c10 = 1.0
      c11 = 0.0
      c01 = 1.0

c Evaluation of the pre-calculated threshold for the team strategy and
c for the slave sensor

      plamss = (c10 - c00)/(c01 - c11)
      plamt  = plamss

c Evaluation of the ratio between a priori probabilities and LRT threshold
```

```
c  for the host sensor and the slave sensor, considering each sensor is
c  centralized individually.

      t0  = p0/p1
      Ths = (mh0 + mh1)/2.0 + (sigma**2/(mh1 - mh0))*LOG(plamt*t0)
      Tss = (ms0 + ms1)/2.0 + (sigma**2/(ms1 - ms0))*LOG(plamss*t0)

c  Evaluation of the final threshold after communication

      a0s = (Tss - ms0)/ss0
      a1s = (Tss - ms1)/ss1

      CALL QfunE(a0s, Qa0s)
      CALL QfunE(a1s, Qa1s)

      fus0 = plamss * t0 * (1.0 - Qa0s)/(1.0 - Qa1s)
      fus1 = plamss * t0 * Qa0s/Qa1s

c  Evaluation of Q(y)-function values with fus0 and fus1

      fus0h0 = (fus0 - mh0)/sh0
      fus1h0 = (fus1 - mh0)/sh0
      fus0h1 = (fus0 - mh1)/sh1
      fus1h1 = (fus1 - mh1)/sh1

      CALL QfunE(fus0h0, Qfus0h0)
      CALL QfunE(fus1h0, Qfus1h0)
      CALL QfunE(fus0h1, Qfus0h1)
      CALL QfunE(fus1h1, Qfus1h1)

c  Calculation of an error probability by the team effort

      peteam = (Qfus0h0*(1.0-Qa0s) + Qfus1h0*Qa0s) * p0
     *     + ((1.0-Qfus0h1)*(1.0-Qa1s) + (1.0-Qfus1h1)*Qa1s) * p1

c  Calculation of the host sensor error probability and that of
c  communication probability, pehs and pzcom, respectively.
c  These probabilities are TL and TU dependent which means that
c  whenever TL and TU changes, values of pehs and pzcom also change.

      tinc = 0.02
      DO 10 ia = 1, 21
       ib = 0
       DO 30 thr = Ths, Ths + 4.0, tinc
        ib = ib + 1
```

```
      TU(ib) = thr
      TL(ib) = Ths - (FLOAT(ib) - 1.0)*tinc

c  Evaluation for pehs for various TL and TU

      tuh0 = (TU(ib) - mh0)/sh0
      tlh1 = (TL(ib) - mh1)/sh1

      CALL QfunE(tuh0, Qtuh0)
      CALL QfunE(tlh1, Qtlh1)

      pehs(ib) = Qtuh0*p0 + (1.0 - Qtlh1)*p1

c  Evaluation for pzcom for various TL and TU

      tlh0 = (TL(ib) - mh0)/sh0
      tuh1 = (TU(ib) - mh1)/sh1

      CALL QfunE(tlh0, Qtlh0)
      CALL QfunE(tuh1, Qtuh1)

      pzcom(ib) = (Qtlh0 - Qtuh0)*p0 + (Qtlh1 - Qtuh1)*p1

c  Evaluation for the expected probability of error of the system, COST

      cbar(ia,ib) = pehs(ib) + (peteam-pehs(ib)+ccc(ia))*pzcom(ib)
      cf(ib) = cbar(ia,ib)

30    CONTINUE

c  Extract the minimum system cost for each case of ccc(ia)

     CALL FINDMIN(cf,ib,mini)
     cfmin(ia) = cf(mini)
     opthr(ia) = TU(mini)

10   CONTINUE

c  Write OUTPUT DATA .....

    WRITE (10,*) 'Ratio of a priori probabilities:  ', t0
    WRITE (10,*) 'The LRT threshold of Slave Sensor:', Tss
    WRITE (10,*) 'Final Threshold when Us = 0:      ', fus0
    WRITE (10,*) 'Final Threshold when Us = 1:      ', fus1
    WRITE (10,*) 'Error caused by team strategies:  ', peteam
```

```
      DO 50 ic = 1, ib
       WRITE (11,1000) TL(ic), TU(ic), (cbar(id,ic),id=1,10)
 50   CONTINUE

      DO 70 ie = 1, ib
       WRITE (12,1000) TL(ie), TU(ie), (cbar(ig,ie),ig=11,20)
 70   CONTINUE

      DO 90 ih = 1, 21
       WRITE (13,*) ccc(ih), cfmin(ih)
 90   CONTINUE

      DO 110 ii = 1, 21
       WRITE (14,*) opthr(ii), cfmin(ii)
 110  CONTINUE

      DO 130 ij = 1, 21
       WRITE (15,*) ccc(ij), opthr(ij)
 130  CONTINUE

c  Format statements

1000   FORMAT (' ',F6.3,1X,11(F6.4,1X))

      STOP
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE QfunE(xx, erfcx)

c  This function calculates the error function and the complimentary
c  error function for the value "xx"

c  Accuracy is to within 1.5E-07.

      REAL x, xx, erfcx
      REAL a1, a2, a3, a4, a5, p, pi, t

      pi = 3.141592654
      x  = ABS(xx)

      a1 = 0.319381530
      a2 = -0.356563782
```

```
      a3 =  1.781477937
      a4 = -1.821255978
      a5 =  1.330274429

      p = 0.2316419
      t = 1.0/(1.0 + p*x)

      s1 = a1*t + a2*t**2 + a3*t**3 + a4*t**4 + a5*t**5
      s2 = s1*EXP(-(x**2)/2.0)

      IF (xx .GE. 0.0) THEN
       erfcx = s2/SQRT(2.0*pi)
      ELSE IF (xx .LT. 0.0) THEN
       erfcx = 1.0 - s2/SQRT(2.0*pi)
      END IF

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE FINDMIN(array,isize,minindex)

      REAL array(isize)
      INTEGER minindex

      int = 1
11    CONTINUE
      DO 10 i = int+1, isize
       IF (array(int) .GT. array(i)) THEN
        minindex = i
        int = minindex
        GOTO 11
       END IF
10    CONTINUE

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B

Program for Cost Evaluation of 3-Sensor-System

In Appendix B, a FORTRAN program used for numerical evaluations of (2.3.4) substituted with (2.5.4.1), (2.5.4.2), and (3.4.2.9) is contained. The main program is called "THREESENSYS" (THREE SENsor SYStem). As is in Chapter 2, the same subroutines, QfunE and FindMin, are also used.

Outputs of this program are represented by Figure 3.3, Figure 3.4, Figure 3.5, and Figure 3.6. Some of the data are also available from Table 5.2.

```
      PROGRAM THREESENSYS

c*************************************************************************

c Author:     Howard C. Choe
c Organization: Department of Electrical Engineering
c             The University of Virginia, Charlottesville
c Purpose:    Master of Science Research

c This program numerically evaluates expected system costs of a
c three-sensor-system which uses team strategies for Gaussian statistics.

c Variable Description
c   p0    : a priori probability of "No target exists", H0
c   p1    : a priori probability of "Target exists", H1
c For Host Sensor
c   mh0   : mean value of H0 received by HS
c   mh1   : mean value of H1 received by HS
c   sh0   : standard deviation of H0 received by HS
c   sh1   : standard deviation of H1 received by HS
c For Slave Sensor 1
c   ms10   : mean value of H0 received by SS1
c   ms11   : mean value of H1 received by SS1
c   ss10   : standard deviation of H0 received by SS1
c   ss11   : standard deviation of H1 received by SS1
c For Slave Sensor 2
c   ms20   : mean value of H0 received by SS2
c   ms21   : mean value of H1 received by SS2
c   ss20   : standard deviation of H0 received by SS2
c   ss21   : standard deviation of H1 received by SS2
c Thresholds
c   For Host Sensor
c     TL() : lower threshold
c     TU() : upper threshold
c   For Slave Sensor 1
c     Tss1  : LRT optimal threshold
c   For Slave Sensor 2
c     Tss2  : LRT optimal threshold
c Pre-cost constant
c   c00   : deciding H0 given H0
c   c10   : deciding H1 given H0
c   c11   : deciding H1 given H1
c   c01   : deciding H0 given H1
c Team effort cost
c   ccc() : communication cost constant for communicating with two sensor
```

```
c Other variables
c   cfmin() : minimum cost in a single case of run, i.e., for a ccc()
c   opthr() : threshold where cfmin() is occured
c   pehs()  : expected error of the system when there is no communication
c   pzcom() : probability of communication would occur
c   cf()    : expected cost of the system at various of thresholds
c   cbar(,) : same as cf() but saved in 2-D array

c Other variables are commented as program is progressed.

c***************************************************************************

c Declaration

      REAL p0, p1
      REAL mh0, mh1, ms10, ms11, ms20, ms21
      REAL sh0, sh1, ss10, ss11, ss20, ss21
      REAL c00, c10, c11, c01
      REAL Tss1, Tss2

      REAL ccc(21)
      REAL cfmin(21), opthr(21)
      REAL TU(1001), TL(1001)
      REAL pehs(1001), pzcom(1001), cf(1001)
      REAL cbar(21,1001)

      DATA ccc/0.00, 0.05, 0.10, 0.15, 0.20,
     *         0.25, 0.30, 0.35, 0.40, 0.45,
     *         0.50, 0.55, 0.60, 0.65, 0.70,
     *         0.75, 0.80, 0.85, 0.90, 0.95, 1.00/

c Determination of communication cost constant for communicating
c with 2 slave sensors

c The input data

      p0 = 0.5
      p1 = 0.5

      mh0  = -1.0
      mh1  =  1.0
      ms10 = -1.0
      ms11 =  1.0
      ms20 = -1.0
      ms21 =  1.0
```

```
      sigma = 1.0

      c00 = 0.0
      c10 = 1.0
      c11 = 0.0
      c01 = 1.0

c  Assign all standard deviation to the same value

      sh0  = sigma
      sh1  = sigma
      ss10 = sigma
      ss11 = sigma
      ss20 = sigma
      ss21 = sigma

c  Evaluation of the pre-calculated threshold for the team strategy and
c  for the slave sensor

      plamss1 = (c10 - c00)/(c01 - c11)
      plamss2 = (c10 - c00)/(c01 - c11)
      plamt   = plamss1

c  Evaluation of the ratio between a priori probabilities and LRT threshold
c  for the host sensor and the slave sensor, considering each sensor is
c  centralized individually.

      t0   = p0/p1
      Ths  = (mh0 + mh1)/2.0 + (sigma**2/(mh1 - mh0))*LOG(plamt*t0)
      Tss1 = (ms10 + ms11)/2.0
     *     + (sigma**2/(ms11 - ms10))*LOG(plamss1*t0)
      Tss2 = (ms20 + ms21)/2.0
     *     + (sigma**2/(ms21 - ms20))*LOG(plamss2*t0)

c  Slave sensor 1
c  bs10  : integration limit for Q-function under H0
c  bs11  : integration limit for Q-function under H1
c  Qbs10 : probability of making uss1=1 under H0
c  Qbs11 : probability of making uss1=1 under H1

      bs10 = (Tss1 - ms10)/ss10
      bs11 = (Tss1 - ms11)/ss11

      CALL QfunE(bs10, Qbs10)
      CALL QfunE(bs11, Qbs11)
```

```
c Slave sensor 2
c  bs20  : integration limit for Q-function under H0
c  bs21  : integration limit for Q-function under H1
c  Qbs20 : probability of making uss2=1 under H0
c  Qbs21 : probability of making uss2=1 under H1

      bs20 = (Tss2 - ms20)/ss20
      bs21 = (Tss2 - ms21)/ss21

      CALL QfunE(bs20, Qbs20)
      CALL QfunE(bs21, Qbs21)

c FOR COMMUNICATING WITH 2 SLAVE SENSORS
c Evaluation of the final threshold after communication
c  f00 : final threshold when uss1=0 and uss2=0
c  f01 : final threshold when uss1=0 and uss2=1
c  f10 : final threshold when uss1=1 and uss2=0
c  f11 : final threshold when uss1=1 and uss2=1

      f00 = plamss1*t0*(1.0-Qbs10)*(1.0-Qbs20)/((1.0-Qbs11)*(1.0-Qbs21))
      f01 = plamss1*t0*(1.0-Qbs10)*Qbs20/((1.0-Qbs11)*Qbs21)
      f10 = plamss2*t0*Qbs10*(1.0-Qbs20)/(Qbs11*(1.0-Qbs21))
      f11 = plamss2*t0*Qbs10*Qbs20/(Qbs11*Qbs21)

c Evaluation of Q(y)-function values using the above values
c  f00h0 : integration limit with f00 under H0
c  f01h0 : integration limit with f01 under H0
c  f10h0 : integration limit with f10 under H0
c  f11h0 : integration limit with f11 under H0

      f00h0 = (f00 - mh0)/sh0
      f01h0 = (f01 - mh0)/sh0
      f10h0 = (f10 - mh0)/sh0
      f11h0 = (f11 - mh0)/sh0

c  Qf00h0 : probability of making uf=1 with f00 under H0
c  Qf01h0 : probability of making uf=1 with f01 under H0
c  Qf10h0 : probability of making uf=1 with f10 under H0
c  Qf11h0 : probability of making uf=1 with f11 under H0

      CALL QfunE(f00h0, Qf00h0)
      CALL QfunE(f01h0, Qf01h0)
      CALL QfunE(f10h0, Qf10h0)
      CALL QfunE(f11h0, Qf11h0)
```

```
c  f00h0 : integration limit with f00 under H1
c  f01h0 : integration limit with f01 under H1
c  f10h0 : integration limit with f10 under H1
c  f11h0 : integration limit with f11 under H1

      f00h1 = (f00 - mh1)/sh1
      f01h1 = (f01 - mh1)/sh1
      f10h1 = (f10 - mh1)/sh1
      f11h1 = (f11 - mh1)/sh1

c  Qf00h1 : probability of making uf=1 with f00 under H1
c  Qf01h1 : probability of making uf=1 with f01 under H1
c  Qf10h1 : probability of making uf=1 with f10 under H1
c  Qf11h1 : probability of making uf=1 with f11 under H1

      CALL QfunE(f00h1, Qf00h1)
      CALL QfunE(f01h1, Qf01h1)
      CALL QfunE(f10h1, Qf10h1)
      CALL QfunE(f11h1, Qf11h1)

c  Calculation of an error probability by the team strategy
c  with communicating with 2 slave sensors

      peteam = (  Qf00h0*(1.0-Qbs10)*(1.0-Qbs20)
     *          + Qf01h0*(1.0-Qbs10)*Qbs20
     *          + Qf10h0*Qbs10*(1.0-Qbs20)
     *          + Qf11h0*Qbs10*Qbs20 ) * p0
     *        + (  (1.0-Qf00h1)*(1.0-Qbs11)*(1.0-Qbs21)
     *          + (1.0-Qf01h1)*(1.0-Qbs11)*Qbs21
     *          + (1.0-Qf10h1)*Qbs11*(1.0-Qbs21)
     *          + (1.0-Qf11h1)*Qbs11*Qbs21 )*p1

c  Calculation of the host sensor error probability, and
c  that of communication frequency probability with 2 sensors-pehs,
c  pzcom-respectively.

      tinc = 0.02
      DO 10 ia = 1, 21
       ib = 0
       DO 30 thr = Ths, Ths + 4.0, tinc
        ib = ib + 1
        TU(ib) = thr
        TL(ib) = Ths - (FLOAT(ib) - 1.0)*tinc

c  Evaluation for pehs for various TL and TU
```

```
      tuh0 = (TU(ib) - mh0)/sh0
      tlh1 = (TL(ib) - mh1)/sh1

      CALL QfunE(tuh0, Qtuh0)
      CALL QfunE(tlh1, Qtlh1)

      pehs(ib) = Qtuh0*p0 + (1.0 - Qtlh1)*p1

c Evaluation for pzcom for various TL and TU

      tlh0 = (TL(ib) - mh0)/sh0
      tuh1 = (TU(ib) - mh1)/sh1

      CALL QfunE(tlh0, Qtlh0)
      CALL QfunE(tuh1, Qtuh1)

      pzcom(ib) = (Qtlh0 - Qtuh0)*p0 + (Qtlh1 - Qtuh1)*p1

c Evaluation for the expected probability of error of the system, COST

      cbar(ia,ib) = pehs(ib) + (peteam-pehs(ib)+ccc(ia))*pzcom(ib)
      cf(ib) = cbar(ia,ib)

30    CONTINUE

c Extract the minimum system cost for each case of ccc(ia)

     CALL FINDMIN(cf,ib,mini)
     cfmin(ia) = cf(mini)
     opthr(ia) = TU(mini)

10   CONTINUE

c Write OUTPUT DATA .....

    WRITE (10,*) 'Ratio of a priori probabilities: ', t0
    WRITE (10,*) 'LRT Threshold for SS1 ---------: ', tss1
    WRITE (10,*) 'LRT Threshold for SS2 ---------: ', tss2
    WRITE (10,*) 'FT at HS when Us1=0 and Us2=0 -: ', f00
    WRITE (10,*) 'FT at HS when Us1=0 and Us2=1 -: ', f01
    WRITE (10,*) 'FT at HS when Us1=1 and Us2=0 -: ', f10
    WRITE (10,*) 'FT at HS when Us1=1 and Us2=1 -: ', f11
    WRITE (10,*) 'Error caused by Team Strategy -: ', peteam

    DO 50 ic = 1, ib
```

```
      WRITE (11,1000) TL(ic), TU(ic), (cbar(id,ic),id=1,10)
 50   CONTINUE

      DO 70 ie = 1, ib
      WRITE (12,1000) TL(ie), TU(ie), (cbar(ig,ie),ig=11,20)
 70   CONTINUE

      DO 90 ih = 1, 21
      WRITE (13,*) ccc(ih), cfmin(ih)
 90   CONTINUE

      DO 110 ii = 1, 21
      WRITE (14,*) opthr(ii), cfmin(ii)
110   CONTINUE

      DO 130 ij = 1, 21
      WRITE (15,*) ccc(ij), opthr(ij)
130   CONTINUE

c  Format statements

1000  FORMAT (' ',F6.3,1X,11(F6.4,1X))

      STOP
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE QfunE(xx, erfcx)

c  This function calculates the error function and the complimentary
c  error function for the value "xx"

c  Accuracy is to within 1.5E-07.

      REAL x, xx, erfcx
      REAL a1, a2, a3, a4, a5, p, pi, t

      pi = 3.141592654
      x  = ABS(xx)

      a1 =  0.319381530
      a2 = -0.356563782
      a3 =  1.781477937
```

```
      a4 = -1.821255978
      a5 =  1.330274429

      p = 0.2316419
      t = 1.0/(1.0 + p*x)

      s1 = a1*t + a2*t**2 + a3*t**3 + a4*t**4 + a5*t**5
      s2 = s1*EXP(-(x**2)/2.0)

      IF (xx .GE. 0.0) THEN
       erfcx = s2/SQRT(2.0*pi)
      ELSE IF (xx .LT. 0.0) THEN
       erfcx = 1.0 - s2/SQRT(2.0*pi)
      END IF

      RETURN
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE FINDMIN(array,isize,minindex)

      REAL array(isize)
      INTEGER minindex

      int = 1
11    CONTINUE
      DO 10 i = int+1, isize
       IF (array(int) .GT. array(i)) THEN
        minindex = i
        int = minindex
        GOTO 11
       END IF
10    CONTINUE

      RETURN
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX C

Program for Cost Evaluation of 2/3-Sensor-System

The program listed in Appendix C is used for numerical evaluation of 2/3-Sensor-System's expected costs which is described by (4.2.1) substituted with (4.5.1), (4.5.2), (4.5.3), (4.5.4), and (4.5.5).

Figure 4.3, Figure 4.4, Figure 4.5, Figure 4.6, and Table 5.3 are constructed by using outputs from this program.

```
      PROGRAM TWO3SENSYS

c*************************************************************************

c  Author:     Howard C. Choe
c  Organization: Department of Electrical Engineering
c              The University of Virginia, Charlottesville

c  This program numerically evaluates expected costs of
c  a three-sensor-system which uses team strategies for Gaussian
c  statistics.

c  Variable Description
c    p0    : a priori probability of "No target exists", H0
c    p1    : a priori probability of "Target exists", H1
c  For Host Sensor
c    mh0   : mean value of H0 received by HS
c    mh1   : mean value of H1 received by HS
c    sh0   : standard deviation of H0 received by HS
c    sh1   : standard deviation of H1 received by HS
c  For Slave Sensor 1
c    ms10  : mean value of H0 received by SS1
c    ms11  : mean value of H1 received by SS1
c    ss10  : standard deviation of H0 received by SS1
c    ss11  : standard deviation of H1 received by SS1
c  For Slave Sensor 2
c    ms20  : mean value of H0 received by SS2
c    ms21  : mean value of H1 received by SS2
c    ss20  : standard deviation of H0 received by SS2
c    ss21  : standard deviation of H1 received by SS2
c  Thresholds
c   For Host Sensor
c     TL1  : lower threshold 1
c     TL2  : lower threshold 2
c     TU1  : upper threshold 1
c     TU2  : upper threshold 2
c   For Slave Sensor 1
c     Tss1 : LRT optimal threshold
c   For Slave Sensor 2
c     Tss2 : LRT optimal threshold
c  Pre-cost constant
c    c00   : deciding H0 given H0
c    c10   : deciding H1 given H0
c    c11   : deciding H1 given H1
c    c01   : deciding H0 given H1
```

```
c Team effort cost
c   ccc1   : communication cost constant for communicating with one sensor
c   ccc2   : communication cost constant for communicating with two sensor

c*****************************************************************************

c Declaration

      REAL p0, p1
      REAL mh0, mh1, ms10, ms11, ms20, ms21
      REAL sh0, sh1, ss10, ss11, ss20, ss21
      REAL c00, c10, c11, c01
      REAL Tss1, Tss2

      REAL ccc1(21), ccc2(21)
      REAL cfmin(21), opthr(21)
      REAL TU1(1001), TU2(1001), TL1(1001), TL2(1001)
      REAL pehs(1001), pzcom1(1001), pzcom2(1001), cf(1001)
      REAL cbar(21,1001)

      DATA ccc2/0.00, 0.05, 0.10, 0.15, 0.20,
     *        0.25, 0.30, 0.35, 0.40, 0.45,
     *        0.50, 0.55, 0.60, 0.65, 0.70,
     *        0.75, 0.80, 0.85, 0.90, 0.95, 1.00/

c Determination of communication cost constant for communicating
c with 2 slave sensors

      DO 5 i = 1, 21
       ccc1(i) = 0.5*ccc2(i)
5     CONTINUE

c The input data

      p0 = 0.5
      p1 = 0.5

      mh0  = -1.0
      mh1  =  1.0
      ms10 = -1.0
      ms11 =  1.0
      ms20 = -1.0
      ms21 =  1.0

      sigma = 1.0
```

```
      c00 = 0.0
      c10 = 1.0
      c11 = 0.0
      c01 = 1.0

c  Assign all standard deviation to the same value

      sh0  = sigma
      sh1  = sigma
      ss10 = sigma
      ss11 = sigma
      ss20 = sigma
      ss21 = sigma

c  Evaluation of the pre-calculated threshold for the team strategy and
c  for the slave sensor

      plamss1 = (c10 - c00)/(c01 - c11)
      plamss2 = (c10 - c00)/(c01 - c11)
      plamt   = plamss1

c  Evaluation of the ratio between a priori probabilities and LRT threshold
c  for the host sensor and the slave sensor, considering each sensor is
c  centralized individually.

      t0   = p0/p1
      Ths  = (mh0 + mh1)/2.0 + (sigma**2/(mh1 - mh0))*LOG(plamt*t0)
      Tss1 = (ms10 + ms11)/2.0
     *     + (sigma**2/(ms11 - ms10))*LOG(plamss1*t0)
      Tss2 = (ms20 + ms21)/2.0
     *     + (sigma**2/(ms21 - ms20))*LOG(plamss2*t0)

c  Slave sensor 1

      bs10 = (Tss1 - ms10)/ss10
      bs11 = (Tss1 - ms11)/ss11

      CALL QfunE(bs10, Qbs10)
      CALL QfunE(bs11, Qbs11)

c  Slave sensor 2

      bs20 = (Tss2 - ms20)/ss20
      bs21 = (Tss2 - ms21)/ss21
```

```
      CALL QfunE(bs20, Qbs20)
      CALL QfunE(bs21, Qbs21)

c FOR COMMUNICATING WITH 1 SLAVE SENSOR
c Evaluation of the final threshold after communication

      fs0 = plamss1*t0*(1.0-Qbs10)/(1.0-Qbs11)
      fs1 = plamss1*t0*Qbs10/Qbs11

c Evaluation of Q(y)-function values using the above values

      fs0h0 = (fs0 - mh0)/sh0
      fs1h0 = (fs1 - mh0)/sh0

      CALL QfunE(fs0h0, Qfs0h0)
      CALL QfunE(fs1h0, Qfs1h0)

      fs0h1 = (fs0 - mh1)/sh1
      fs1h1 = (fs1 - mh1)/sh1

      CALL QfunE(fs0h1, Qfs0h1)
      CALL QfunE(fs1h1, Qfs1h1)

c Calculation of an error probability by the team strategy
c with communication with 1 slave sensor

      peteam1 = (Qfs0h0*(1.0-Qbs10) + Qfs1h0*Qbs10) * p0
     *     + ((1.0-Qfs0h1)*(1.0-Qbs11) + (1.0-Qfs1h1)*Qbs11) * p1

c FOR COMMUNICATING WITH 2 SLAVE SENSORS
c Evaluation of the final threshold after communication


      f00 = plamss1*t0*(1.0-Qbs10)*(1.0-Qbs20)/((1.0-Qbs11)*(1.0-Qbs21))
      f01 = plamss1*t0*(1.0-Qbs10)*Qbs20/((1.0-Qbs11)*Qbs21)
      f10 = plamss2*t0*Qbs10*(1.0-Qbs20)/(Qbs11*(1.0-Qbs21))
      f11 = plamss2*t0*Qbs10*Qbs20/(Qbs11*Qbs21)

c Evaluation of Q(y)-function values using the above values

      f00h0 = (f00 - mh0)/sh0
      f01h0 = (f01 - mh0)/sh0
      f10h0 = (f10 - mh0)/sh0
      f11h0 = (f11 - mh0)/sh0
```

```
      CALL QfunE(f00h0, Qf00h0)
      CALL QfunE(f01h0, Qf01h0)
      CALL QfunE(f10h0, Qf10h0)
      CALL QfunE(f11h0, Qf11h0)

      f00h1 = (f00 - mh1)/sh1
      f01h1 = (f01 - mh1)/sh1
      f10h1 = (f10 - mh1)/sh1
      f11h1 = (f11 - mhi)/sh1

      CALL QfunE(f00h1, Qf00h1)
      CALL QfunE(f01h1, Qf01h1)
      CALL QfunE(f10h1, Qf10h1)
      CALL QfunE(f11h1, Qf11h1)

c Calculation of an error probability by the team strategy
c with communicating with 2 slave sensors

      peteam2 = ( Qf00h0*(1.0-Qbs10)*(1.0-Qbs20)
     *         + Qf01h0*(1.0-Qbs10)*Qbs20
     *         + Qf10h0*Qbs10*(1.0-Qbs20)
     *         + Qf11h0*Qbs10*Qbs20 ) * p0
     *       + ( (1.0-Qf00h1)*(1.0-Qbs11)*(1.0-Qbs21)
     *         + (1.0-Qf01h1)*(1.0-Qbs11)*Qbs21
     *         + (1.0-Qf10h1)*Qbs11*(1.0-Qbs21)
     *         + (1.0-Qf11h1)*Qbs11*Qbs21 )*p1

c Calculation of the host sensor error probability, that of
c communication frequency probability with 1 sensor, and that
c of communication frequency probability with 2 sensors-pehs,
c pzcom1, and pzcom2-respectively.

c pehs is TL1 and TU1 dependent, pzcom1 depends on TL1 and TL2,
c and TU2 and TU1, and pzcom2 is dependent upon TL2 and TU2.

      tinc = 0.02
      DO 10 ia = 1, 21
       ib = 0
       DO 30 thr = Ths, Ths + 4.0, tinc
        ib = ib + 1
        TU1(ib) = thr
        TL1(ib) = Ths - (FLOAT(ib) - 1.0)*tinc

        TU2(ib) = (TU1(ib)-Ths)/2.0
        TL2(ib) = TL1(ib)+(Ths-TL1(ib))/2.0
```

```
c Evaluation for pehs for various TL and TU

      tu1h0 = (TU1(ib) - mh0)/sh0
      tl1h1 = (TL1(ib) - mh1)/sh1

      CALL QfunE(tu1h0, Qtu1h0)
      CALL QfunE(tl1h1, Qtl1h1)

      pehs(ib) = Qtu1h0*p0 + (1.0 - Qtl1h1)*p1

c Evaluation for pzcom1 for various TL1, TL2, TU1, and TU2

      tl1h0 = (TL1(ib) - mh0)/sh0
      tl2h0 = (TL2(ib) - mh0)/sh0
      tu1h0 = (TU1(ib) - mh0)/sh0
      tu2h0 = (TU2(ib) - mh0)/sh0

      CALL QfunE(tl1h0, Qtl1h0)
      CALL QfunE(tl2h0, Qtl2h0)
      CALL QfunE(tu1h0, Qtu1h0)
      CALL QfunE(tu2h0, Qtu2h0)

      tl1h1 = (TL1(ib) - mh1)/sh1
      tl2h1 = (TL2(ib) - mh1)/sh1
      tu1h1 = (TU1(ib) - mh1)/sh1
      tu2h1 = (TU2(ib) - mh1)/sh1

      CALL QfunE(tl1h1, Qtl1h1)
      CALL QfunE(tl2h1, Qtl2h1)
      CALL QfunE(tu1h1, Qtu1h1)
      CALL QfunE(tu2h1, Qtu2h1)

      pzcom1(ib) = ( Qtl1h0 - Qtl2h0 + Qtu2h0 - Qtu1h0 ) * p0
     *           + ( Qtl1h1 - Qtl2h1 + Qtu2h1 - Qtu1h1 ) * p1

c Evaluation for pzcom2 for various TL2 and TU2

      pzcom2(ib) = (Qtl2h0 - Qtu2h0)*p0 + (Qtl2h1 - Qtu2h1)*p1

c Evaluation for the expected probability of error of the system, COST

      cbar(ia,ib) = pehs(ib)
     *    + (peteam1-pehs(ib)+ccc1(ia))*pzcom1(ib)
     *    + (peteam2-pehs(ib)+ccc2(ia))*pzcom2(ib)
     *    + (pehs(ib)-peteam1-peteam2-ccc1(ia)-ccc2(ia))
```

```
*      *pzcom1(ib)*pzcom2(ib)

        cf(ib) = cbar(ia,ib)

 30     CONTINUE

c Extract the minimum system cost for each case of ccc(ia)

      CALL FINDMIN(cf,ib,mini)
      cfmin(ia) = cf(mini)
      opthr(ia) = TU1(mini)

 10   CONTINUE

c Write OUTPUT DATA .....

      WRITE (10,*) 'Ratio of a priori probabilities --:', t0
      WRITE (10,*) 'LRT Threshold of SS1 -------------:', Tss1
      WRITE (10,*) 'LRT Threshold of SS2 -------------:', Tss2
      WRITE (10,*) 'Error caused by TS using SS1 only :', peteam1
      WRITE (10,*) 'FT at HS when Us1=0 --------------:', fs0
      WRITE (10,*) 'FT at HS when Us1=1 --------------:', fs1
      WRITE (10,*) '     '
      WRITE (10,*) 'Error caused by TS using SS1 & SS2:', peteam2
      WRITE (10,*) 'FT at HS when Us1=0 and Us2=0 ----:', f00
      WRITE (10,*) 'FT at HS when Us1=0 and Us2=1 ----:', f01
      WRITE (10,*) 'FT at HS when Us1=1 and Us2=0 ----:', f10
      WRITE (10,*) 'FT at HS when Us1=1 and Us2=1 ----:', f11

      DO 45 ja = 1, ib
       WRITE (9,*) TL1(ja), TL2(ja), TU2(ja), TU1(ja)
 45   CONTINUE

      DO 50 ic = 1, ib
       WRITE (11,1000) TU1(ic), (cbar(id,ic),id=1,10)
 50   CONTINUE

      DO 70 ie = 1, ib
       WRITE (12,1000) TU1(ie), (cbar(ig,ie),ig=11,20)
 70   CONTINUE

      DO 90 ih = 1, 21
       WRITE (13,*) ccc2(ih), cfmin(ih)
 90   CONTINUE
```

```
      DO 110 ii = 1, 21
       WRITE (14,*) opthr(ii), cfmin(ii)
110   CONTINUE

      DO 130 ij = 1, 21
       WRITE (15,*) ccc2(ij)  opthr(ij)
130   CONTINUE

c  Format statements

1000    FORMAT (' ',11(F6.4,1X))

      STOP
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE QfunE(xx, erfcx)

c  This function calculates the error function and the complimentary
c  error function for the value "xx"

c  Accuracy is to within 1.5E-07.

      REAL x, xx, erfcx
      REAL a1, a2, a3, a4, a5, p, pi, t

      pi = 3.141592654
      x  = ABS(xx)

      a1 =  0.319381530
      a2 = -0.356563782
      a3 =  1.781477937
      a4 = -1.821255978
      a5 =  1.330274429

      p = 0.2316419
      t = 1.0/(1.0 + p*x)

      s1 = a1*t + a2*t**2 + a3*t**3 + a4*t**4 + a5*t**5
      s2 = s1*EXP(-(x**2)/2.0)

      IF (xx .GE. 0.0) THEN
       erfcx = s2/SQRT(2.0*pi)
```

```
      ELSE IF (xx .LT. 0.0) THEN
       erfcx = 1.0 - s2/SQRT(2.0*pi)
      END IF

      RETURN
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE FINDMIN(array,isize,minindex)

      REAL array(isize)
      INTEGER minindex

      int = 1
 11   CONTINUE
      DO 10 i = int+1, isize
       IF (array(int) .GT. array(i)) THEN
        minindex = i
        int = minindex
        GOTO 11
       END IF
 10   CONTINUE

      RETURN
      END


c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX D

Program for Calculation of Dubious Decision Probabilities

In this appendix, a program UNPRO (UNcertainty PRObability) is attached. This program evaluates the dubious decision probability at the host sensor in 2SS, 3SS, and 23SS when the optimal thresholds for given communication cost constants are known. These thresholds can be obtained from the programs attached in Appendix A, Appendix B, and Appendix C.

The information obtained by this program are plotted in Figure 5.1, Figure 5.2, and Figure 5.3. The tabulated data can also be found in Table 5.8, Table 5.9, and Table 5.10.

```
      PROGRAM UNPRO

c*************************************************************************

c This program UNPRO (UNcertainty PRObability) is written to evaluate
c the probability of the observation that falls in the uncertainty
c region of the host sensor. This program reads in the optimal threshold
c locations which are evaluated using programs such as 2sensys.f,
c 3sensys.f, and 2/3sensys.f (These programs are listed in Appendix A, B,
c and C, respectively.).

c*************************************************************************

c Declaration

      REAL otp(3,21), error(3,21)
      REAL qlimR(3,21), qlimL(3,21)
      REAL ytintR(3,21), ytintL(3,21)
      REAL prob(3,21)

c Open data file and rewind

      OPEN (UNIT=11,FILE='fort.11')
      OPEN (UNIT=12,FILE='fort.12')
      OPEN (UNIT=13,FILE='fort.13')

      REWIND (11)
      REWIND (12)
      REWIND (13)

c Statistics for Gaussian observation

      avg   = 1.0
      sigma = 1.0

c Read in input data from files opened

      DO 10 i = 1, 21
       READ (11,*) otp(1,i), error(1,i)
       READ (12,*) otp(2,i), error(2,i)
       READ (13,*) otp(3,i), error(3,i)
10    CONTINUE

c Calculation of limits (qlimR and qlimL) for Q(y)-function
```

```
c  and evaluate the corresponding probabilities.

      DO 30 i = 1, 3
       DO 50 j = 1, 21
        qlimR(i,j) = (otp(i,j)-avg)/sigma
        qlimL(i,j) = (-otp(i,j)-avg)/sigma

        xxR = qlimR(i,j)
        CALL QfunE(xxR,erfcxR)

        xxL = qlimL(i,j)
        CALL QfunE(xxL,erfcxL)

        ytintR(i,j) = erfcxR
        ytintL(i,j) = erfcxL
50      CONTINUE
30     CONTINUE

c  Evaluate the probability of dubious decision for one hypothesis

      DO 70 i = 1, 3
       DO 90 j = 1, 21
        prob(i,j) = ytintL(i,j) - ytintR(i,j)
90      CONTINUE
70     CONTINUE

c  prob(,) is multiplied by 2.0 since prob(,) is the probability of
c  a observation that lands in the uncertainty region of HS given H1.
c  The dubious decision probability, when H0 is considered, is the same.

      DO 110 i = 1, 21
       WRITE (21,1000) otp(1,i), prob(1 i)*2.0
       WRITE (22,1000) otp(2,i), prob(2,i)*2.0
       WRITE (23,1000) otp(3,i), prob(3,i)*2.0
110    CONTINUE

      STOP

1000  FORMAT (' ',F8.3,1X,F8.4)

      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
      SUBROUTINE QfunE(xx, erfcx)

c  This function calculates the error function and the complimentary
c  error function for the value "xx"

c  Accuracy is to within 1.5E-07.

      REAL x, xx, erfcx
      REAL a1, a2, a3, a4, a5, p, pi, t

      pi = 3.141592654
      x  = ABS(xx)

      a1 =  0.319381530
      a2 = -0.356563782
      a3 =  1.781477937
      a4 = -1.821255978
      a5 =  1.330274429

      p = 0.2316419
      t = 1.0/(1.0 + p*x)

      s1 = a1*t + a2*t**2 + a3*t**3 + a4*t**4 + a5*t**5
      s2 = s1*EXP(-(x**2)/2.0)

      IF (xx .GE. 0.0) THEN
       erfcx = s2/SQRT(2.0*pi)
      ELSE IF (xx .LT. 0.0) THEN
       erfcx = 1.0 - s2/SQRT(2.0*pi)
      END IF

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX E

Program Listing of System Simulation

The program, SENSIM (SENsor SIMulation), simulates 2SS, 3SS, and 2/3SS for Gaussian observations. This program incorporates the programs listed in Appendix A, B, and C. These programs - TWOSENSYM, THREESENSYM, and TWO3SENSYM - become subroutines named SETBAND2, SETBAND3, and SETBAND23, respectively. These subroutines return the optimal thresholds location and the final decision thresholds at the host sensor for a given communication cost constant. There are other subroutines which are used to generate Gaussian random observation (or Gaussian random number). Because of the mutually independent observations among sensors, each sensor is provided with its own Gaussian random observation generator.

Outputs from this simulation are presented in Figure 6.1, Figure 6.2, and Figure 6.3. Tabulated data of these figures are in Table 6.1 and Table 6.2.

In the subroutine SETBAND2, SETBAND3, and SETBAND23, all the comments are omitted since they are the same as the programs attached in Appendix A, B, and C.

```
      PROGRAM SENSIM
c*************************************************************************
c AUTHOR: HOWARD C. CHOE
c ORGANIZATION: Department of Electrical Engineering
c           University of Virginia, Charlottesville

c This program simulates the sensor systems ( 2, 3, and 2/3 sensor
c system) which uses team strategies. The host sensor and the slave
c sensors receive independent observations from the binary hypothesis
c environment under Gaussian model.

c*************************************************************************

c Find the optimum thresholds of the host sensor (TL & TU or TL1, TL2,
c TU2, and TU1) for the different system.

      WRITE (6,*) 'ENTER ccc for each system, 2, 3, & 23'
      READ (5,*) ccc2, ccc3, ccc23

      WRITE (6,*) 'ENTER # of iterations desired'
      READ (5,*) nter

      WRITE (6,*) 'ENTER # seed for a random # generation'
      READ (5,*) iseed


      TLRT = 0.0
      CALL SETBAND2(ccc2,TL2,TU2,F20,F21,T2)
      CALL SETBAND3(ccc3,TL3,TU3,F300,F301,F310,F311,T31,T32)
      CALL SETBAND23(ccc23,TL31,TL32,TU32,TU31,
     *          F0,F1,F00,F01,F10,F11,T231,T232)

      WRITE (10,*) '********* ONE-SENSOR-SYSTEM *********'
      WRITE (10,*) ' '
      WRITE (10,*) 'LRT Threshold ---------------: ', TLRT
      WRITE (10,*) '********* TWO-SENSOR-SYSTEM *********'
      WRITE (10,*) 'Lower Threshold -------------: ', TL2
      WRITE (10,*) 'Upper Threshold -------------: ', TU2
      WRITE (10,*) 'LRT threshold of Slave Sensor: ', T2
      WRITE (10,*) 'Final Threshold when Us = 0  : ', F20
      WRITE (10,*) 'Final Threshold when Us = 1  : ', F21
      WRITE (10,*) ' '
      WRITE (10,*) '******** THREE-SENSOR-SYSTEM ********'
```

```
      WRITE (10,*) 'Lower Threshold ------------: ', TL3
      WRITE (10,*) 'Upper Threshold ------------: ', TU3
      WRITE (10,*) 'LRT threshold of SS1 --------: ', T31
      WRITE (10,*) 'LRT threshold of SS2 --------: ', T32
      WRITE (10,*) 'FT when Us1 = 0 & Us2 = 0 ---: ', F300
      WRITE (10,*) 'FT when Us1 = 0 & Us2 = 1 ---: ', F301
      WRITE (10,*) 'FT when Us1 = 1 & Us2 = 0 ---: ', F310
      WRITE (10,*) 'FT when Us1 = 1 & Us2 = 1 ---: ', F311
      WRITE (10,*) '   '
      WRITE (10,*) '****** TWO/THREE-SENSOR-SYSTEM ******'
      WRITE (10,*) 'Lower Threshold 1 -----------: ', TL31
      WRITE (10,*) 'Lower Threshold 2 -----------: ', TL32
      WRITE (10,*) 'Upper Threshold 2 -----------: ', TU32
      WRITE (10,*) 'Upper Threshold 1 -----------: ', TU31
      WRITE (10,*) 'LRT threshold of SS1 --------: ', T231
      WRITE (10,*) 'LRT threshold of SS2 --------: ', T232
      WRITE (10,*) 'FT when Us1 = 0 -------------: ', F0
      WRITE (10,*) 'FT when Us1 = 0 -------------: ', F1
      WRITE (10,*) 'FT when Us1 = 0 & Us2 = 0 ---: ', F00
      WRITE (10,*) 'FT when Us1 = 0 & Us2 = 1 ---: ', F01
      WRITE (10,*) 'FT when Us1 = 1 & Us2 = 0 ---: ', F10
      WRITE (10,*) 'FT when Us1 = 1 & Us2 = 1 ---: ', F11

c  Standard deviation of observation

      sigma = 1.0

c  ITERATION STARTS

      ie0 = 0
      ie1 = 0

      icd1 = 0
      ifa1 = 0
      imt1 = 0

      icd2 = 0
      ifa2 = 0
      imt2 = 0

      icd3 = 0
      ifa3 = 0
      imt3 = 0

      icd23 = 0
```

```
      ifa23 = 0
      imt23 = 0

c  Get system clock time for random seeds

c     itm = mclock()
c     WRITE (6,*) 'itm = ', itm
c     iseed   = 74591 + 2*MOD(1000*itm,500)

      CALL SRAND(iseed)

      DO 10 ia = 1, nter

c  Generate Environment

11      CALL GENENV(ienv)
        env = FLOAT(ienv)
        IF (env .EQ. -1.0) THEN
         ie0 = ie0 + 1
        ELSE IF (env .EQ. 1.0) THEN
         ie1 = ie1 + 1
        ELSE
         WRITE (6,*) '### Generated ENV is NOT either -1 or 1 ###'
         GO TO 11
        END IF

c  Generate Observations at each sensors
c   For 1-Sensor-System
        CALL HS1(env,sigma,yh1)

c   For 2-Sensor-System
c      CALL HS2(env,sigma,yh2)
       CALL SS2(env,sigma,ys2)

c   For 3-Sensor-System
c      CALL HS3(env,sigma,yh3)
c      CALL SS31(env,sigma,ys31)
       CALL SS32(env,sigma,ys32)

c   For 23-Sensor-System
c      CALL HS23(env,sigma,yh23)
c      CALL SS231(env,sigma,ys231)
c      CALL SS232(env,sigma,ys232)

       yh2   = yh1
```

```
      yh3  = yh1
      yh23 = yh1

      ys31 = ys2
      ys231 = ys2
      ys232 = ys32

c     WRITE (33,1100) env,yh1,yh2,ys2,yh3,ys31,ys32,yh23,ys231,ys232
c1100  FORMAT (' ',10(F7.3,1X))

c Single Sensor using LRT Threshold
      IF (yh1 .LE. TLRT) THEN
       uh1 = -1.0
      ELSE IF (yh1 .GT. TLRT) THEN
       uh1 = 1.0
      END IF
      iuh1 = INT(uh1)

c  Count False alarm, missing target, and correct detection
      IF (iuh1 .EQ. ienv) THEN
       icd1 = icd1 + 1
      ELSE IF (iuh1.EQ.1 .AND. ienv.EQ.-1) THEN
       ifa1 = ifa1 + 1
      ELSE IF (iuh1.EQ.-1 .AND. ienv.EQ.1) THEN
       imt1 = imt1 + 1
      END IF

c Use Team Strategies to make Decision
c  For 2-Sensor-System
      IF (yh2 .LE. TL2) THEN
       uh2  = -1.0
      ELSE IF (yh2 .GE. TU2) THEN
       uh2  = 1.0
      ELSE IF (yh2.GT.TL2 .AND. yh2.LT.TU2) THEN
       IF (ys2 .LE. T2) THEN
        us2 = -1.0
        IF (yh2 .LE. F20) THEN
         uh2 = -1.0
        ELSE IF (yh2 .GT. F20) THEN
         uh2 = 1.0
        END IF
       ELSE IF (ys2 .GT. T2) THEN
        us2 = 1.0
        IF (yh2 .LE. F21) THEN
         uh2 = -1.0
```

```
          ELSE IF (yh2 .GT. F21) THEN
           uh2 = 1.0
          END IF
         END IF
        END IF
        iuh2 = INT(uh2)

c   Count False alarm, missing target, and correct detection
        IF (iuh2 .EQ. ienv) THEN
         icd2 = icd2 + 1
        ELSE IF (iuh2.EQ.1 .AND. ienv.EQ.-1) THEN
         ifa2 = ifa2 + 1
        ELSE IF (iuh2.EQ.-1 .AND. ienv.EQ.1) THEN
         imt2 = imt2 + 1
        END IF

c   For 3-Sensor-System
        IF (yh3 .LE. TL3) THEN
         uh3  = -1.0
        ELSE IF (yh3 .GE. TU3) THEN
         uh3  = 1.0
        ELSE IF (yh3.GT.TL3 .AND. yh3.LT.TU3) THEN
         IF (ys31.LE.T31 .AND. ys32.LE.T32) THEN
          us31 = -1.0
          us32 = -1.0
          IF (yh3 .LE. F300) THEN
           uh3 = -1.0
          ELSE IF (yh3 .GT. F300) THEN
           uh3 = 1.0
          END IF
         ELSE IF (ys31.LE.T31 .AND. ys32.GT.T32) THEN
          us31 = -1.0
          us32 =  1.0
          IF (yh3 .LE. F301) THEN
           uh3 = -1.0
          ELSE IF (yh3 .GT. F301) THEN
           uh3 = ' 0
          END IF
         ELSE IF (ys31.GT.T31 .AND. ys32.LE.T32) THEN
          us31 = 1.0
          us32 = -1.0
          IF (yh3 .LE. F310) THEN
           uh3 = -1.0
          ELSE IF (yh3 .GT. F310) THEN
           uh3 = 1.0
```

```
       END IF
      ELSE IF (ys31.GT.T31 .AND. ys32.GT.T32) THEN
       us31 = 1.0
       us32 = 1.0
       IF (yh3 .LE. F311) THEN
        uh3 = -1.0
       ELSE IF (yh3 .GT. F311) THEN
        uh3 = 1.0
       END IF
      END IF
     END IF
     iuh3 = INT(uh3)
c    Count False alarm, missing target, and correct detection
     IF (iuh3 .EQ. ienv) THEN
      icd3 = icd3 + 1
     ELSE IF (iuh3.EQ.1 .AND. ienv.EQ.-1) THEN
      ifa3 = ifa3 + 1
     ELSE IF (iuh3.EQ.-1 .AND. ienv.EQ.1) THEN
      imt3 = imt3 + 1
     END IF

c    For 2/3-Sensor-System
     IF (yh23 .LE. TL31) THEN
      uh23 = -1.0
     ELSE IF (yh23 .GE. TU31) THEN
      uh23 = 1.0
     ELSE IF (yh23.GT.TL31 .AND. yh23.LT.TL32 .OR.
   *        yh23.GT.TU32 .AND. yh23.LT.TU31) THEN
      IF (ys231 .LE. T231) THEN
       us231 = -1.0
       IF (yh23 .LE. F0) THEN
        uh23 = -1.0
       ELSE IF (yh23 .GT. F0) THEN
        uh23 = 1.0
       END IF
      ELSE IF (ys231 .GT. T231) THEN
       us231 = 1.0
       IF (yh23 .LE. F1) THEN
        uh23 = -1.0
       ELSE IF (yh23 .GT. F1) THEN
        uh23 = 1.0
       END IF
      END IF
     ELSE IF (yh23.GE.TL32 .AND. yh23.LE.TU32) THEN
      IF (ys231.LE.T231 .AND. ys232.LE.T232) THEN
```

```
          us231 = -1.0
          us232 = -1.0
          IF (yh23 .LE. F00) THEN
            uh23 = -1.0
          ELSE IF (yh23 .GT. F00) THEN
            uh23 = 1.0
          END IF
        ELSE IF (ys231.LE.T231 .AND. ys232.GT.T232) THEN
          us231 = -1.0
          us232 =  1.0
          IF (yh23 .LE. F01) THEN
            uh23 = -1.0
          ELSE IF (yh23 .GT. F01) THEN
            uh23 = 1.0
          END IF
        ELSE IF (ys231.GT.T231 .AND. ys232.LE.T232) THEN
          us231 = 1.0
          us232 = -1.0
          IF (yh23 .LE. F10) THEN
            uh23 = -1.0
          ELSE IF (yh23 .GT. F10) THEN
            uh23 = 1.0
          END IF
        ELSE IF (ys231.GT.T231 .AND. ys232.GT.T232) THEN
          us231 = 1.0
          us232 = 1.0
          IF (yh23 .LE. F11) THEN
            uh23 = -1.0
          ELSE IF (yh23 .GT. F11) THEN
            uh23 = 1.0
          END IF
        END IF
      END IF
      iuh23 = INT(uh23)

c   Count False alarm, missing target, and correct detection
      IF (iuh23 .EQ. ienv) THEN
        icd23 = icd23 + 1
      ELSE IF (iuh23.EQ.1 .AND. ienv.EQ.-1) THEN
        ifa23 = ifa23 + 1
      ELSE IF (iuh23.EQ.-1 .AND. ienv.EQ.1) THEN
        imt23 = imt23 + 1
      END IF

10    CONTINUE
```

```
c Find the percentage of 0's and 1's in total environment generated

      pev0 = 100.0*FLOAT(ie0)/FLOAT(ia)
      pev1 = 100.0*FLOAT(ie1)/FLOAT(ia)

      WRITE (10,*) '    '
      WRITE (10,*) '@@@@@@ % of 0 or 1 of the Environment @@@@@'
      WRITE (10,*) '% of 0s :', pev0
      WRITE (10,*) '% of 1s :', pev1

c Find the percentage of correct detection, false alarm, and missing target
c   for each system.

c 1-Sensor-System
      pcd1 = 100.0*FLOAT(icd1)/FLOAT(ia)
      pfa1 = 100.0*FLOAT(ifa1)/FLOAT(ia)
      pmt1 = 100.0*FLOAT(imt1)/FLOAT(ia)

c 2-Sensor-System
      pcd2 = 100.0*FLOAT(icd2)/FLOAT(ia)
      pfa2 = 100.0*FLOAT(ifa2)/FLOAT(ia)
      pmt2 = 100.0*FLOAT(imt2)/FLOAT(ia)

c 3-Sensor-System
      pcd3 = 100.0*FLOAT(icd3)/FLOAT(ia)
      pfa3 = 100.0*FLOAT(ifa3)/FLOAT(ia)
      pmt3 = 100.0*FLOAT(imt3)/FLOAT(ia)

c 23-Sensor-System
      pcd23 = 100.0*FLOAT(icd23)/FLOAT(ia)
      pfa23 = 100.0*FLOAT(ifa23)/FLOAT(ia)
      pmt23 = 100.0*FLOAT(imt23)/FLOAT(ia)

c WRITE the percentages
      WRITE (10,*) '  '
      WRITE (10,*) '// % of CD, FA, and MT for 1-Sensor-System \'
      WRITE (10,*) 'Correct Decision %: ', pcd1
      WRITE (10,*) 'False Alarm % ----: ', pfa1
      WRITE (10,*) 'Missing Target %  : ', pmt1
      WRITE (10,*) 'CD + FA + MT in % : ', pcd1+pfa1+pmt1
      WRITE (10,*) '  '
      WRITE (10,*) '// % of CD, FA, and MT for 2-Sensor-System \'
      WRITE (10,*) 'Correct Decision %: ', pcd2
      WRITE (10,*) 'False Alarm % ----: ', pfa2
```

```
      WRITE (10,*) 'Missing Target %  : ', pmt2
      WRITE (10,*) 'CD + FA + MT in % : ', pcd2+pfa2+pmt2
      WRITE (10,*) '  '
      WRITE (10,*) '// % of CD, FA, and MT for 3-Sensor-System \'
      WRITE (10,*) 'Correct Decision %: ', pcd3
      WRITE (10,*) 'False Alarm % ----: ', pfa3
      WRITE (10,*) 'Missing Target %  : ', pmt3
      WRITE (10,*) 'CD + FA + MT in % : ', pcd3+pfa3+pmt3
      WRITE (10,*) '  '
      WRITE (10,*) '// % of CD, FA, and MT for 23-Sensor-System \'
      WRITE (10,*) 'Correct Decision %: ', pcd23
      WRITE (10,*) 'False Alarm % ----: ', pfa23
      WRITE (10,*) 'Missing Target %  : ', pmt23
      WRITE (10,*) 'CD + FA + MT in % : ', pcd23+pfa23+pmt23

      STOP

      END

c%% Subroutine for the generation of environment %%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE GENENV(ienv)

      irn = irand()
      xrn = FLOAT(irn)/FLOAT(2**15 - 1)
      renv = xrn - 0.5

      IF (renv .LE. 0.0) THEN
       ienv = -1
      ELSE IF (renv .GT. 0.0) THEN
       ienv =  1
      END IF

      RETURN
      END

c%% Subroutine for Single-Sensor-System %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE HS1(env,sigma,yh1)

      a = 0.0
      DO 10 i = 1, 12
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
```

```
10  CONTINUE

    yh1 = (a - 6.0)*sigma + env

    RETURN
    END

c%% Subroutines for 2-Sensor-System %%%%%%%%%%%%%%%%%%%%%%%%%

    SUBROUTINE HS2(env,sigma,yh2)

    a = 0.0
    DO 10 i = 1, 12
     irn = irand()
     xrn = FLOAT(irn)/FLOAT(2**15 - 1)
     a = a + xrn
10  CONTINUE

    yh2 = (a - 6.0)*sigma + env

    RETURN
    END

c---------------------------------------------------------------------

    SUBROUTINE SS2(env,sigma,ys2)

    a = 0.0
    DO 10 i = 1, 12
     irn = irand()
     xrn = FLOAT(irn)/FLOAT(2**15 - 1)
     a = a + xrn
10  CONTINUE

    ys2 = (a - 6.0)*sigma + env

    RETURN
    END

c%% Subroutines for 3-Sensor-System %%%%%%%%%%%%%%%%%%%%%%%%%

    SUBROUTINE HS3(env,sigma,yh3)

    a = 0.0
    DO 10 i = 1, 12
```

```
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
 10    CONTINUE

      yh3 = (a - 6.0)*sigma + env

      RETURN
      END

c---------------------------------------------------------------------

      SUBROUTINE SS31(env,sigma,ys31)

      a = 0.0
      DO 10 i = 1, 12
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
 10    CONTINUE

      ys31 = (a - 6.0)*sigma + env

      RETURN
      END

c---------------------------------------------------------------------

      SUBROUTINE SS32(env,sigma,ys32)

      a = 0.0
      DO 10 i = 1, 12
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
 10    CONTINUE

      ys32 = (a - 6.0)*sigma + env

      RETURN
      END

c%% Subroutines for 23-Sensor-System %%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE HS23(env,sigma,yh23)
```

```
      a = 0.0
      DO 10 i = 1, 12
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
10     CONTINUE

      yh23 = (a - 6.0)*sigma + env

      RETURN
      END

c------------------------------------------------------------------------

      SUBROUTINE SS231(env,sigma,ys231)

      a = 0.0
      DO 10 i = 1, 12
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
10     CONTINUE

      ys231 = (a - 6.0)*sigma + env

      RETURN
      END

c------------------------------------------------------------------------

      SUBROUTINE SS232(env,sigma,ys232)

      a = 0.0
      DO 10 i = 1, 12
       irn = irand()
       xrn = FLOAT(irn)/FLOAT(2**15 - 1)
       a = a + xrn
10     CONTINUE

      ys232 = (a - 6.0)*sigma + env

      RETURN
      END
```

```
c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE SETBAND2(ccc2,TL2,TU2,F20,F21,T2)

      REAL p0, p1
      REAL mh0, mh1, ms0, ms1
      REAL sh0, sh1, ss0, ss1
      REAL c00, c10, c11, c01
      REAL Tss

      REAL ccc2, cfmin, oplthr, oputhr
      REAL TU(1001), TL(1001)
      REAL pehs(1001), pzcom(1001)
      REAL cbar(1001)

      p0 = 0.5
      p1 = 0.5

      mh0 = -1.0
      mh1 =  1.0
      ms0 = -1.0
      ms1 =  1.0

      sigma = 1.0

      c00 = 0.0
      c10 = 1.0
      c11 = 0.0
      c01 = 1.0

      sh0 = sigma
      sh1 = sigma
      ss0 = sigma
      ss1 = sigma

      plamss = (c10 - c00)/(c01 - c11)
      plamt  = plamss

      t0  = p0/p1
      Ths = (mh0 + mh1)/2.0 + (sigma**2/(mh1 - mh0))*LOG(plamt*t0)
      Tss = (ms0 + ms1)/2.0 + (sigma**2/(ms1 - ms0))*LOG(plamss*t0)

      a0s = (Tss - ms0)/ss0
      a1s = (Tss - ms1)/ss1
```

```
      CALL QfunE(a0s, Qa0s)
      CALL QfunE(a1s, Qa1s)

      fus0 = plamss * t0 * (1.0 - Qa0s)/(1.0 - Qa1s)
      fus1 = plamss * t0 * Qa0s/Qa1s

      fus0h0 = (fus0 - mh0)/sh0
      fus1h0 = (fus1 - mh0)/sh0
      fus0h1 = (fus0 - mh1)/sh1
      fus1h1 = (fus1 - mh1)/sh1

      CALL QfunE(fus0h0, Qfus0h0)
      CALL QfunE(fus1h0, Qfus1h0)
      CALL QfunE(fus0h1, Qfus0h1)
      CALL QfunE(fus1h1, Qfus1h1)

      peteam = (Qfus0h0*(1.0-Qa0s) + Qfus1h0*Qa0s) * p0
     *     + ((1.0-Qfus0h1)*(1.0-Qa1s) + (1.0-Qfus1h1)*Qa1s) * p1

      tinc = 0.02
      ib = 0
      DO 30 thr = Ths, Ths + 2.0, tinc
       ib = ib + 1
       TU(ib) = thr
       TL(ib) = Ths - (FLOAT(ib) - 1.0)*tinc

       tuh0 = (TU(ib) - mh0)/sh0
       tlh1 = (TL(ib) - mh1)/sh1

       CALL QfunE(tuh0, Qtuh0)
       CALL QfunE(tlh1, Qtlh1)

       pehs(ib) = Qtuh0*p0 + (1.0 - Qtlh1)*p1

       tlh0 = (TL(ib) - mh0)/sh0
       tuh1 = (TU(ib) - mh1)/sh1

       CALL QfunE(tlh0, Qtlh0)
       CALL QfunE(tuh1, Qtuh1)

       pzcom(ib) = (Qtlh0 - Qtuh0)*p0 + (Qtlh1 - Qtuh1)*p1

       cbar(ib) = pehs(ib) + (peteam-pehs(ib)+ccc2)*pzcom(ib)

30    CONTINUE
```

```
      CALL FINDMIN(cbar,ib,mini)
      cfmin = cbar(mini)
      oplthr = TL(mini)
      oputhr = TU(mini)

      TL2 = oplthr
      TU2 = oputhr
      F20 = fus0
      F21 = -fus0
      T2  = Tss

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE SETBAND3(ccc3,TL3,TU3,F300,F301,F310,F311,T31,T32)

      REAL p0, p1
      REAL mh0, mh1, ms10, ms11, ms20, ms21
      REAL sh0, sh1, ss10, ss11, ss20, ss21
      REAL c00, c10, c11, c01
      REAL Tss1, Tss2

      REAL ccc3
      REAL cfmin, oplthr, oputhr
      REAL TU(1001), TL(1001)
      REAL pchs(1001), pzcom(1001)
      REAL cbar(1001)

      p0 = 0.5
      p1 = 0.5

      mh0  = -1.0
      mh1  =  1.0
      ms10 = -1.0
      ms11 =  1.0
      ms20 = -1.0
      ms21 =  1.0

      sigma = 1.0

      c00 = 0.0
      c10 = 1.0
```

```
c11 = 0.0
c01 = 1.0

sh0  = sigma
sh1  = sigma
ss10 = sigma
ss11 = sigma
ss20 = sigma
ss21 = sigma

plamss1 = (c10 - c00)/(c01 - c11)
plamss2 = (c10 - c00)/(c01 - c11)
plamt   = plamss1

t0   = p0/p1
Ths  = (mh0 + mh1)/2.0 + (sigma**2/(mh1 - mh0))*LOG(plamt*t0)
Tss1 = (ms10 + ms11)/2.0
*      + (sigma**2/(ms11 - ms10))*LOG(plamss1*t0)
Tss2 = (ms20 + ms21)/2.0
*      + (sigma**2/(ms21 - ms20))*LOG(plamss2*t0)

bs10 = (Tss1 - ms10)/ss10
bs11 = (Tss1 - ms11)/ss11

CALL QfunE(bs10, Qbs10)
CALL QfunE(bs11, Qbs11)

bs20 = (Tss2 - ms20)/ss20
bs21 = (Tss2 - ms21)/ss21

CALL QfunE(bs20, Qbs20)
CALL QfunE(bs21, Qbs21)

f00 = plamss1*t0*(1.0-Qbs10)*(1.0-Qbs20)/((1.0-Qbs11)*(1.0-Qbs21))
f01 = plamss1*t0*(1.0-Qbs10)*Qbs20/((1.0-Qbs11)*Qbs21)
f10 = plamss2*t0*Qbs10*(1.0-Qbs20)/(Qbs11*(1.0-Qbs21))
f11 = plamss2*t0*Qbs10*Qbs20/(Qbs11*Qbs21)

f00h0 = (f00 - mh0)/sh0
f01h0 = (f01 - mh0)/sh0
f10h0 = (f10 - mh0)/sh0
f11h0 = (f11 - mh0)/sh0

CALL QfunE(f00h0, Qf00h0)
CALL QfunE(f01h0, Qf01h0)
```

```
CALL QfunE(f10h0, Qf10h0)
CALL QfunE(f11h0, Qf11h0)

f00h1 = (f00 - mh1)/sh1
f01h1 = (f01 - mh1)/sh1
f10h1 = (f10 - mh1)/sh1
f11h1 = (f11 - mh1)/sh1

CALL QfunE(f00h1, Qf00h1)
CALL QfunE(f01h1, Qf01h1)
CALL QfunE(f10h1, Qf10h1)
CALL QfunE(f11h1, Qf11h1)

peteam = ( Qf00h0*(1.0-Qbs10)*(1.0-Qbs20)
*         + Qf01h0*(1.0-Qbs10)*Qbs20
*         + Qf10h0*Qbs10*(1.0-Qbs20)
*         + Qf11h0*Qbs10*Qbs20 ) * p0
*       + ( (1.0-Qf00h1)*(1.0-Qbs11)*(1.0-Qbs21)
*         + (1.0-Qf01h1)*(1.0-Qbs11)*Qbs21
*         + (1.0-Qf10h1)*Qbs11*(1.0-Qbs21)
*         + (1.0-Qf11h1)*Qbs11*Qbs21 )*p1

tinc = 0.02
ib = 0
DO 30 thr = Ths, Ths + 2.0, tinc
 ib = ib + 1
 TU(ib) = thr
 TL(ib) = Ths - (FLOAT(ib) - 1.0)*tinc

 tuh0 = (TU(ib) - mh0)/sh0
 tlh1 = (TL(ib) - mh1)/sh1

 CALL QfunE(tuh0, Qtuh0)
 CALL QfunE(tlh1, Qtlh1)

 pchs(ib) = Qtuh0*p0 + (1.0 - Qtlh1)*p1

 tlh0 = (TL(ib) - mh0)/sh0
 tuh1 = (TU(ib) - mh1)/sh1

 CALL QfunE(tlh0, Qtlh0)
 CALL QfunE(tuh1, Qtuh1)

 pzcom(ib) = (Qtlh0 - Qtuh0)*p0 + (Qtlh1 - Qtuh1)*p1
```

```
      cbar(ib) = pehs(ib) + (peteam-pehs(ib)+ccc3)*pzcom(ib)

30    CONTINUE

      CALL FINDMIN(cbar,ib,mini)
      cfmin = cbar(mini)
      oplthr = TL(mini)
      oputhr = TU(mini)

      TL3  = oplthr
      TU3  = oputhr
      F300 = f00
      F301 = f01
      F310 = f10
      F311 = -f00
      T31  = Tss1
      T32  = Tss2

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE SETBAND23(ccc23,TL31,TL32,TU32,TU31,
     *                F0,F1,F00,F01,F10,F11,T231,T232)

      REAL p0, p1
      REAL mh0, mh1, ms10, ms11, ms20, ms21
      REAL sh0, sh1, ss10, ss11, ss20, ss21
      REAL c00, c10, c11, c01
      REAL Tss1, Tss2

      REAL ccc1, ccc2
      REAL cfmin
      REAL optl1, optl2, optu2, optu1
      REAL TU1(1001), TU2(1001), TL1(1001), TL2(1001)
      REAL pehs(1001), pzcom1(1001), pzcom2(1001)
      REAL cbar(1001)

      ccc1 = ccc23/2.0
      ccc2 = ccc23

      p0 = 0.5
      p1 = 0.5
```

```
mh0  = -1.0
mh1  =  1.0
ms10 = -1.0
ms11 =  1.0
ms20 = -1.0
ms21 =  1.0

sigma = 1.0

c00 = 0.0
c10 = 1.0
c11 = 0.0
c01 = 1.0

sh0  = sigma
sh1  = sigma
ss10 = sigma
ss11 = sigma
ss20 = sigma
ss21 = sigma

plamss1 = (c10 - c00)/(c01 - c11)
plamss2 = (c10 - c00)/(c01 - c11)
plamt   = plamss1

t0   = p0/p1
Ths  = (mh0 + mh1)/2.0 + (sigma**2/(mh1 - mh0))*LOG(plamt*t0)
Tss1 = (ms10 + ms11)/2.0
*     + (sigma**2/(ms11 - ms10))*LOG(plamss1*t0)
Tss2 = (ms20 + ms21)/2.0
*     + (sigma**2/(ms21 - ms20))*LOG(plamss2*t0)

bs10 = (Tss1 - ms10)/ss10
bs11 = (Tss1 - ms11)/ss11

CALL QfunE(bs10, Qbs10)
CALL QfunE(bs11, Qbs11)

bs20 = (Tss2 - ms20)/ss20
bs21 = (Tss2 - ms21)/ss21

CALL QfunE(bs20, Qbs20)
CALL QfunE(bs21, Qbs21)

fs0 = plamss1*t0*(1.0-Qbs10)/(1.0-Qbs11)
```

```
fs1 = plamss1*t0*Qbs10/Qbs11

fs0h0 = (fs0 - mh0)/sh0
fs1h0 = (fs1 - mh0)/sh0

CALL QfunE(fs0h0, Qfs0h0)
CALL QfunE(fs1h0, Qfs1h0)

fs0h1 = (fs0 - mh1)/sh1
fs1h1 = (fs1 - mh1)/sh1

CALL QfunE(fs0h1, Qfs0h1)
CALL QfunE(fs1h1, Qfs1h1)

peteam1 = (Qfs0h0*(1.0-Qbs10) + Qfs1h0*Qbs10) * p0
*       + ((1.0-Qfs0h1)*(1.0-Qbs11) + (1.0-Qfs1h1)*Qbs11) * p1

f00 = plamss1*t0*(1.0-Qbs10)*(1.0-Qbs20)/((1.0-Qbs11)*(1.0-Qbs21))
f01 = plamss1*t0*(1.0-Qbs10)*Qbs20/((1.0-Qbs11)*Qbs21)
f10 = plamss2*t0*Qbs10*(1.0-Qbs20)/(Qbs11*(1.0-Qbs21))
f11 = plamss2*t0*Qbs10*Qbs20/(Qbs11*Qbs21)

f00h0 = (f00 - mh0)/sh0
f01h0 = (f01 - mh0)/sh0
f10h0 = (f10 - mh0)/sh0
f11h0 = (f11 - mh0)/sh0

CALL QfunE(f00h0, Qf00h0)
CALL QfunE(f01h0, Qf01h0)
CALL QfunE(f10h0, Qf10h0)
CALL QfunE(f11h0, Qf11h0)

f00h1 = (f00 - mh1)/sh1
f01h1 = (f01 - mh1)/sh1
f10h1 = (f10 - mh1)/sh1
f11h1 = (f11 - mh1)/sh1

CALL QfunE(f00h1, Qf00h1)
CALL QfunE(f01h1, Qf01h1)
CALL QfunE(f10h1, Qf10h1)
CALL QfunE(f11h1, Qf11h1)

peteam2 = ( Qf00h0*(1.0-Qbs10)*(1.0-Qbs20)
*         + Qf01h0*(1.0-Qbs10)*Qbs20
*         + Qf10h0*Qbs10*(1.0-Qbs20)
```

```
*        + Qf11h0*Qbs10*Qbs20 ) * p0
*      + (  (1.0-Qf00h1)*(1.0-Qbs11)*(1.0-Qbs21)
*        + (1.0-Qf01h1)*(1.0-Qbs11)*Qbs21
*        + (1.0-Qf10h1)*Qbs11*(1.0-Qbs21)
*        + (1.0-Qf11h1)*Qbs11*Qbs21 )*p1

      tinc = 0.02
      ib = 0
      DO 30 thr = Ths, Ths + 2.0, tinc
       ib = ib + 1
       TU1(ib) = thr
       TL1(ib) = Ths - (FLOAT(ib) - 1.0)*tinc

       TU2(ib) = (TU1(ib)+Ths)/2.0
       TL2(ib) = TL1(ib)+(Ths-TL1(ib))/2.0

       tu1h0 = (TU1(ib) - mh0)/sh0
       tl1h1 = (TL1(ib) - mh1)/sh1

       CALL QfunE(tu1h0, Qtu1h0)
       CALL QfunE(tl1h1, Qtl1h1)

       pehs(ib) = Qtu1h0*p0 + (1.0 - Qtl1h1)*p1

       tl1h0 = (TL1(ib) - mh0)/sh0
       tl2h0 = (TL2(ib) - mh0)/sh0
       tu1h0 = (TU1(ib) - mh0)/sh0
       tu2h0 = (TU2(ib) - mh0)/sh0

       CALL QfunE(tl1h0, Qtl1h0)
       CALL QfunE(tl2h0, Qtl2h0)
       CALL QfunE(tu1h0, Qtu1h0)
       CALL QfunE(tu2h0, Qtu2h0)

       tl1h1 = (TL1(ib) - mh1)/sh1
       tl2h1 = (TL2(ib) - mh1)/sh1
       tu1h1 = (TU1(ib) - mh1)/sh1
       tu2h1 = (TU2(ib) - mh1)/sh1

       CALL QfunE(tl1h1, Qtl1h1)
       CALL QfunE(tl2h1, Qtl2h1)
       CALL QfunE(tu1h1, Qtu1h1)
       CALL QfunE(tu2h1, Qtu2h1)

       pzcom1(ib) = ( Qtl1h0 - Qtl2h0 + Qtu2h0 - Qtu1h0 ) * p0
```

```
*         + ( Qtl1h1 - Qtl2h1 + Qtu2h1 - Qtu1h1 ) * p1

      pzcom2(ib) = (Qtl2h0 - Qtu2h0)*p0 + (Qtl2h1 - Qtu2h1)*p1

      cbar(ib) = pehs(ib)
*     + (peteam1-pehs(ib)+ccc1)*pzcom1(ib)
*     + (peteam2-pehs(ib)+ccc2)*pzcom2(ib)
*     + (pehs(ib)-peteam1-peteam2-ccc1-ccc2)
*       *pzcom1(ib)*pzcom2(ib)

30    CONTINUE

      CALL FINDMIN(cbar,ib,mini)
      cfmin = cbar(mini)
      optl1 = TL1(mini)
      optl2 = TL2(mini)
      optu2 = TU2(mini)
      optu1 = TU1(mini)

      TL31 = optl1
      TL32 = optl2
      TU32 = optu2
      TU31 = optu1
      F0  = fs0
      F1  = -fs0
      F00 = f00
      F01 = f01
      F10 = f10
      F11 = -f00
      T231 = Tss1
      T232 = Tss2

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE QfunE(xx, erfcx)

      REAL x, xx, erfcx
      REAL a1, a2, a3, a4, a5, p, pi, t

      pi = 3.141592654
      x  = ABS(xx)
```

```
      a1 =  0.319381530
      a2 = -0.356563782
      a3 =  1.781477937
      a4 = -1.821255978
      a5 =  1.330274429

      p = 0.2316419
      t = 1.0/(1.0 + p*x)

      s1 = a1*t + a2*t**2 + a3*t**3 + a4*t**4 + a5*t**5
      s2 = s1*EXP(-(x**2)/2.0)

      IF (xx .GE. 0.0) THEN
       erfcx = s2/SQRT(2.0*pi)
      ELSE IF (xx .LT. 0.0) THEN
       erfcx = 1.0 - s2/SQRT(2.0*pi)
      END IF

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      SUBROUTINE FINDMIN(array,isize,minindex)

      REAL array(isize)
      INTEGER minindex

      int = 1
 11   CONTINUE
      DO 10 i = int+1, isize
       IF (array(int) .GT. array(i)) THEN
        minindex = i
        int = minindex
        GOTO 11
       END IF
 10   CONTINUE

      RETURN
      END

c%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

References

[1] R. R. Tenney and N. R. Sandell, Jr., "Detection with Distributed Sensors," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-17, No. 4, July 1981.

[2] Z. Chair and P. K. Varshney, "Optimal Data Fusion in Multiple Sensor Detector System," *IEEE Transactions on Aerospace and Electronics Systems*, Vol. AES-22, No. 1, January, 1986.

[3] Z. Chair and P. K. Varshney, "Neyman-Pearson Hypothesis Testing in Distributed Networks," *Proceedings of the 26th conference on Decision and Control*, Los Angeles, CA, December 1987.

[4] S. C. A. Thomopoulos, R. Viswanathan, and D. Bougoulias, "Optimal Decision Fusion in Multiple Sensor System," *IEEE Transaction on Aerospace and Electronic Systems*, Vol. AES-23, No. 5, September 1987.

[5] J. D. Papastavrou and M. Athans, "A distributed Hypothesis-Testing Team Decision Problem with Communication Cost," *Proceedings of 25th Conference on Decision and Control*, Athens, Greece, December 1986.

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.