An adaptive fusion model for distributed detection systems with unequiprobable sources

Yuzheng Zhang
*New Jersey Institute of Technology*

Follow this and additional works at: [https://digitalcommons.njit.edu/theses](https://digitalcommons.njit.edu/theses)

Part of the **Electrical and Electronics Commons**

**Recommended Citation**


[https://digitalcommons.njit.edu/theses/1231](https://digitalcommons.njit.edu/theses/1231)

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.
Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation.

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen.
The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.
ABSTRACT

An Adaptive Fusion Model For Distributed Detection Systems With Unequiprobable Sources

by
Yuzheng Zhang

In a traditional communication system, a single sensor such as a radar or a sonar is used to detect targets. Since the reliability of a single sensor is limited, distributed detection systems in which several sensors are employed simultaneously have received increasing attention in recent years. We consider a distributed detection system which consists of a number of independent local detectors and a fusion center. Chair and Varshney have derived an optimal decision rule for fusing decisions based on the Bayesian criterion. To implement such a rule, the probability of detection $P_D$ and the probability of false alarm $P_F$ for each local detector must be known. This thesis introduces an adaptive fusion model using the fusion result as a supervisor to estimate the $P_D$ and $P_F$. The fusion results are classified as “reliable” and “unreliable.” Reliable results will be used as a reference to update the weights in the fusion center. Unreliable results will be discarded. The thesis concludes with simulation results which conform to the analysis.
AN ADAPTIVE FUSION MODEL FOR DISTRIBUTED DETECTION SYSTEMS WITH UNEQUIPROBABLE SOURCES

by

Yuzheng Zhang

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

January 1994
APPROVAL PAGE

AN ADAPTIVE FUSION MODEL FOR DISTRIBUTED DETECTION SYSTEMS WITH UNEQUIPROBABBLE SOURCES

Yuzheng Zhang

Dr. Nirwan Ansari, Thesis Advisor
Associate Professor of
Electrical and Computer Engineering Department
New Jersey Institute of Technology

Dr. Joseph Frank, Committee Member
Associate Professor of
Electrical and Computer Engineering Department
New Jersey Institute of Technology

Dr. Edwin Hou. Committee Member
Assistant Professor of
Electrical and Computer Engineering Department
New Jersey Institute of Technology
BIOGRAPHICAL SKETCH

Author: Yuzheng Zhang

Degree: Master of Science in Electrical Engineering

Date: January 1994

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1994

- Bachelor of Science in Electrical Engineering,
  Shanghai Jiao Tong University, Shanghai, P.R.China, 1984

Major: Electrical Engineering

Presentations and Publications:

This work is dedicated to my family
ACKNOWLEDGMENT

I wish to express my sincere thanks to Dr. Nirwan Ansari for his guidance and care throughout my master program.

I also wish to gratefully acknowledge the assistance and evaluation of this work from Dr. Joseph Frank and Dr. Edwin Hou as members of my thesis committee.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Preliminaries</td>
<td>2</td>
</tr>
<tr>
<td>2 DATA FUSION RULE</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Derivation of The Data Fusion Rule</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Properties of The Fusion Rule</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 Lemma I</td>
<td>7</td>
</tr>
<tr>
<td>2.2.2 Lemma II</td>
<td>9</td>
</tr>
<tr>
<td>3 MODEL ANALYSIS</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Classification of Fusion Results</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1 Lemma III</td>
<td>15</td>
</tr>
<tr>
<td>3.1.2 Lemma IV</td>
<td>17</td>
</tr>
<tr>
<td>3.1.3 Lemma V</td>
<td>17</td>
</tr>
<tr>
<td>3.2 Updating</td>
<td>18</td>
</tr>
<tr>
<td>3.2.1 Lemma VI</td>
<td>20</td>
</tr>
<tr>
<td>4 SIMULATION</td>
<td>22</td>
</tr>
<tr>
<td>4.1 Conditional Probability Mass Function of ( y )</td>
<td>23</td>
</tr>
<tr>
<td>4.2 Convergence of Weights</td>
<td>24</td>
</tr>
<tr>
<td>5 CONCLUSIONS</td>
<td>25</td>
</tr>
<tr>
<td>APPENDIX A SIMULATION PROGRAMS</td>
<td>26</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>34</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>23</td>
</tr>
<tr>
<td>4.3</td>
<td>23</td>
</tr>
<tr>
<td>4.4</td>
<td>24</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Background

There has been a growing interest in developing efficient and reliable distributed detection systems (the multiple sensor systems) for target recognition and communications. There are two major options for data processing in multiple sensor systems. In the first option, complete sensor observations are transmitted to the central processor. This requires a large communication bandwidth. The second option is to have distributed systems. Some or all of the processing results can be done at each local sensor system and then transferred to a fusion center. Tenney and Sandell [1] were one of the first to study the problem of detection with distributed sensors. They applied the classical single sensor detection theory to a two-sensor two-hypothesis test. An optimum local decision rule was established to minimize a global cost. Sadjadi [2] generalized the work of [1] to \( n \) detectors and \( m \) hypotheses, and obtained similar conclusions. Chair and Varshney [3] assumed that each local detector had fixed thresholds and each local decision was independent. With these assumptions, an optimum fusion model was generated.

Optimal techniques have also been developed for other criterions. When \( a \) \( priori \) probabilities were unknown, Thomopoulos [4] used the Neyman-Pearson (NP) test both at the local detector level as well as the decision fusion level. An optimal decision scheme was derived. Demirbas [5] applied the maximum a posterior (MAP) concept for object recognition in multi sensor environment and showed that the maximum a posterior (MAP) estimation approach minimized mean square error estimation.

In the distributed system with a data fusion shown in Figure 1.1, some data processing is done at each sensor and partial results are transmitted to the data fusion center for further processing. The final result are then available at the data
fusion center. In considering the cost, reliability, survivability, and communication bandwidth, the option of distributed processing is more attractive for many applications.

![Figure 1.1 Distributed detection system with a data fusion center](image)

**Figure 1.1** Distributed detection system with a data fusion center

We consider the optimal decision rule which was derived by Chair and Varshney [3]. To implement such rule, the probability of detection $P_D$ and the probability of false alarm $P_F$ for each detector must be known, but this information is not always available in practice. In this thesis, we proposed and developed an adaptive model which is extended from Chair and Varshney's work [3], and uses the fusion results as a supervisor to estimate $P_D$ and $P_F$. The model for equiprobable sources has been presented [6]. This thesis consider the performance and analysis for unequiprobable sources. Various components of the model will be covered: the fusion rule, classification of fusion results, updating algorithm for the fusion center, and computer simulations.

### 1.2 Preliminaries

Let us consider a binary hypothesis testing problem with the following two hypotheses

- $H_0$: signal is absent
- $H_1$: signal is present
The *a priori* probabilities of the two hypotheses are denoted by $P(H_0) = P_0$ and $P(H_1) = P_1$. As shown in Figure 1.1, we assume that there are $n$ detectors, and the observations at each detector are denoted by $x_i, i = 1, ..., n$. We further assume that the observations at the individual detectors are statistically independent and that the conditional probability is denoted by $P(x_i|H_j), i = 1, ..., n, j = 0, 1$. Each detector employs a decision rule $g_i(x_i)$ to make a decision $u_i, i = 1, ..., n$, where

$$u_i = \begin{cases} 
-1, & \text{if } H_0 \text{ is declared} \\
+1, & \text{if } H_1 \text{ is declared}
\end{cases} \quad (1.1)$$

The probabilities of false alarm and miss for each detector are denoted by $P_F$ and $P_M$, respectively.

After processing the observations locally, the decisions $u_i$ are transmitted to the data fusion center. This reduces the communication bandwidth required as compared to what is needed if the complete observations $x_i$ were transmitted. The data fusion center determines the overall decision $u$ for the system based on the individual decisions, i.e.,

$$u = f(u_1, ..., u_n). \quad (1.2)$$

In the next chapter, an optimum data fusion rule for our model will be derived.
CHAPTER 2
DATA FUSION RULE

2.1 Derivation of The Data Fusion Rule

We assume that each local detector has fixed thresholds and the probabilities of false alarm and miss for each detector, \( P_F \) and \( P_M \), are known. Each local decision is also assumed to be independent.

The data fusion problem can be viewed as a binary-hypothesis detection problem with individual detector's decisions being the observations. We consider the Bayes decision criterion which employs a systematic procedure of assigning a cost to each correct and incorrect decision, and then minimizing the total average cost, denoted by \( B \). If we let \( C_{jk} \) be the cost of making decision \( D_j \) when \( H_k \) is true, then for binary decision problem there are four possible costs:

\[
C_{10} = \text{Cost of deciding } D_1 \text{ when } H_0 \text{ is true}
\]

\[
C_{00} = \text{Cost of deciding } D_0 \text{ when } H_0 \text{ is true}
\]

\[
C_{01} = \text{Cost of deciding } D_0 \text{ when } H_1 \text{ is true}
\]

\[
C_{11} = \text{Cost of deciding } D_1 \text{ when } H_1 \text{ is true}
\]

where \( D_1 : u = +1 \), \( D_0 : u = -1 \). The total average cost is:

\[
B = E[C_{jk}]
\]

\[
= C_{10}P[D_1, H_0] + C_{00}P[D_0, H_0] + C_{01}P[D_0, H_1] + C_{11}P[D_1, H_1].
\]

To minimize the cost \( B \), the optimum decision rule is given by the following likelihood ratio test [7] (Bayes decision criterion):

\[
\frac{P(u_1, \ldots, u_n|H_1)}{P(u_1, \ldots, u_n|H_0)} \stackrel{H_1}{\geq} \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})}.
\]

The quantity on the left-hand side is the likelihood ratio, and the Bayes optimum threshold is on the right-hand side. In our model, we use the minimum probability of error criterion, that is, \( C_{00} = C_{11} = 0 \), and \( C_{10} = C_{01} = 1 \). Therefore,

\[
\frac{P(u|H_1)}{P(u|H_0)} \stackrel{H_0}{\geq} \frac{P_0}{P_1}.
\]
where \( \mathbf{u} \) is a vector of observations. Using Bayes rule:

\[
P(x, y) = P(x|y)P(y) = P(y|x)P(x),
\]

to express the conditional probabilities, we have:

\[
\frac{P(\mathbf{u}|H_1)\rho_1}{P(\mathbf{u}|H_0)\rho_0} = \frac{P(\mathbf{u}, H_1)}{P(\mathbf{u}, H_0)} = \frac{P(H_1|\mathbf{u})H_1}{P(H_0|\mathbf{u})H_0} \geq 1. \tag{2.3}
\]

Thus the corresponding log-likelihood ratio test is

\[
\log \frac{P(H_1|\mathbf{u})H_1}{P(H_0|\mathbf{u})H_0} \geq 0 \tag{2.4}
\]

Let \( S_+ \) be the set of all local detectors \( i \) such that the local decision \( u_i = +1 \), and \( S_- \) be the set of all local detectors \( i \) such that the local decision \( u_i = -1 \). Assume that \( u_i \)'s are independent,

\[
P(H_1|\mathbf{u}) = \frac{P_1P(\mathbf{u}|H_1)}{P(\mathbf{u})} = \frac{P_1}{P(\mathbf{u})}\prod_{s_+} P(u_i = +1|H_1) \cdot \prod_{s_-} P(u_i = -1|H_1). \tag{2.5}
\]

Similarly,

\[
P(H_0|\mathbf{u}) = \frac{P_0P(\mathbf{u}|H_0)}{P(\mathbf{u})} = \frac{P_0}{P(\mathbf{u})}\prod_{s_+} P(u_i = +1|H_0) \cdot \prod_{s_-} P(u_i = -1|H_0). \tag{2.6}
\]

Thus, from Eq.(2.5) and Eq.(2.6):

\[
\log \frac{P(H_1|\mathbf{u})}{P(H_0|\mathbf{u})} = \log \frac{P_1}{P_0} + \sum_{s_+} \log \frac{P(u_i = +1|H_1)}{P(u_i = +1|H_0)} + \sum_{s_-} \log \frac{P(u_i = -1|H_1)}{P(u_i = -1|H_0)} \tag{2.7}
\]

Define

\[
w_0 = \log \frac{P_1}{P_0},
\]

\[
w_i = \begin{cases} 
\log \frac{P(u_i = +1|H_1)}{P(u_i = +1|H_0)}, & \text{if } u_i = +1, \\
\log \frac{P(u_i = -1|H_0)}{P(u_i = -1|H_1)}, & \text{if } u_i = -1,
\end{cases} \tag{2.8}
\]
or

\[ w_0 = \log \frac{P_1}{P_0}. \]

\[ w_i = \begin{cases} 
\log \frac{1 - P_{M_i}}{P_F}, & \text{if } u_i = +1, \\
\log \frac{1 - P_{F_i}}{P_M}, & \text{if } u_i = -1. 
\end{cases} \quad (2.9) \]

Eq. (2.7) can be expressed as

\[
\log \frac{P_{|H_1(u)|}}{P_{|H_0(u)|}} = w_0 + \sum_{s_+} w_i - \sum_{s_-} w_i \tag{2.10}
\]

\[ = w_0 + \sum_{i=1}^{n} w_i u_i. \]

Therefore, from Eq. (2.4) and Eq. (2.10), we have the data fusion rule as

\[ u = f(u_1, \ldots, u_n) = \begin{cases} 
1, & \text{if } w_0 + \sum_{i=1}^{n} w_i u_i > 0, \\
-1, & \text{otherwise}. \tag{2.11}
\end{cases} \]

The optimum data fusion rule can be implemented as shown in Figure 2.1.

where

\[ y = w_0 + \sum_{i=1}^{n} w_i u_i. \]

![Figure 2.1 The fusion center structure](image)
2.2 Properties of The Fusion Rule

There are two interesting properties about the fusion rule. One describes the relationship between two conditional probability mass functions: $P(y - w_0 = \zeta|H_1)$ and $P(y - w_0 = \zeta|H_0)$, described as Lemma I. It is very useful to analyze the performance, when we use $y$ (see Figure 2.1) as a supervisor to train each weight in the fusion center. The other describes the conditions, which can make the system approach optimum, described as Lemma II.

2.2.1 Lemma I:

When each weight in the fusion center is optimum (described in section 2.1), the conditional probability mass functions $P(y - w_0 = \zeta|H_1)$ and $P(y - w_0 = \zeta|H_0)$ satisfy the following equation:

$$e^\zeta = \frac{P(y - w_0 = \zeta|H_1)}{P(y - w_0 = \zeta|H_0)},$$

where $\zeta$ is a possible value of $y - w_0$.

Proof:

Consider the structure shown in Figure 2.1. We have:

$$y = w_0 - \sum_{j=1}^{n} w_j u_j$$

or

$$y = w_0 + \sum_{j \in S^+} w_j - \sum_{j \in S^-} w_j.$$  \hspace{1cm} (2.12)

Here $S^+ = \{j : u_j = 1\}$, and $S^- = \{j : u_j = -1\}$. From Eq.(2.8), Eq.(2.13):

$$y = w_0 + \sum_{j \in S^+} \log \frac{P(u_j = 1|H_1)}{P(u_j = 1|H_0)} - \sum_{j \in S^-} \log \frac{P(u_j = -1|H_0)}{P(u_j = -1|H_1)}$$

$$y = w_0 + \log \left[ \prod_{j \in S^+} \frac{P(u_j = 1|H_1)}{P(u_j = 1|H_0)} / \prod_{j \in S^-} \frac{P(u_j = -1|H_0)}{P(u_j = -1|H_1)} \right]$$

or

$$e^{y - w_0} = \frac{\prod_{j \in S^+} P(u_j = 1|H_1) \prod_{j \in S^-} P(u_j = -1|H_1)}{\prod_{j \in S^+} P(u_j = 1|H_0) \prod_{j \in S^-} P(u_j = -1|H_0)}.$$  \hspace{1cm} (2.14)
Let $\zeta$ be a possible value of $y - w_0$ and each local decision $u_j$ is independent,

$$P(y - w_0 = \zeta | H_1) = \sum_{u \in U} P(w^T u = \zeta | H_1),$$

where $u$ is a vector of $u_i, i = 1, 2, \cdots$. $w$ is a vector of $w_i, i = 1, 2, \cdots$, and

$$U = \{u : w^T u = \zeta\}.$$

By defining $S$ as $\{S^+, S^\pm\}$: a combination of $S^+$ and $S^-$ such that

$$\sum_{j \in S^+} w_j - \sum_{j \in S^-} w_j = \zeta,$$

$$P(y - w_0 = \zeta | H_1) = \frac{\sum_s \prod_{j \in S^+} P(u_j = 1 | H_1) \prod_{j \in S^-} P(u_j = -1 | H_1)}{\sum_s \prod_{j \in S^+} P(u_j = 1 | H_0) \prod_{j \in S^-} P(u_j = -1 | H_0)}. \quad (2.15)$$

From Eq. (2.14) and the following equality

$$\frac{a}{b} = \frac{c}{d} = k \quad \Rightarrow \quad \frac{a + c}{b + d} = \frac{bk + dk}{b - d} = k, \quad (b \neq 0, d \neq 0, b + d \neq 0)$$

then

$$\frac{P(y - w_0 = \zeta | H_1)}{P(y - w_0 = \zeta | H_0)} = \frac{\sum_s \prod_{j \in S^+} P(u_j = 1 | H_1) \prod_{j \in S^-} P(u_j = -1 | H_1)}{\sum_s \prod_{j \in S^+} P(u_j = 1 | H_0) \prod_{j \in S^-} P(u_j = -1 | H_0)}$$

$$= \frac{\prod_{j \in S^+} P(u_j = 1 | H_1) \prod_{j \in S^-} P(u_j = -1 | H_1)}{\prod_{j \in S^+} P(u_j = 1 | H_0) \prod_{j \in S^-} P(u_j = -1 | H_0)}$$

$$\frac{P(y - w_0 = \zeta | H_1)}{P(y - w_0 = \zeta | H_0)} = e^{y - w_0} = e^\zeta. \quad (2.16)$$

□
Eq.(2.16) is a very interesting result. The ratio of the conditional probabilities under $H_1$ and $H_0$ only depends on the value $y - w_0$, even the probability mass functions $P(y - w_0 = \zeta | H_1)$ and $P(y - w_0 = \zeta | H_1)$ may not be monotonic with $\zeta$. This is illustrated in Figure 2.2.

![Figure 2.2](image)

**Figure 2.2** Relationship between $P(y - w_0 = \zeta / H_1)$ and $P(y - w_0 = \zeta / H_0)$

### 2.2.2 Lemma II

Let the fused results be classified as reliable and unreliable. Denoting the reliable results by $H_1$ and $H_0$, and the unreliable results by $H_x$:

$$P(H_1) + P(H_0) + P(H_x) = 1.$$  \hspace{1cm} (2.17)

Only reliable results $H_1$ and $H_0$ are used to update each weight, denoted by $\tilde{w}_i$.

Let $\alpha = \frac{P(H_1 | H_0)}{P(H_1 | H_1)}$, $\beta = \frac{P(H_0 | H_1)}{P(H_0 | H_0)}$ and $\gamma = \frac{P(H_1 | H_0)}{P(H_0 | H_0)}$. If $\alpha = 0$, $\beta = 0$ and $\gamma = 1$, then $\epsilon = |w_i - \tilde{w}_i| = 0$. $i = 0, 1, \ldots$, i.e., $\tilde{w}_i$'s are perfect estimation of $w_i$ in this case.

**Proof:**
Let \( r_i = \log w_i \) and \( \hat{r}_i = \log \hat{w}_i \), we have,

\[
\left\{ \begin{array}{l}
\hat{r}_0 = \frac{P_1}{P_0} \\
 r_i = \begin{cases}
\frac{P(u_i = 1|H_1)}{P(u_i = 1|H_0)} & \text{if } u_i = +1 \\
\frac{P(u_i = -1|H_0)}{P(u_i = -1|H_1)} & \text{if } u_i = -1,
\end{cases}
\end{array} \right.
\tag{2.18}
\]

and

\[
\left\{ \begin{array}{l}
\hat{r}_0 = \frac{P(\bar{H}_1)}{P(\bar{H}_0)} \\
 \hat{r}_i = \begin{cases}
\frac{P(u_i = 1|\bar{H}_1)}{P(u_i = 1|\bar{H}_0)} & \text{if } u_i = +1 \\
\frac{P(u_i = -1|\bar{H}_0)}{P(u_i = -1|\bar{H}_1)} & \text{if } u_i = -1.
\end{cases}
\end{array} \right.
\tag{2.19}
\]

Using the total probability theorem \( P(B|A) = P(B|A)P(A) \):

\[
P(u_i = 1|\bar{H}_0) = \frac{P(u_i = 1, \bar{H}_0)}{P(\bar{H}_0)}
\]

\[
= \frac{P(u_i = 1, \bar{H}_0|H_0)P(H_0) + P(u_i = 1, \bar{H}_0|H_1)P(H_1)}{P(\bar{H}_0)}
\]

\[
= \frac{P(u_i = 1|H_0)P(\bar{H}_0|u_i = 1,H_0)P(H_0) + P(u_i = 1|H_1)P(\bar{H}_0|u_i = 1,H_1)P(H_1)}{P(\bar{H}_0)} \cdot P(\bar{H}_1)
\]

Similarly,
Using Eq. (2.19) and the above formulas:

if \( u_i = +1 \)

\[
\tilde{r}_i = \frac{\left[ P(u_i = 1|H_0)P(\bar{H}_1|u_i = 1,H_0)P(H_0) + P(u_i = 1|H_1)P(\bar{H}_1|u_i = 1,H_1)P(H_1) \right]P(\bar{H}_0)}{\left[ P(u_i = 1|H_0)P(\bar{H}_1|u_i = 1,H_0)P(H_0) + P(u_i = 1|H_1)P(\bar{H}_1|u_i = 1,H_1)P(H_1) \right]P(\bar{H}_1)}
\]

\[
= \frac{P(u_i = 1|H_1)P(\bar{H}_1|H_1)P(H_1)P(\bar{H}_0)}{P(u_i = 1|H_0)P(\bar{H}_0)P(H_0)P(\bar{H}_1)} \cdot \frac{P(u_i = 1|H_0)P(\bar{H}_1|H_0)P(H_0)}{P(u_i = 1|H_1)P(\bar{H}_1|H_1)P(H_1)} + 1
\]

\[
= r_i \cdot \frac{P(\bar{H}_1|H_1)}{P(\bar{H}_0|H_0)} \cdot \frac{P(H_1)}{P(H_0)} \cdot \frac{P(\bar{H}_0)}{P(\bar{H}_1)} \cdot \frac{1}{r_i} \frac{P(\bar{H}_1|H_0)}{P(\bar{H}_0|H_0)} + 1
\]

Here,
Thus,

$$P(\tilde{H}_1|H_1) \cdot P(H_1) \cdot P(\tilde{H}_0) = P(\tilde{H}_1, H_1) \cdot P(\tilde{H}_0) = P(H_1|\tilde{H}_1)$$

$$= \frac{P(\tilde{H}_1|H_1)}{P(\tilde{H}_1|H_0) + P(\tilde{H}_1|H_1)} \cdot \frac{P(\tilde{H}_0|H_0) + P(\tilde{H}_0|H_1)}{P(\tilde{H}_0|H_0)} \quad (\text{by Baye's Theorem})$$

$$= \frac{1}{1 + \frac{P(\tilde{H}_0|H_0)}{P(\tilde{H}_0|H_1)} + \left[1 + \frac{P(\tilde{H}_0|H_1)}{P(\tilde{H}_0|H_0)}\right]}.$$ 

Thus,

$$\hat{r}_i = r_i \cdot \frac{1 + \frac{P(\tilde{H}_0|H_1)}{P(\tilde{H}_0|H_0)}}{1 + \frac{P(\tilde{H}_1|H_0)}{P(\tilde{H}_1|H_1)}} \cdot \frac{1 + \frac{P(\tilde{H}_1|H_0)}{P(\tilde{H}_0|H_1)}}{1 + \frac{P(\tilde{H}_1|H_0)}{P(\tilde{H}_1|H_0)}}.$$ 

Let $\alpha = \frac{P(\tilde{H}_1|H_0)}{P(\tilde{H}_1|H_1)}$, and $\beta = \frac{P(\tilde{H}_0|H_1)}{P(\tilde{H}_0|H_0)}$, then

$$\hat{r}_i = r_i \cdot \frac{1 + \beta}{1 + \alpha} \cdot \frac{1 + \alpha}{1 + r_i r_0 \beta}.$$ (2.20)

Using the following equality

$$\frac{1}{1 + x} = 1 - x + x^2 - x^3 + x^4 - \cdots = 1 - x + O(x) \quad (|x| < 1)$$ (2.21)

where $O(x)$ is the higher order terms of $x$.

If $\alpha \ll 1$, and $\beta \ll 1$. we have

$$\hat{r}_i = r_i (1 + \beta)(1 + \frac{\alpha}{r_i r_0})(1 - \alpha + O(\alpha))(1 - r_i r_0 \beta + O(\beta))$$

$$= r_i \cdot \left[1 + \alpha \cdot \left(1/(r_i r_0) - 1\right) + \beta \cdot (1 - r_i r_0) + O[\alpha, \beta]\right],$$ (2.22)

where $O[\alpha, \beta]$ is higher order terms of $\alpha$ and $\beta$ that is close to 0.

Similarly, if $u_i = -1$, 

From Eq. (2.19):
\[
\tilde{r}_i = r_i \cdot \frac{1 + \frac{r_0 P(\tilde{H}_0 | H_1)}{r_i P(\tilde{H}_0 | H_0)}}{1 + \frac{P(\tilde{H}_0 | H_1)}{P(\tilde{H}_0 | H_0)}} \cdot \frac{P(\tilde{H}_1 | H_0)}{P(\tilde{H}_1 | H_1)}
\]

\[
= r_i \left(1 + \frac{r_0}{r_i} \beta \frac{1 + \alpha}{1 + \beta} \right)
\]

\[
= r_i (1 + \frac{r_0}{r_i} \beta) (1 - \frac{r_i}{r_0} \alpha + O[\alpha]) (1 + \alpha) (1 - \beta + O[\beta])
\]

\[
= r_i \left[1 + \beta \cdot (r_0/r_i - 1) + \alpha \cdot (1 - r_i/r_0) + O[\alpha, \beta]\right]. \quad (2.23)
\]

From Eq. (2.19):
\[
\tilde{r}_0 = \frac{P(\tilde{H}_1)}{P(\tilde{H}_0)}
\]

\[
= \frac{P_1}{P_0} \cdot \frac{P(\tilde{H}_1 | H_1)}{P(\tilde{H}_0 | H_0)}
\]

\[
\tilde{r}_0 = r_0 \cdot \gamma \quad \quad (2.24)
\]

where \(\gamma = \frac{P(\tilde{H}_1 | H_1)}{P(\tilde{H}_0 | H_0)}\).

From Eq. (2.22), Eq. (2.23), Eq. (2.24), if \(\alpha = \frac{P(\tilde{H}_1 | H_0)}{P(\tilde{H}_1 | H_1)} \rightarrow 0\), \(\beta = \frac{P(\tilde{H}_0 | H_1)}{P(\tilde{H}_0 | H_0)} \rightarrow 0\) and \(\gamma = \frac{P(\tilde{H}_1 | H_1)}{P(\tilde{H}_0 | H_0)} \rightarrow 1\), then \(\tilde{r}_i \rightarrow r_i\), \(\tilde{w}_i \rightarrow w_i\), for \(i = 0, 1, \cdots\).
CHAPTER 3
MODEL ANALYSIS

3.1 Classification of Fusion Results

Recall the data fusion center structure shown in Figure 2.1. If the reference signals are given, they can be used as a "reference" to train the system such that weights will converge to the optimal values defined by Eq.(2.8). However, in practice such a reference is not readily available and at the same time, the $P_D$ and $P_F$ of a detector may vary with time. Since the fused decisions are usually better than local decisions, they can be considered as the reference. When the $i$th local decision $u_i$ is equal to the fused decision $u$, then $u_i$ is considered to be correct, otherwise, $u_i$ is considered to be incorrect. Since $y = w_0 - \sum_{i=1}^{n} w_i u_i$, the fused decision $u$ has already taken into account the decision of the $i$th detector, $u_i$. If $u$ is used as a reference for $u_i$, a bias is established for $u_i$. Thus, in the proposed system, the decision of the $i$th local detector $u_i$ is arbitrated by the fused decision of all the other $(n-1)$ local detectors. Denote this fused decision as $\widehat{u}_i$, and define

$$y_i = y - w_0 - w_i = \sum_{j\neq i} w_j u_j$$

The decision $\widehat{u}_i$ in the fusion center for updating $\widehat{u}_i$ depends on the value $y$. The value $y_i$ is divided into reliable and unreliable range. We denote the lower and upper
limit of the unreliable range as $\tau_1$ and $\tau_2$ shown in Figure 3.1. We call $\tau_1$ and $\tau_2$ the reliability thresholds. Only the fused decision $\bar{u}_i$ which satisfy $y_i < \tau_1$ or $y_i > \tau_2$ are chosen to adapt the weight $w_i$. These decisions are considered as reliable decisions, defined by $\bar{H}_1$ when $y_i > \tau_2$, and $\bar{H}_0$ when $y_i < \tau_1$. This type of learning belongs to the class of reinforcement learning [8][9][10][11].

3.1.1 Lemma III

$$\alpha = \frac{P(\bar{H}_1|H_0)}{P(\bar{H}_1|H_1)}$$

is monotonically decreasing with $\tau_2$.

**proof:**

$$\alpha = \frac{P(\bar{H}_1,H_0)}{P(\bar{H}_1,H_1)}$$

$$= \frac{P(y_i > \tau_2|H_0)}{P(y_i > \tau_2|H_1)}$$

$$= \frac{\sum_{j=1}^{n} P(y_i = \zeta_j|H_0)}{\sum_{j=1}^{n} P(y_i = \zeta_j|H_1)}.$$

Without lose of generality, assume that $\zeta_1 > \zeta_2 > \cdots > \zeta_n > \tau_2$ are all possible $y_i$. Note that as $\tau_2$ becomes larger, $n$ becomes smaller.

From Eq.(2.16)

$$e^{-\zeta} = \frac{P(y_i = \zeta|H_0)}{P(y_i = \zeta|H_1)},$$

we have,

$$\frac{P(y_i = \zeta_1|H_0)}{P(y_i = \zeta_1|H_1)} < \frac{P(y_i = \zeta_2|H_0)}{P(y_i = \zeta_2|H_1)} < \cdots < \frac{P(y_i = \zeta_n|H_0)}{P(y_i = \zeta_n|H_1)} \quad (3.2)$$

Denote $A_k = \sum_{j=1}^{k} P(y_i = \zeta_j|H_1)$, $B_k = \sum_{j=1}^{k} P(y_i = \zeta_j|H_0)$, and $\alpha_k = \frac{A_k}{B_k}$. The objective is to show that $\alpha_k > \alpha_{k-1}$ for $k = 1, 2, \cdots, n$. First we need to show $\alpha_2 > \alpha_1.$
Using the following inequality and Eq.(3.2),
\[
\frac{X}{Y} < \frac{a}{b} \Rightarrow \frac{X}{Y} < \frac{X + a}{Y + b} < \frac{a}{b} \quad (Y, b > 0),
\] (3.3)
we have:
\[\alpha_2 > \alpha_1.\] (3.4)

Next, we shall show that, if \(\alpha_k > \alpha_{k-1}\), then \(\alpha_{k+1} > \alpha_k\).

Since
\[\alpha_{k-1} < \alpha_k\] (3.5)

\[\frac{A_{k-1}}{B_{k-1}} < \frac{A_k}{B_k}\]
\[\Rightarrow \frac{A_{k-1}}{B_{k-1}} < \frac{A_{k-1} + P(y_i = \zeta_k | H_0)}{B_{k-1} + P(y_i = \zeta_k | H_1)}\] (3.6)

Using inequality Eq.(3.3) again,
\[\frac{A_{k-1}}{B_{k-1}} < \frac{A_{k-1} + P(y_i = \zeta_k | H_0)}{B_{k-1} + P(y_i = \zeta_k | H_1)} < \frac{P(y_i = \zeta_k | H_0)}{P(y_i = \zeta_k | H_1)}\] (3.7)

Applying Eq.(3.2) and Eq.(3.6) yields:
\[\frac{A_{k-1} + P(y_i = \zeta_k | H_0)}{B_{k-1} + P(y_i = \zeta_k | H_1)} < \frac{P(y_i = \zeta_k | H_0)}{P(y_i = \zeta_k | H_1)} < \frac{P(y_i = \zeta_{k+1} | H_0)}{P(y_i = \zeta_{k+1} | H_1)}\]

Using Eq.(3.3)
\[\frac{A_{k-1} + P(y_i = \zeta_k | H_0)}{B_{k-1} + P(y_i = \zeta_k | H_1)} < \frac{A_{k-1} + P(y_i = \zeta_k | H_0) + P(y_i = \zeta_{k+1} | H_0)}{B_{k-1} + P(y_i = \zeta_k | H_1) + P(y_i = \zeta_{k+1} | H_1)}\]
\[\Rightarrow \frac{A_k}{B_k} < \frac{A_{k+1}}{B_{k+1}} \Rightarrow \alpha_k < \alpha_{k+1}\] (3.8)

From Eq.(3.4), Eq.(3.7) and Eq.(3.8), the \(\alpha\) decreases monotonically with \(\tau_2\) increasing.
3.1.2 Lemma IV
\[ \beta = \frac{P(H_0|H_1)}{P(H_0|H_0)} \] is monotonically increasing with \( \tau_1 \).

Proof:
It is similar to the proof of Lemma III.

\[ \square \]

3.1.3 Lemma V
If \( \tau_1 \) decreases and \( \tau_2 \) increases, \( \epsilon = |w_i - \hat{w}_i| \) is decreasing.

Proof:
From Lemma III and Lemma IV as \( \tau_1 \) decreases and \( \tau_2 \) increases, \( \alpha \) and \( \beta \) are decreasing. Thus, From Lemma II, \( \epsilon = |w_i - \hat{w}_i| \) is decreasing.

\[ \square \]
3.2 Updating

The distributed decision system is assumed to have no knowledge of the probability mass functions of the observations. Thus, the estimated probability of detection and false alarm for the $i$th detector $\hat{P}_{D_i}$ and $\hat{P}_{F_i}$ can be approximated by relative frequencies. Let $m$ be the number of $H_1$, $n$ be the number of $H_0$. Let $m_i$ and $n_i$ be, respectively, the number of decisions made by the $i$th detector that conform to and contradict to the reliable fused decisions. Hence $m$, $n$, $m_i$ and $n_i$ can simply be obtained by counting in the simulations. That is,

$$\frac{m}{n} \approx \frac{P(H_1)}{P(H_0)}.$$

Let $m_i$ be the number of decisions made by the $i$th detector that conform to $H_1$ and $n_i$ the number of decisions that contradict to $H_1$. Hence

$$\frac{m_i}{n_i} \approx \begin{cases} \frac{P(u_i = +1, H_1)}{P(u_i = +1, H_0)}, & \text{if } u_i = +1. \\ \frac{P(u_i = +1, H_0)}{P(u_i = +1, H_1)}, & \text{if } u_i = -1. \end{cases}$$

(3.9)

We shall next develop the updating rule for the fusion center. From Eq.(2.19),

$$\hat{w}_0 = \log \frac{P(H_1)}{P(H_0)}.$$

$$\hat{w}_i = \begin{cases} \log \frac{P(u_i = +1, H_1)}{P(u_i = +1, H_0)}, & \text{if } u_i = +1. \\ \log \frac{P(u_i = +1, H_0)}{P(u_i = +1, H_1)}, & \text{if } u_i = -1. \end{cases}$$

Using the Bayes rule $P(x, y) = p(x|y)P(y)$,

$$\hat{w}_i = \begin{cases} \log \frac{P(u_i = +1, H_1)}{P(u_i = +1, H_0)P(H_1)}, & \text{if } u_i = +1. \\ \log \frac{P(u_i = +1, H_0)}{P(u_i = +1, H_1)P(H_0)}, & \text{if } u_i = -1. \end{cases}$$

(3.10)
Applying Eq.(3.9) and Eq.(2.19) yields

\[ \hat{w}_0 \approx \log \frac{m}{n} \]

and

\[ \hat{w}_i \approx \begin{cases} 
\log \frac{m_i}{n_i} - \hat{w}_0, & \text{if } u_i = +1 \\
\log \frac{m_i}{n_i} + \hat{w}_0, & \text{if } u_i = -1 
\end{cases} \quad (3.11) \]

and

\[ m \approx e^{\hat{w}_0 n} \]

\[ m_i \approx -\frac{1}{n_i} \begin{cases} 
e^{\hat{w}_i + \hat{w}_0 n_i}, & \text{if } u_i = +1 \\
\ne^{\hat{w}_i - \hat{w}_0 n_i}, & \text{if } u_i = -1 
\end{cases} \quad (3.12) \]

Talking the partial derivative of Eq. (3.11) with respect to \( m \) and \( n \), respectively.

\[ \frac{\partial \hat{w}_0}{\partial m} \approx \frac{1}{m}, \quad (3.13) \]

\[ \frac{\partial \hat{w}_0}{\partial n} \approx -\frac{1}{n} = -\frac{1}{m} e^{\hat{w}_0}. \]

and

\[ \frac{\partial \hat{w}_i}{\partial m_i} \approx \frac{1}{m_i} \]

\[ \frac{\partial \hat{w}_i}{\partial n_i} \approx \begin{cases} -\frac{1}{m_i} e^{\hat{w}_i + \hat{w}_0}, & \text{when } u = +1 \\
-\frac{1}{m_i} e^{\hat{w}_i - \hat{w}_0}, & \text{when } u = -1 
\end{cases} \quad (3.14) \]

If the current local detector’s decision conforms to the reliable fusion, its weight \( \hat{w}_i \) should be reinforced. In this case,

\[ \Delta \hat{w}_i \approx \frac{1}{m_i} \Delta m_i = \frac{1}{m_i} \]

On other hand, if the current local decision contradicts to the reliable decision, its weight \( \hat{w}_i \) should be reduced. That is,

\[ \Delta \hat{w}_i \approx \begin{cases} -\frac{1}{n_i} \Delta n_i = -\frac{1}{m_i} e^{\hat{w}_i + \hat{w}_0} \Delta n_i = -\frac{1}{m_i} e^{\hat{w}_i + \hat{w}_0}, & \text{if } u = +1 \\
-\frac{1}{n_i} \Delta n_i = -\frac{1}{m_i} e^{\hat{w}_i - \hat{w}_0} \Delta n_i = -\frac{1}{m_i} e^{\hat{w}_i - \hat{w}_0}, & \text{if } u = +1 
\end{cases} \quad (3.15) \]
and

\[
\Delta \hat{w}_0 \approx \begin{cases} \frac{1}{m} \Delta m = \frac{1}{m} & \text{When } H_1 \text{ occurs} \\ -\frac{1}{m} \Delta n = -\frac{1}{m} e^{\hat{w}_0} \Delta n = -\frac{1}{m} e^{\hat{w}_0} & \text{When } H_0 \text{ occurs} \end{cases}
\]

(3.17)

Thus, we obtain the following updating rule:

\[
\hat{w}_i^+ = \hat{w}_i^- + \Delta \hat{w}_i, \ i = 0, 1, 2, \ldots
\]

(3.18)

where \(\hat{w}_i^+\) and \(\hat{w}_i^-\) represent the weight after and before each update. Since the steady state \(\hat{w}_i\)'s are what we are trying to compute, for actual implementation, we use the current estimated weight \(\hat{w}_i^-\) to compute \(\Delta \hat{w}_i\). That is, to update the weights according to Eq.(3.18), \(\Delta \hat{w}_i\) is computed according to the following table:

<table>
<thead>
<tr>
<th>(H_1)</th>
<th>(H_0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u = +1)</td>
<td>(u = -1)</td>
</tr>
<tr>
<td>(\Delta \hat{w}_0)</td>
<td>(\frac{1}{m_i})</td>
</tr>
<tr>
<td>(\Delta \hat{w}_i)</td>
<td>(\frac{1}{m_i})</td>
</tr>
</tbody>
</table>

### 3.2.1 Lemma VI

Using the updating rule according to Eq.(3.18) and the above table, \(\hat{w}_i^-\) will converge to the desired steady state estimate weight \(\hat{w}_i\).

**Proof:**

At steady state,

\[
E[\hat{w}_i^+ - \hat{w}_i^-] = 0
\]

(3.19)

Using the definition \(E[X] = \sum x_i P(x_i)\) and the updating rule according to Eq.(3.18) and table. Eq.(3.19) becomes,

\[
\frac{1}{m_i} P(u = +1, H_1) \frac{1}{m_i} e^{\hat{w}_i^- + \hat{w}_0^-} P(u = +1, H_0) = 0
\]
Using Eq.(3.10) for further simplification yields,

$$\hat{w}_i^- + \hat{w}_0^- = \hat{w}_i + \hat{w}_0. \tag{3.20}$$

Similarly, if $u = -1$, we have,

$$\hat{w}_i^- - \hat{w}_0^- = \hat{w}_i - \hat{w}_0. \tag{3.21}$$

For $i = 0$, the following condition can similarly be obtained at steady state:

$$\frac{1}{m} P(\hat{H}_1) - \frac{1}{m} e^{\hat{w}_0} P(\hat{H}_0) = 0.$$ 

Thus,

$$\hat{w}_0^- = \hat{w}_0. \tag{3.22}$$

Hence, $\hat{w}_i^- \rightarrow \hat{w}_i$, for $i = 0, 1, \ldots$. 

□
CHAPTER 4

SIMULATION

Figure 4.1 shows the simulation set up to validate the proposed adaptive fusion model.

In this simulation, presented here, the source produces binary signal with $P(H_1) = 0.3$ and $P(H_0) = 0.7$, where $H_1 : +1$ and $H_0 : -1$. Eight sensors are used. The probabilities of false alarm and detection $P_F$ and $P_D$ of each sensor are fixed, but not known to the system. The channel is additive Gaussian noise. The Gaussian random variables are generated according to the following transformation.

$$
\begin{align*}
    x &= (-2 \ln r_1)^{1/2} \cos 2\pi r_2 \\
    y &= (-2 \ln r_1)^{1/2} \sin 2\pi r_2
\end{align*}
$$

where $r_1$ and $r_2$ are uniformly distributed on $(0,1]$, and $(x,y)$ becomes a pair of orthogonal normalized Gaussian random variables. The additive Gaussian variable for each sensor is zero-mean with standard deviation ranged from 0.5 to 1.2.
4.1 Conditional Probability Mass Function of $y$

Figure 4.2 shows the histograms of $P(y = \zeta | H_0)$ and $P(y = \zeta | H_1)$ for 8 sensors and 250000 samples. We can see that the waveforms are not monotonic. Figure 4.3

![Figure 4.2 Probability mass functions $P(y/H_1)$ and $P(y/H_0)$](image)

**Figure 4.2** Probability mass functions $P(y/H_1)$ and $P(y/H_0)$

![Figure 4.3 The log-ratio of probability mass functions log $P(y/H_1)/P(y/H_0)$ shows ln $P(y = \zeta | H_1)/P(y = \zeta | H_0)$](image)

**Figure 4.3** The log-ratio of probability mass functions log $P(y/H_1)/P(y/H_0)$ shows ln $P(y = \zeta | H_1)/P(y = \zeta | H_0)$. It is almost a straight line which conforms to Lemma I:

$$e^{\epsilon} = \frac{P(y - w_0 = \zeta | H_1)}{P(y - w_0 = \zeta | H_0)}$$
4.2 Convergence of Weights

Figure 4.4 shows average errors of weights $|w_i - \hat{w}_i|$ for different $\tau$, $\tau = 0$, 0.5, and 0.75. As shown in the figure, the larger the $\tau$, the smaller the error, which agrees with Lemma II. As the number of unreliable samples increases, the training time becomes longer.

![Figure 4.4 The error with different reliability threshold](image-url)
CHAPTER 5
CONCLUSIONS

In the real-world environment, the probability mass functions of the observations at local detectors may not be known and the performance of the local detectors may not be consistent. Under such circumstances, a system which can adapt itself during the decision making process is needed. The major advantage is that the system can still have smaller error and does not need \textit{a priori} knowledge of the probability density functions of the observations. Simulation results conform to our theoretical analysis.
APPENDIX A
SIMULATION PROGRAMS

main.c

#include <stdio.h>
#include <math.h>
#include "main.h"
#include "noise.h"
#include "sensor.h"

double Na[SensorNum], Wo[SensorNum], W1[SensorNum], Wdc;
double Y, Ym1, Ym0, PH1=0.3;
int i, d, D, u[SensorNum];
long t, P1, P11[SensorNum], P10[SensorNum], P01[SensorNum], P00[SensorNum], Pe=0;

w_adj(T)

double T;
{
    int i,j;
    double B, s, y;
    static double delta=0.0001;

    #ifdef SHOW_W_ADJ
    double yi[SensorNum];

    printf("source: H%d\tfusion decision=%d\ty=%4f\ttao=%4f\nu=%4f\n\t", d, D, Y-Wdc, T);
    for(i=0; i<SensorNum; i++)
        printf("%d\tu[i]=", u[i]);
    printf("\nW0:\n%");
    for(i=0; i<SensorNum; i++)

    return 0;
}
printf("%.4f\t", W0[i]);
printf("\nW1:\t")
for(i=0; i<SensorNum; i++)
    printf("%.4f\t", W1[i]);
printf("\n\t");
#endif SHOW_W_ADJ

for(i=0; i<SensorNum; i++){
  if( u[i] == 1 ){
    /*
     y = (Y-Wdc-W1[i])/(Ym1-W1[i]); */
    y = (2.* (Y-Wdc-W1[i])-(Ym1-Ym0-W1[i]))/(Ym1+Ym0-W1[i]);
    if( y > T ){
      W1[i] += delta;
    }else if( y < -T ){
      W1[i] -= exp(W1[i]+Wdc)*delta;
    }
  }
  else if( y < -T ){
    W1[i] -= exp(W1[i] + Wdc)*delta;
  }
}
#endif SHOW_W_ADJ

#endif SHOW_W_ADJ
else {
/*
 y = (Y-Wdc+WO[i])/(Ym0-WO[i]); */
 y = (2.* (Y-Wdc+WO[i])-(Ym1-Ym0-WO[i]))/(Ym1+Ym0-WO[i]);
 if( y < -T ){
   W0[i] += delta;
 }else if( y > T ){
   W0[i] -= exp(W0[i]-Wdc)*delta;
 }
#endif SHOW_W_ADJ
}
#endif SHOW_W_ADJ
else if( y > T ){
    W0[i] -= exp(W0[i]-Wdc)*delta;
#endif SHOW_W_ADJ
}
#endif SHOW_W_ADJ
else
    printf("W1\t");
#endif SHOW_W_ADJ

y = \frac{2 \cdot (Y - Wdc) - (Ym1 - Ym0)}{Ym1 + Ym0};

if (y > T) {
    Wdc += delta;
    y = (2 \cdot (Y - Wdc) - (Ym1 - Ym0)) / (Ym1 + Ym0);
}
else if (y < -T) {
    Wdc -= \exp(Wdc) \cdot delta;
}
else
    Wdc = 0;

void main(argc, argv)
int argc;
char **argv;
{
    double tmp_d, tmp_d1, tao=0;
    int tmp_i;

    #ifdef STATISTICS_OF_Y
    long SY0[101], SY1[101];
    for (i=0; i<101; i++) {
        SY0[i] = 0;
        SY1[i] = 0;
    }
    #endif
if(argc>1) tao = atof(argv[1]);
if(argc>2) PH1 = atof(argv[2]);

/* initialize each weights */
for(i=0; i<SensorNum; i++)
    tmp_d = .4; /* initial probability for every weights */
    Na[i] = (double)i*0.1+.5;
    W1[i] = W0[i] = log((1.0-tmp_d)/tmp_d);
    P11[i] = P10[i] = P01[i] = P00[i] = 0;
}
Wdc = -1;
sensor_init();

printf("RunTime=%ld tao=%lf PH1=%lf\n", (long)RunTime, tao, PH1);

#ifdef CONVERGE
    printf("CONVERGE\n%ld %ld\n", (long)RunTime, (long)SanIntv);
#endif

for(t=1; t<=RunTime; t++)
    if((d = Data(PH1))==1) P1++;

#ifdef DISPLAY_SOURCE
    printf("%d", d);
    if(t%50 == 0) printf("\n");
#endif DISPLAY_SOURCE

#ifdef DEBUG
    printf("d=%d \nu=", d);
#endif DEBUG

    /* local decision */
    for(i=0; i<SensorNum; i++)
        u[i] = sensor( gausNoise(Na[i]) + d*2-1,i);

#ifdef DEBUG
    printf("%d\t", u[i]);
#endif DEBUG

} 

#ifdef DEBUG
    /* diplay weights */
    printf("\nW0=\t");
for(i=0; i<SensorNum; i++)
    printf("%.4f\t", W0[i]);
printf("\nW1=\t");
for(i=0; i<SensorNum; i++)
    printf("%.4f\t", W1[i]);
printf("\n");
#endif DEBUG

/* fusion */
for(i=0, Y=Wdc, Ym0=0, Ym1=0; i<SensorNum; i++)
    Ym1 += W1[i]; Ym0 += W0[i];
    if( u[i] == 1 ) Y += W0[i];
else Y -= W0[i];
}
#endif STATISTICS_OF_Y

tmp_i = ((Y-Wdc)/Ym1+1.)*50;
if( (tmp_i >=0) & (tmp_i<=100) ){
    if( d == 0 ) SY0[tmp_i]++;
else SY1[tmp_i]++;
}
#endif

D = (Y>0)?1:0;
#endif DEBUG
printf("Y=%..4f\t D=%d", Y,D);
#endif DEBUG

w_adj(tao);

#endif DEBUG

/* display weights */
printf("\nW0=\t");
for(i=0; i<SensorNum; i++)
    printf("%.4f\t", W0[i]);
printf("\nW1=\t");
for(i=0; i<SensorNum; i++)
    printf("%.4f\t", W1[i]);
printf("\n\n");
#endif DEBUG

if( D != d ) Pe++;
for(i=0; i<SensorNum; i++)
    if( (u[i] == 1) & (d == 1) ) P11[i]++;
else if((u[i] == 0) & (d == 0) ) POO[i]++;

30
else if((u[i] == 1) && (d == 0)) P10[i]++;
else
   P01[i]++;

#ifdef CONVERGE
if( t%SanIntv == 0){
   printf("t=%ld\nWdc=%.4f\nW0:\t",t,Wdc);
   for(i=0; i<SensorNum; i++)
      printf("%.4f\t",W0[i]);
   printf("\nW1:\t");
   for(i=0; i<SensorNum; i++)
      printf("%.4f\t",W1[i]);
   tmp_d1 = (double)P1/(double)t;
   printf("\n\nWdc=%.4f\nW0:\t",log(tmp_di/(1-tmp_d1)));
   for( i=0; i<SensorNum; i++)
      tmp_d =log((double)P00[i]/(double)P01[i]*tmp_d1/(1-tmp_d1));
   printf("\nW1:\t");
   for( i=0; i<SensorNum; i++)
      tmp_d =log((double)P11[i]/(double)P10[i]*(1-tmp_d1)/tmp_d);
   printf("\n\n");
}
#endif

printf("PE_DATA\n");
printf("Pe=%e\n",(double)Pe/RunTime);

#ifdef STATISTICS_OF_Y
printf("STAT_Y\n");
for(i=0; i<101; i++)
   printf("%e %e %e\n",(float)(i-50)/50,(double)SY0[i]/RunTime,
   (float)(i-50)/50,(double)SY1[i]/RunTime);
#endif

#ifdef IBM_PC
   getch();
#endif

/**********************************************************
* noise.h
**********************************************************/
#include <stdlib.h>
#endif _noise
#define _noise

```c
#ifdef IBM_PC
#define RND (((double)rand()+1.)/32768.)
#else
#define RND (((double)rand()+1.)/2147483600.)
#endif

/* Data will generate 1 bit data. P(H1) = prob */
#define Data(prob) ((RND<prob)? 1:0)
double gausNoise(double sigma);/* gausNoise */
#endif _noise

/***********************************************************************/
noise.h
/***********************************************************************/
#include <stdlib.h>
#ifndef _noise
#define _noise
#ifdef IBM_PC
#define RND (((double)rand()+1.)/32768.)
#else
#define RND (((double)rand()+1.)/2147483600.)
#endif
/* Data will generate 1 bit data. P(H1) = prob */
#define Data(prob) ((RND<prob)? 1:0)
double gausNoise(double sigma);/* gausNoise */
#endif _noise

/************************************************************
sensor.c
/************************************************************/
#include "main.h"
double T[SensorNum];

int sensor(double s,int no){
    if( s>T[no] ) return 1;
    else return 0;
}

sensor_init(){
    int i;
    for(i=0; i<SensorNum; i++)
        T[i] = 0;
}
# makefile

CC=gcc
CFLAGS=-g

SRCS= main.c noise.c sensor.c
HDRS= main.h noise.h sensor.h
OBJS= main.o noise.o sensor.o
LIB= -lm

all: main

clean:
rm -f $(TESTS) *.o *.ln *

main: $(OBJS) $(HDRS)
$(CC) $(CFLAGS) -o $@ $(OBJS) $(LIB)

bk:
zip bk *.c *.h makefile

tar:
tar cvf - makefile *.c *.h | compress > bk.tar.Z
REFERENCES


