INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
AN INFORMATION BASED APPROACH
TO ANOMALY DETECTION
IN DYNAMIC SYSTEMS

by
KI-TAE OH

submitted in partial fulfillment of the requirements
for the Degree of Philosophy

Thesis Advisor: Dr. Kenneth A. Loparo

Department of Systems and Controls Engineering
CASE WESTERN RESERVE UNIVERSITY
January, 1995
We hereby approve the thesis of

Ki-Tae Oh

candidate for the Ph. D.
degree.*

(signed) Kenneth A. Goward
(chair)

(date 12/12/94)

*We also certify that written approval has been
obtained for any proprietary material contained
therein.
AN INFORMATION BASED APPROACH TO
ANOMALY DETECTION IN DYNAMIC SYSTEMS

Abstract

by

KI-TAE OH

An ultimate goal in dynamic system control is the development of a failure
tolerant control system. Such systems require an anomaly detection mechanism and
a reconfigurable controller. In this dissertation, a new anomaly detection method is
developed for high speed detection. The problem is formulated mathematically in the
observation space, and a hypothesized input-output model is assumed and associated
with each anomaly class. The detection method developed is based on statistical
information theory. The overall performance depends on the speed at which distin-
guishable features in the information are accumulating. Posterior distributions are
calculated and used for the decision procedure. The speed of detection can be in-
creased by effectively using the input signal to probe the system. This probing signal
is synthesized in the time domain in feedback form. The main idea is to maximize
the relative entropy between future output distributions of the two most plausible
models. Modeling uncertainty is considered in the detection and in the synthesis of
the probing signal.

These detection mechanisms are extended to a Detection Network to handle
a Nonlinear Non-Gaussian model. This is important for application purposes, because
some anomaly classes may not be mathematically described using a Linear Gaussian
model. In this Detection Network, a prediction module and a Probability Density
Function (p.d.f.) module are used to obtain the necessary posterior distribution.
and each module is built by using the Gaussian Basis Function Network (GBFN). In the p.d.f. module, the Maximum Likelihood estimation method incorporating Expectation Maximization algorithms is used for training, and various low complexity models in GBN are considered. The Minimum Description Length Principle is applied to the p.d.f. model selection procedure to obtain a good compromise between simplicity and likelihood.
DEDICATION

To My Parents
ACKNOWLEDGMENTS

It is my pleasure to thank my advisor, Professor Kenneth A. Loparo, for his guidance and encouragement. This topic was given to me by him and the thesis was completed under his patient advice.

I wish to express my appreciation to dissertation committee members. Professor Vira Chankong, Professor Howard Chizeck, and Professor Wojbor Woyczynski for their valuable advice and suggestions.

I would like to thank Dr. X. Feng for his encouragement and suggestions during the early stages, and my colleagues. Dr. K. Praprost, Dr. Fang, Ganga. Yang and Liu for their helpful discussions.

I am also grateful to GoldStar Co., in South Korea for granting me a full scholarship during the past four years. I am thankful to President Hun Jo Lee, Directors Kum Sik Im and Dal Woong Kim and Manager. Beong Sik Kang. I am especially grateful to Vice President, Hwan Duck Yu and Senior Director. Mu Soo No for their encouragement. Particularly I am thankful Laboratory Chief Tae Hwan Yun for his kind and gracious support.

I would like to thank Case Western Reserve University friends, Dr. Sun Tae Hwang, Dr. Suck Chan Kim and Dr. Lee Jung Hyun for their personal help in my early school days, and Dr. Kwang Hun Park and Dr. Dukki Jung for their helpful discussions.

It is my pleasure to thank my tutor, Patricia Morrison, for helping me to understand American customs and the art of communication.

I also thank my old friends Dr. Sung Jin Kim and Dr. Eoi Suck Kim for their valuable academic discussion.

I am forever grateful to my father and my mother for their deep and dedicated love.

I sincerely thank my wife, Soo Jung Park, for her encouragement, patience and love.
Contents

List of Figures viii

1 Introduction 1
   1.1 An Overview of Signal Processing for Dynamic Anomaly Detection 1
   1.2 Study Motivations and Thesis Summary 5

2 An Overview of Statistical Failure Detection and Identification 15
   2.1 Stochastic Dynamic Processes 15
   2.2 Multiple Model Adaptation Detector 18
   2.3 Sequential Probability Ratio Test (SPRT) 19
      2.3.1 Binary SPRT 19
      2.3.2 SPRT in Development 22
      2.3.3 M-ary SPRT and FDI 24
   2.4 Signal Selection for Detection 26

3 A New Formulation 29
   3.1 Input-Output Predictive Model 29
   3.2 Formulation 30

4 Signal Selection 38
   4.1 Binary Detection Case 39
   4.2 Input Signal Selection for Binary Detection of Linear Gaussian Processes 44
   4.3 M-ary detection case 47

5 Change Detection in Dynamic Patterns and Time Estimation 51
   5.1 Change Detection in M-ary Detection 51
   5.2 Generalized Change Detection 57

6 Robust Detection 61
   6.1 Robust Detection in the Presence of Model Uncertainties 61
   6.2 An Example 64
   6.3 Probing Signal Selection in the Presence of Uncertainty 66

vi
List of Figures

2.1 The Multiple Model Adaptation Detector (or Bayesian-Kalman Filter) 28
3.1 A decision boundary ........................................... 37
4.1 A log posterior ratio space. .................................. 50
7.1 Detection Network ............................................. 74
7.2 Generalized Posterior Network .............................. 117
8.1 An example of the detection result when there is no probing signal. 134
8.2 Evolution of the log posterior ratio when there is no probing signal. 135
8.3 An example of the detection when PRBS is applied. ............ 136
8.4 Evolution of the log posterior ratios when PRBS is applied. .... 137
8.5 An example of the detection result when MRES is applied. .... 138
8.6 Evolution of the log posterior ratios when MRES is applied. .... 139
8.7 An example of the detection result when VSMRES is applied. ... 140
8.8 Result of the Curve Fitting by using a GBN. ..................... 141
8.9 Result of the p.d.f. estimation by using a GBN. ................ 142
8.10 Result of the p.d.f. estimation when one Gaussian kernel is used. 143
8.11 Result of the MVC and the detection without the probing signal when the model switched from H1 to H2. ......................... 144
8.12 Result of the MVC and the detection with probing signal when the model switched from H1 to H2. .......................... 145
8.13 Result of the output and the detection when the model switched from H1 to H4. No control and no probing signal was used. .... 146
8.14 Result of the MVC and detection without the probing signal when model switched from H1 to H4. .......................... 147
9.1 Entropy levels and decision boundaries ........................ 157
B.1 Target function and noise-free data. .......................... 173
B.2 Result of the surface fitting when noise-free data was used. 81 kernels were used. ................................................. 174
B.3 Result of the surface fitting when different number of kernels were used. 175
B.4 Noise contaminated training data. ................................. 176
B.5 Results of the surface fitting when noise in contained in the dataset.
49 kernels were used. .................................................. 177
B.6 Result of the surface fitting when noise is contained in the dataset. 178
B.7 Target p.d.f. .......................................................... 179
B.8 Samples and the result of parameter initialization obtained by applying
the k-mean clustering algorithm. ...................................... 180
B.9 Log likelihood evolution during the estimation procedure. and esti-
mated parameters using the GBFN. .................................. 181
B.10 Result of the p.d.f. estimation. .................................... 182
B.11 Error functions in the p.d.f. when different number of kernels were
used. Each estimation was performed using k-mean clustering for ini-
tial parameters and further tuning was done using ML method with
the EM algorithm. ...................................................... 183
Chapter 1

Introduction

1.1 An Overview of Signal Processing for Dynamic Anomaly Detection

Detection is the subject that deals with the processing of signals for the purpose of extracting embedded information. This has many applications in diverse fields. The signal from which the desired information is to be extracted may be voltage, pressure, or data from the Dow-Jones Index. The detection problem can be logically formulated as follows. "Suppose we are observing a signal that is coming from one of M possible (static or dynamic) models corrupted by random noise, then our objective is to decide which one of the models is present". Since a signal may be noisy, this problem can be addressed probabilistically. When it is formulated using one specific probabilistic model for each hypothesis, it is often referred to as the \textit{Information Theoretic} approach in the statistical sense.

In communication systems where data transfer is an important task, data is usually converted to binary form, modulated, and transmitted to destinations through several channels. From a receiver’s point of view, the objective is to determine the correct signal that has been received from the transmitter through the channels by analyzing the observations at every moment.

Radar detection is another example where one transmits a pulsed electro-
magnetic signal and waits for the possible return signal to be reflected from a target. This return signal is often contaminated by noise and must be processed to extract the information as to whether there is or is not an object present.

When a dynamic system is involved, the detection task becomes more difficult since there are more aspects and uncertainties that should be considered. Some examples are channel equalization problems in communication and dynamic anomaly detection problems in control. An important but difficult application area is the anomaly detection (or failure detection or fault detection) in a dynamic system controls. This thesis concentrates on the anomaly detection problem.

A failure is an unexpected change of a characteristic property, and it does not necessarily represent the failing of a physical component (Tzafestas and Watanabe 90). Once a failure occurs, if the system is tolerant, then only a small degradation in the quality of control may occur. However mitigation of the failure event might require a structural modification or intelligent adaptation from the controller. Thus, in order to correctly generate and send the control signal, knowing the occurrence of a fault and knowing the source of the fault (for example, which sensor or actuator failed?) is essential. This is often referred to as Failure Detection and Identification (FDI) in the control field. Once the source failure is identified, isolation of the location of the fault is system dependent. For this reason, it has also been called Fault Detection and Isolation (FDI). It is quite sensible that a good control requires good automatic diagnosis. For a good anomaly detection, the detection should be made as quickly as possible and the possibility of an incorrect detection should be minimal. In an aircraft control system, early FDI has been essential since it provides time for reconfiguration which allows the aircraft to continue to function (possibly with reduced performance) in the presence of a fault. Furthermore, early FDI is essential for safety reasons. Generally when FDIs are applied to problems where security is emphasized (e.g., guidance controls or nuclear system controls), timely detection and isolation are the most important issues.

There are several factors that should be considered particularly in the dynamic anomaly detection problem. For example, if the number of hypotheses is more than two, an $M$-ary detection structure is needed. The decision should be made in
variable time rather than fixed time, but this is complicated by the fact that the time the fault occurred is usually unknown and therefore must be estimated. The data samples collected are not independent, and because they are temporally correlated, dynamic models are needed to extract the feature vector. The mathematical models usually do not accurately represent the given class of dynamic behavior, thus there are uncertainties.

For these reasons, it is impossible to make a detector that can treat all kinds of situations. A good policy is to make a detector that is problem dependent. Using a linear model is a good example. Under such a circumstance, simplified detectors have shown reasonable performance.

There have been many methods in FDI for dynamic systems. Roughly speaking, the structure of the FDI consists of two stages: feature generation and decision making. In FDI, a feature refers to the processed measurements from which the desired information can be obtained. Feature vectors can be generated in a number of ways. Some significant methods are:

1. The Detection Filter method (or failure sensitive filter method), where the filter gain is calculated so that the effect of each failure corresponds to a pre-designed fixed direction in residual¹ space (Willsky and Jones 76)

2. The innovation using A Linear Filter (or Kalman Filter) where the innovation process is generated from a linear filter and analyzed stochastically (Mehra and Peschon 71, Willsky and Jones 74, 76)

3. The innovations using A Bank of Linear Filters (or a Bank of Kalman Filters) where each innovation is generated from each Kalman Filter which is designed to fit each fault model

4. The Dedicated Observer method where detection of a failure in a given sensor is accomplished by comparing the each of the estimated state which is calculated from each dedicated observer (Clark 75)

5. The Unknown Input Observer method where the filter gain is calculated so that the effect of each failure corresponds to a pre-designed fixed direction on which

¹When we say ‘residual’ signal, it usually means the state (or output) estimation error.
the disturbances or uncertainties have no effect²

(6) The Parity Check method where temporal consistencies (i.e. parities) are evaluated and checked by using input-output of the process (Chow and Willsky 84)

(7) The Eigenstructure Assignment method which is similar to the Unknown Input Observer method, but the decoupling procedure is performed between the output estimation error and the disturbances

(8) The Identification method where model parameters are estimated and classified (Isermann 84). The common ideas behind these approaches are to construct each model that can represent the each anomaly’s dynamic behavior, and to generate the feature vectors that can represent the consistency (or inconsistencies) between the established model and the measurements of the actual process.

Also, there are a number of procedures for decision making:

(1) The Clustering Method where static feature vectors are classified on the feature space (Zolghadri et al. 93; Sorsa and Koivo 93)

(2) The Generalized Likelihood Ratio method where the likelihood of each anomaly event is calculated by correlating the innovation with each prescribed fault signature (Willsky and Jones 74. 76)

(3) The Multiple Model Adaptation Detector method (or Multiple Model Adaptation Filter method) where posterior probabilities are calculated and the decision is made based on these posterior probabilities

(4) The Sequential Probability Ratio Test method where each ratio of log-likelihood functions is evaluated for each pair of hypotheses and threshold tests are performed (Basseville 88, Zhang 91)

(5) The Statistical property Test method where whiteness of the residual sequence is checked (Mehra and Peschon 71)

(6) The Voting method where a decision is made based on hardware redundancy. There are some survey papers: Willsky 76; Isermann 84; Basseville 88. Frank and Seliger 91; Tzafestas and Watanabe.

²This can be performed by a decoupling procedure
1.2 Study Motivations and Thesis Summary

FDI has been studied for more than twenty years. Most of the studies have been performed concerning the formulation of the detection problem. Although various formulations have been introduced the basic trend is to use the model based approach. In this model based approach, a hypothesized model is assumed and associated with each anomaly class. With this approach, most of the detection methods extract valuable features by effectively processing the observations from the system and then determine which model is consistent with the observations. Hence it is a matter of course that each model should accurately represent the class of dynamics.

Linear models enable tractable mathematical manipulation, but they are applicable only as long as such models can well represent the given anomaly classes. On the other hand, if accuracy is emphasized requiring a complex non-linear model, then mathematically it is intractable for the detection problem even if the non-linear model can well represent the given class of dynamics.

Therefore if each anomaly model can be represented by linear models, then various formulations about the detection problem will be possible. Under these formulations, quantities that do not fit well to the linear system structure are included in the ubiquitous uncertainties. To reflect these uncertainties in the mathematical linear model, terms of parametric perturbations with known bounds or terms of additive noise with known statistical properties have been considered. Hence when a linear model is used, the uncertainty terms are comprised of many factors, e.g., modeling error, measurement noise, disturbances, unmodeled dynamics, and uncertainties caused by the change of surroundings. If these uncertainties are large then the model can not well represent the given dynamic behavior and therefore there will be a gap from their original aim, i.e., model based approach. In the control problem, the objective is to control the system in the presence of uncertainties. In the detection problem, these uncertainties degrade performance.

If the non-linear representation is used then the modeling errors can be reduced and hence the uncertainties. In effect, in a real system, it may be impossible to treat all faults in the system as linear Gaussian systems. Fault dynamics can be
highly non-linear, and the noise distributions can be far from Gaussian. Conventional methods do not seem to fit such a general situation well. That is, the conventional methods may not work effectively. However, only a few studies have addressed FDI for non-linear systems.

The overall efficiency and performance of the detector will be problem dependent. For example, if the output dimension is large and the noise is not a serious problem, then deterministic approaches will be good in the sense of efficiency because they lead to less computation without seriously degrading the performance. However if the output dimension is small and the noise is large then a stochastic approach will be preferred. If given classes of dynamics can be well discriminated from each other by using each linearized model (possibly with small amount of uncertainties) then linear approaches will be efficient. However, if our models are highly non-linear and the detection requires taking advantage of the non-linear difference between each dynamics of the anomaly then non-linear models will be more desirable. Hence the model based approach requires how efficient the model is in respect to the given class of the detection problem which is problem dependent.

This thesis aims at the class of detection problems which is able to work in difficult conditions but requires fast detection. To be more specific, we will concern the class of dynamics where some faults are highly non-linear, noise is seriously considered, and output dimension is small. Moreover we intend to formulate a detector problem such that it is compatible with the linear Gaussian models (possibly with some terms denoting state dependent uncertainties), because part of the anomalies can be well represented by using linear Gaussian description rather than non-linear non-Gaussian description.

The first step to FDI synthesis toward our intention is the use of the observation space. Feature vectors are output estimation errors rather than state estimation errors or parameters, and detection is performed over the observation space rather than over state space or over parameter space. A statistical hypothesis method is applied for detection in the observation space. The data processing inequality (Thomas.
91) shows that no clever manipulation of the data can improve the inference that can be obtained from the data.

Based on the above observations, Willsky (76), Basseville (88) and Zhang(91) seem to be close to our intentions. Willsky has studied the hypothesis method for FDI problems, where each posterior probability for each hypothesis is computed by using a bank of Kalman filters and a detection alarm is made when a posterior probability goes higher than a prescribed threshold. However, in his formulation relations for the probabilities of incorrect decisions are not involved. Basseville's method can only be applied to the binary switching detection problem. Zhang's method is applicable to the $M$-ary detection problem, but it needs $M(M - 1)/2$ comparisons to make a decision. For these reasons, we will formulate a different detection mechanism where decision conditions are adjustable by the probabilities of incorrect detection. This mechanism has the Bayesian property, is applicable to the $M$-ary detection problem, and needs only $M - 1$ comparisons to make a decision. It requires the computation of each posterior probability and a decision is made based on predetermined probabilities of incorrect detection. To be more specific, the detector alerts at the time the strongest hypothesis suppresses all the other hypotheses with a certain amount of information. The overall performance of the detection system depends on the speed at which information that is capable of distinguishing the failure modes is accumulating. The minimum time detection property, in the sense of Bayesian risk, could not be proven yet and is left as an open problem. This will be discussed in Chapter 3.

In most cases, if detection starts then the first detected model will be a normal model. In this case, rather than terminate the detection, we want to continue the test so that abnormal behaviors can be identified if they occur. This requires the design of the detection mechanism that can systematically detect the change from a known hypothesis to an unknown hypothesis. Since we do not know when the model switches, it is a difficult problem. To accomplish this need the Bayesian detection can be continued. In that case, lower bounds will be set to the posterior probabilities to avoid the excessive time delay in the change detection stage. This will insure
that when a model switches the posterior probabilities can start their changes from reasonable initial values. However, this leads to a problem in designing the detection procedure; there is a trade-off between the probability of an incorrect detection at the beginning stage and that of a change detection in the later stages of the detection process. This often leads to the following dilemma: if thresholds are designed to fit the probability of error at the beginning stage, then the detection would be prolonged in the switching detection stage. On the other hand, if we design thresholds to fit to the probability of error at the change detection stage, then there will be high rates of false alarms in the beginning stages. This undesirable behavior can be more or less overcome by estimating the source hypothesis and the destination hypothesis (i.e. by monitoring from where to where the change occurs), and estimating the time. Estimating the time of failure can be helpful for control purposes by supplying important information regarding the changes in the dynamic behavior. For the binary detection case, this problem has been partially answered (Bansal and Papantoni-Kazakos 86). This thesis generalizes this change detection problem to $M$-ary detection case. Application to $M$-ary detection problems is not an easy task, because the simplifications used in the binary detection case are not applicable. For this reason, rather than give a mathematical justification, a heuristic method will be studied in Chapter 5. One disadvantage of the proposed method is that it requires a large amount of computation. More studies should be performed on change detection problem in the $M$-ary detections.

Often linear models with additive uncertainty terms are preferred if the uncertainties are not large. The reason for this is that the linear equations are beneficial in processing the data and hypotheses because of their simplicity. For example, in the filtering problem for a linear Gaussian system, the conditional density is Gaussian which requires a simple computation of the variance and mean$^3$. How well a linearized model represents the underlying dynamic behavior of a real system is problem dependent. To model the uncertainties we have included the additive noise term (in

$^3$The Gaussian p.d.f. is uniquely determined by the mean values and covariances.
most cases, an i.i.d. process), however there are often state dependent uncertainties (so the uncertainty is not i.i.d.). The uncertainties that depend on the state should not be grouped together with the additive i.i.d. noise process. These (parametrically structured) uncertainties will be modeled by Gaussian random variables. The advantage of such an approach is that the noise processes depend explicitly on the variance parameters, and this automatically gives different contributions to the net information at each time step. If the corresponding parameters of the state dependent uncertainties can be identified, then the proposed approach gives even better detection performance. This notion will be explained in Chapter 6.

The overall performance of the detection system depends on the speed at which information that is capable of distinguishing the failure modes is growing. The rate at which information is growing can be increased if we effectively select an input signal for the system. This ‘probing’ signal, which is entirely generated for the detection purpose, should be consistent with the controller’s purpose and should be easy to compute. To insure that the probing does not overwhelm the regulatory input, there should be a (time varying) bound for the magnitude of the probing input. The synthesis of the probing signal depends on how one chooses a performance functional that is maximized over the bounded input space. The design problem for the probing signal should be decoupled from the control problem. This is important for simplicity.

Usually the allowable power of the probing signal will be small. However the amount of the information that is delivered by this small signal can be great if we properly select the signal’s direction. Zhang (89) studied the probing signal generation where the signal is obtained by maximizing the linear combination of relative entropies between the estimated future distribution of the normal model and that of each anomaly model. However as the number of hypotheses increases, the average ‘distance’ between each of the distributions of the model decreases, and the critical information to reach the detection is not efficiently captured by maximizing the linear combination of the performance function. As the number of hypothesized models increases, the probing signal’s responsibility increases. However, since there are constraints on the signals, it is difficult to select which signal is appropriate
for probing needs in each of the hypothesized models. For example, if there is one hypothesized normal model and five different anomaly models, then there will be \( \binom{1+5}{2} = 15 \) different conditional relative entropies. We do not want the probing signals to be worked to distinguish between two or more impossible models. The critical information that is necessary for efficient detection depends on the operating situation, and only the necessary information is required for fast detection.

The proposed idea for selecting the probing signal in an \( M \)-ary detection problem is to find the signal that maximizes the conditional relative entropy between the future output distribution of the two most plausible models. In this way the entire energy of the probing signal is dedicated to the discrimination of the two most likely hypothesized models. Computational simulations show that the probing signal developed in this work is far superior to those using Zhang’s method (Zhang 89), with respect to the speed of detection and the probability of an incorrect decision.

In order to obtain the posterior probabilities and compute the likelihood functions, predictive models are required and linear Gaussian models are commonly used. However in real systems, such linear models might not accurately represent the dynamic behavior associated with each failure mode. There is a lack of sufficient mathematical results to deal with nonlinear non-Gaussian type models, thus empirical approaches are required. Mixed structures of models, where some models are of the linear Gaussian type and some are of the nonlinear non-Gaussian type, may be important in practical situations. This can be accomplished by using an information based detection approach, where detections are made probabilistically in the observation space. The detection procedure can be synthesized without changing the basic working mechanism by replacing each linear Gaussian model with an improved model, if necessary. For these reasons, we will incorporate a Detection Network which can accommodate various generalized models including non-linear, non-Gaussian models as well as linear Gaussian models.

For empirical non-linear, non-Gaussian models, we will consider the use of Neural Networks (NN). NNs have attracted considerable attention because highly
complex nonlinear functions can be closely approximated by such networks using compositions of smooth functions. NNs provide a means for general functional approximation, by using a distributed information structure with appropriate memories. By using such NNs, we can build prediction models using supervised learning methods (for example a numerical search method with Least Mean Square (LMS) criterion) and p.d.f. models with an unsupervised learning method (for example, numerical search method with Maximum Likelihood criterion). These networks can be classified according to their model structure: parametric or nonparametric. In the parametric model, we define a NN structure with some number of unknown parameters, and the estimation procedure is performed by tuning all of the unknown parameters. This parameter tuning is performed by minimizing some penalty (or maximizing some performance) function. In the nonparametric case, particularly in kernel based structures, we define a kernel function that is applied to each sample, and the results are added together. In the Detection Network, the parametric method will be preferred because the nonparametric method is computationally expensive. In the Detection Network, the real time evaluation of both the predictive model and the p.d.f. model for each hypothesis and the computation of the Jacobian matrix are necessary. Among many parametric NNs, the Gaussian-Basis Function Networks (GBFNs) will be preferred because in such networks each node will generally have compact support for which the output is greater than some small value, say $\epsilon > 0$. This is an important feature since training can be efficiently performed. A small change in a parameter does not cause global changes in the other parameters, which leads to a training procedure that is not as computationally intensive as Multi-Layer Perceptron (MLP). Various GBRFs can be defined by constraining the parameters in the representation. For example in multi-variate case, GBRF can be a linear superposition of the independent ellipses or can be a linear superposition of the dependent circles. This notion can be applied to curve fitting problems or to p.d.f. estimation problems. More details are given in Chapter 7.

In our formulation, the amount of information that can be gained through a single observation depends on the evaluation of the log-likelihood function at that
observation. This is directly affected by the covariance structure of the p.d.f. estimation. A distance between two points over the observation space is determined by the covariance structure of the true probability measure. Thus, a short distance in the Euclidean sense can be a long distance in statistical information sense, and a long distance in the Euclidean sense can be short the in statistical information sense. It is well known that the Kullback distance is maximized when our p.d.f. estimation is maximally close to the true measure. Thus estimating p.d.f.s is very important because it determines the net information that can be obtained from samples. and simple Gaussian assumptions may not be adequate in some cases. In obtaining the p.d.f. estimate by using the GBN, the Maximum Likelihood (ML) estimation will be applied with the Expectation Maximization (EM) algorithm. The ML estimation method minimizes the Kullback distance (or relative entropy) from the true distribution. However, the ML method, if used alone, does not provide a sufficient answer: rather than giving an explicit p.d.f. estimator, simultaneous nonlinear equations need to be solved. An effective method was developed to address this problem: the EM algorithm (Dempster et al., 77). This method was developed for the incomplete data processing problem (for example, the ‘credit-assignment’ problem), and was applied to a wide range of supervised and unsupervised learning problems. Generating an observation can be viewed as a two step procedure. First, one component of the mixture is selected, and then a sample is drawn from the distribution of that component. An indicator variable is introduced to denote a category from which one sample is to be picked up. This indicator variable comprises ‘complete data’ together with the observed data. In this sense the observed data alone can be viewed as incomplete data. The EM algorithm is an approach to iterative computation of the maximum-likelihood estimates when the observed data can be viewed as incomplete data. Each iteration consists of an Expectation (E) step, which finds the expectation of the complete-data log-likelihood given the values of the observed data and the current estimate of the parameters, and a Maximization (M) step, which re-estimates the parameters under the assumption that the distribution found in the E step is correct. It was shown that each iteration improves the true likelihood from the data. Various applications have been studied and most used the Gaussian Mixture (where
each covariance matrix is independent and has full freedom). For example, the
density estimation problem (McLachlan and Basford 88; Nowlan 91) and the supervised
learning problem (Ghahramani and Jordan 94) were studied. This algorithm will be
used for the derivations of the training laws for the various kinds of GBFNs, and this
will be explained in Chapter 7.

Even in the GBFN, there are additional degrees of freedom: the number
of kernels and the type of the base function. These are important because under-
parameterization and over-parameterization can occur when the samples are insuffi-
cient. Because every sample is contaminated with noise, this often results in over-
fitting or under-fitting the data in the training procedure, so the generalization per-
formance of the network is degraded. This lack of ability to generalize is detrimental
to the detection performance. A NN method can be used, but the result may be
worse than that of the simple linear Gaussian model. This shortcoming can possibly
be overcome, if the appropriate parameterization can be found. One method that has
been used is the Cross Validation method. However this yields acceptable results only
when the number of samples is large, and there seems to be no obvious solution when
we have a small number of samples. If this is the case, the linear Gaussian models
are preferred. When a moderate number of samples are available, there are other
approaches. One way is to use the Minimum Description Length (MDL) approach in
the model selection problem and this will be explained in Chapter 7.

In information theory, the MDL principle has been studied as an effective
way of measuring the descriptive complexity. The descriptive complexity means the
length of the minimum required codes such that the observed data sequence is re-
constructable. This principle is used in measuring the (descriptive) complexity of
the candidate model given a data sequence. The MDL criteria is a method of model
selection criteria which leads to a good trade-off between the likelihood of the data
and the model complexity. This can be used in supervised learning problems and
unsupervised learning problems. In this thesis it will be used in the p.d.f. estimation
problem. The MDL approach requires two components: codes for representing the
parameters and codes for representing the data given those parameter codes. The ac-
tual codes are not necessary but the length information is necessary. Maximizing the log-likelihood from the estimation's point of view is equivalent to minimizing the data code length from the coding's point of view. The MDL method also has a Bayesian interpretation which will be explained in Chapter 7. Codes can be effectively generated if the distribution of the object to be coded is known, in the Shannon sense. More accurate information about the distribution leads to less redundant codes. A trade-off problem comes between the parameter distribution and the data distribution because parameters should be truncated when they are coded. To be more specific, if parameters are coarsely truncated, then the codes for parameters decrease, but the codes for data increase. On the other hand if parameters are too accurately specified then the codes for data decrease, but the codes for parameters increase. The MDL principle gives the optimal trade-off in the sense of Shannon by giving an optimal truncation. Together with the Cross Validation method, MDL gives a valuable idea about the appropriate empirical parameterization. The MDL approach will be used in model selection among various GBFNs in Chapter 7.
Chapter 2

An Overview of Statistical Failure Detection and Identification

2.1 Stochastic Dynamic Processes

We consider discrete time systems exclusively. The dynamic behavior of a discrete time deterministic system is usually modeled by a system of recursive equations of the form.

\[ x_{k+1} = f_k(x_k, u_k, w_k) \]
\[ y_k = h_k(x_k, u_k, v_k) \]

where \( x_k \in \mathbb{R}^n \) is the \( n \)-dimensional state vector, \( u_k \in \mathbb{R}^m \) is the \( m \)-dimensional known (or measurable) input vector (or control vector), and \( w_k \in \mathbb{R}^g \) is the \( g \)-dimensional unknown disturbance vector, \( y_k \in \mathbb{R}^p \) is the \( p \)-dimensional output vector (i.e., observation) and \( v_k \in \mathbb{R}^h \) is the \( h \)-dimensional measurement noise vector (i.e., unknown measurement error).

The sequence \( \{w_k\}_{k=0}^{\infty} \) is a stochastic process with known probability law: that is the joint probability distribution of the random variables, \( w_0, w_1, \ldots, w_k \), is known for each \( k \). The \( f_k(\cdot, \cdot, \cdot) : Z^+ \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^g \rightarrow \mathbb{R}^n \) is the one-step state transition map. We assume that \( f_0, f_1, \cdots \) are known. As in the case of the input noise sequence, \( \{v_k\}_{k=0}^{\infty} \) is a stochastic process with known joint probability distribution and \( h_k(\cdot, \cdot, \cdot) : Z^+ \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^h \rightarrow \mathbb{R}^p \) is the known output map.
The basic variables are \{x_0, w_0, w_1, ..., v_0, v_1, ...\}. The value of \(x_{k+1}\) is statistically determined by the conditional distribution of \(x_{k+1}\) given \(x_k, u_k\), namely by

\[
P(x_{k+1} \mid x_k, u_k)
\]

If the basic variables \{x_0, w_0, w_1, ..., v_0, v_1, ...\} are all independent then the process can be simplified to a Markov Process for any possible feedback law \(u_k = g(y^k)\). That is

\[
P(x_{k+1} \mid x^k, u^k) = P(x_{k+1} \mid x_k, ..., x_0, u_k, ..., u_0) = P(x_{k+1} \mid x_k, u_k)
\]

Note a stochastic system can be defined by specifying the one-step transition map \(f_k\), the observation map \(h_k\) and the probability distributions of the (independent) basic random variables \(x_0, w_0, w_1, \cdots, v_0, v_1, \cdots\).

Special attention is given to the case when the system model equations are linear (i.e. \(f_k\) and \(h_k\) are linear functions) and when all the basic variables are Gaussian (i.e the input disturbances \(w_k\), measuring error \(v_k\) and initial state \(x_0\) are Gaussian RVs). In this case, recursive equations can be easily derived for the estimation of the state (i.e filtering) or the estimation of the output (i.e prediction) because the state \(x_k\) is also a Gaussian RV. It is of particular importance that for a Gaussian RV, the distribution is determined by specifying the mean and the covariance matrix.

Consider the linear Gaussian stochastic system

\[
x_{k+1} = A_k x_k + B_k u_k + G_k w_k + g_k
\]

\[
y_k = C_k x_k + H_k v_k + b_k
\]

where \(x_k \in \mathbb{R}^m, u_k \in \mathbb{R}^m, y_k \in \mathbb{R}^p, w_k \in \mathbb{R}^q, v_k \in \mathbb{R}^h\) and \(A_k, B_k, G_k, C_k, H_k\) are matrices of appropriate dimension which is possibly time varying. All independent basic random variables, \(x_0, w_0, w_1 \cdots v_0, v_1, \cdots\), are assumed to be Gaussian:

\[
x_0 \sim N(\bar{x}_0, \Sigma_0), \ w_k \sim N(0, Q_k), \ v_k \sim N(0, R_k)
\]

\(b_k\) and \(g_k\) are known deterministic functions of time, which could correspond to biases in the state and output equation. This state system generates a Gauss-Markov
Process, and the residual process can be generated. This residual process is used as a special feature vector because it is sensitive to each of the specific failure. Define the residual as follows

\[ \nu_{k+1} = y_{k+1} - y_{k+1|k} \]
\[ = y_{k+1} - Cx_{k+1|k} - b_k \]

where \( x_{k+1|k} \) is the one step ahead prediction of the state \( x_k \) given data up to time \( k \). Thus the residual process represents the prediction error. The following Kalman Filter equations can be used to compute \( x_{k+1|k} \) and the state estimate \( x_{k+1|k+1} \).

\[ x_{k+1|k} = A_k x_{k|k} + B_k u_k + g_k \]
\[ x_{k+1|k+1} = x_{k+1|k} + K_{k+1} \cdot \nu_{k+1} \]

The Kalman filter gain matrices \( K_{k+1} \) are calculated from the following set of equations:

\[ V_k = C_k \cdot \Sigma_{k|k-1} \cdot C_k' + H_k \cdot R_k \cdot H_k' \]
\[ K_k = \Sigma_{k|k-1} \cdot C_k \cdot V_k^{-1} \]
\[ \Sigma_{k|k} = (I - K_k C_k) \cdot \Sigma_{k|k-1} \]
\[ \Sigma_{k+1|k} = A_k \cdot \Sigma_{k|k} \cdot A_k' + G_k \cdot Q_k \cdot G_k' \]

with initial values.

\[ \Sigma_{0|0} = (I - K_0 \cdot C_0) \cdot \Sigma_0 \]
\[ K_0 = \Sigma_0 \cdot C_0' \cdot (C_0 \cdot \Sigma_0 \cdot C_0' + H_0 \cdot R_0 \cdot H_0')^{-1} \]

Here, \( \Sigma_{k+1|k} \) denotes the covariance matrix of the prediction error \( x_k - x_{k+1|k} \) and \( \Sigma_{k|k} \) is the covariance matrix of the filtering error \( x_k - x_{k|k} \).

The residual sequence (i.e. the innovation process of the Kalman filter) is white noise sequence and therefore all information about the state process has been extracted from the data. The errors that remain result from the intrinsic uncertainty of the model. If the Kalman Filter was designed using incorrect model matrices, then the residuals will not be white and will include errors due to the fact that the prediction was based on an erroneous model. The statistical feature of the residuals can be used for detection.
Under the basic assumption of the Kalman Filter, $y_{k+1|k}$ is an unbiased estimate of $y_{k+1}$, the innovation process is $y_{k+1} - y_{k+1|k} \sim \nu_k \sim N(0, V_k)$, and therefore $p(y_{k+1} | z^k, u_k) \sim N(C_{k+1} x_{k+1|k} + b_k, V_k)$ where $z^k = \{y^k, u^{k-1}\}$. Therefore the conditional probability density function can be deduced as

$$p(y_{k+1} | z^k, u_k) = (2\pi)^{-m/2} \cdot V_{k+1}^{-1/2} \cdot \exp\{ -1/2 \cdot \nu'_{k+1} V_{k+1}^{-1} \nu_{k+1} \}.$$ 

This conditional probability will be used in the Multiple Model Adaptation Detector in the following section.

### 2.2 Multiple Model Adaptation Detector

Consider the situation where we observe the inputs $u_k$ and outputs $y_k$ of a system which is assumed to one of a given finite set of linear stochastic models, indexed by $i = 1, \ldots, M$:

$$x_{k+1}^{(i)} = A_k^{(i)} x_k + B_k^{(i)} u_k + C_k^{(i)} w_k + g_k^{(i)}$$

$$y_k = C_k^{(i)} x_k^{(i)} + H_k^{(i)} v_k^{(i)} + b_k^{(i)}$$

where $w_k^{(i)}$ and $v_k^{(i)}$ are independent, zero mean Gaussian white noise processes, with

$$x_0^{(i)} \sim N(x_0^{(i)}, \Sigma_0^{(i)}), \quad w_k \sim N(0, Q_k^{(i)}), \quad v_k \sim N(0, R_k^{(i)})$$

The matrices $A_k^{(i)}, B_k^{(i)}, C_k^{(i)}, G_k^{(i)}, H_k^{(i)}, Q_k^{(i)}, R_k^{(i)}$ are deterministic functions and the vectors $b_k^{(i)}, g_k^{(i)}$ are assumed to be known for each $k$. The state vector $x_k^{(i)}$ may be of different dimensions for different values of $i$. This allows the different hypothesized models to represent dynamic systems of different orders, possibly modeling equipment failures or malfunctions in real processes.

Now, let $H^{(i)}$ denote the hypothesis that the real system corresponds to the $i$-th model, and let $p(H^{(i)})$ denote the prior that $H^{(i)}$ is true. Similarly, let $p(H^{(i)} | z^k)$ denote the a posterior probability that $H^{(i)}$ is true based on information available up to time step $k$, where $z^k = \{y^k, u^{k-1}\}$ denote the available information. The posterior probabilities can be obtained by using the Bayes formula as follows.

$$p(H^{(i)} | z^{k+1}) = \frac{\prod_{j=1}^M p(y_{k+1} | H^{(i)}, z^k, u_k) \cdot p(H^{(i)} | z^k)}{\sum_{j=1}^M p(y_{k+1} | H^{(j)}, z^k, u_k) \cdot p(H^{(j)} | z^k)}$$
Here the conditional probability \( P(y_k \mid H^{(i)}, z^{k-1}, u_{k-1}) \) is the quantity that must be computed using the one step prediction density generated by the Kalman filter based on the parameters of the \( i \)-th model. Under the assumption that hypothesis \( H^{(i)} \) is true, because \( y_{k+1|k}^{(i)} \) is an unbiased estimate of \( y_{k+1}^{(i)} \), the innovation process has
\[
y_{k+1} - y_{k+1|k}^{(i)} \sim \nu_k^{(i)} \sim N(0, V_k^{(i)})
\]
and therefore
\[
p(y_{k+1} \mid H^{(i)}, z^k, u_k) \sim N(C_k z_{k+1|k}^{(i)} + b_k^{(i)}, V_k^{(i)})
\]
Therefore the conditional probabilities are obtained as follows.
\[
p(y_{k+1} \mid z^k, u_k, H^{(i)}) = (2\pi)^{-m/2} | V_k^{(i)} |^{-1/2} \exp\{-1/2 \cdot \nu_k^{(i)} V_k^{(i)-1} \nu_k^{(i)}\}
\]
The inputs to this Bayesian-Kalman filtering procedure are the \( y_k, u_k \) and the parameters of the various models, and the outputs are \( p(H^{(i)} \mid z^k) \). To implement this, a bank of \( M \) Kalman Filters consist parallely, where each Kalman Filter is tuned to each of the \( M \) hypothesized model. See Figure 2.1. The outputs of each of the Kalman Filters are the innovation sequences \( \nu_k^{(i)} \), which effectively measure how well each of the filters can track and predict the behavior of the state process from the observed data. If the \( i \)-th model is present, then one step prediction error \( \nu_k^{(i)} \) is a white sequence. However all the other \( \nu_k^{(j)}, j \neq i \) will not be white sequences and will include errors more than \( \nu_k^{(i)} \). Thus rather then producing a failure/no failure type of decision, the MMAD calculates the posterior probability that the actual system is operating in each of the \( M \) possible modes and this posterior probability is obtained by the conditional probability.

## 2.3 Sequential Probability Ratio Test (SPRT)

### 2.3.1 Binary SPRT

The basic SPRT mechanism and the application to the failure detection problem are reviewed in this section. The basic SPRT, developed by Wald, starts by assuming that the observations \( \{y_k\}_{k=0}^{\infty} \) are coming from one of the two hypothesized models. The objective is to determine which model is present, as quickly as possible, based on the observations. The detection problem can be formulated in the context
of the detection for deterministic signals in noise. Let

\[ H_0 : \quad Y_k = S_k^{(0)} + N_k^{(0)} \]
\[ H_1 : \quad Y_k = S_k^{(1)} + N_k^{(1)} \]

Here \( S_k^{(0)} \) and \( S_k^{(1)} \) are known deterministic signals generated from the hypothesized models \( H_0 \) and \( H_1 \) respectively, and \( N_k^{(0)} \) and \( N_k^{(1)} \) are noise processes of the models. Assume \( \{n_0^{(0)}, n_1^{(0)}, \ldots\} \) and \( \{n_0^{(1)}, n_1^{(1)}, \ldots\} \) are i.i.d. process with known p.d.f. \( p_N(n_k \mid H_0) \) and \( p_N(n_k \mid H_1) \), respectively\(^1\). The log-likelihood ratio conditioned on the data up to the time \( k \) is given by

\[ \lambda_k = \prod_{i=1}^{k} \log \frac{p_N(y_i - s_i^{(1)} \mid H_1)}{p_N(y_i - s_i^{(0)} \mid H_0)} \]

If the observation signal is coming from known dynamics, say the predictive model with additive noise \( y_k = f(y^{k-1}, u^{k-1}) + n_k \), then \( p_N(y_i - s_i^{(1)} \mid H_1) = p_Y(y_i \mid y^{i-1}, u^{i-1}, H_1) \). The log-likelihood test based on probability density function of noise process \( N \) would be the same as the log-likelihood test based on the probability density functions of \( Y_k \) conditioned on past information.

\[ \lambda_k = \prod_{i=1}^{k} \log \frac{p_Y(y_i \mid y^{i-1}, u^{i-1}, H_1)}{p_Y(y_i \mid y^{i-1}, u^{i-1}, H_0)} \]

In this form, even if the p.d.f. of the RV \( Y_k \) is not i.i.d., its conditional p.d.f. is i.i.d. Then the SPRT compares the probability ratio \( \lambda_k \) to two constants thresholds, \( \mathcal{T} \) and \( \mathcal{T} \) (normally \( -\infty < \mathcal{T} < 0 < \mathcal{T} < \infty \)) for \( k = 1, 2, \ldots \) followed by the following decision rules.

\[
\begin{cases}
  \text{if } \lambda_k \leq \mathcal{T} & \Rightarrow \text{ stop test and declare } H_0 \\
  \text{elseif } \lambda_k \geq \mathcal{T} & \Rightarrow \text{ stop test and declare } H_1 \\
  \text{otherwise} & \Rightarrow \text{ defer decision}
\end{cases}
\]

The sample size \( N \) is the value of \( k \) at which a decision is made. i.e.

\[ N = \inf \{k : \sum_{i=1}^{k} \lambda_i \notin (\mathcal{T}, \mathcal{T})\} \]

\(^1\)Although the null hypothesis model used in Wald’s work didn’t contain the signal part. the model which includes the signal can be considered because we assume that the signal is known.
so $N$ is a RV. Wald showed that the SPRT terminates with probability 1, i.e., $\Pr(N < \infty) = 1$. Let the probabilities of an incorrect detection be denoted as $\alpha = \Pr(\text{decide } H_1|H_0)$ and $\beta = \Pr(\text{decide } H_0|H_1)$. It has been shown that there is a relationship between the thresholds and probabilities of incorrect detection, that is

$$\bar{T} \leq \log \frac{1-\beta}{\alpha}$$
$$T \geq \log \frac{\beta}{1-\alpha}$$

The proof of the above inequalities does not require the i.i.d. assumption, so they are also valid for non i.i.d. observations. The inequalities arise from the crossing of the thresholds $\bar{T}$ and $T$ when the test terminates. If the mean of the log-likelihood ratios are small compared to the size of the given thresholds, the sample size $N$ is large and the overshoot (i.e. $\lambda_N - \bar{T}$) is small on the average. In this case, the overshoot can be neglected and the inequalities can be replaced with equalities, yielding the following approximations.

$$\bar{T} \approx \log \frac{1-\beta}{\alpha}$$
$$T \approx \log \frac{\beta}{1-\alpha}$$

The thresholds are then computed given the probabilities of incorrect decisions. As we have been seen, the SPRT's basic decision mechanism depends on how accurately the probabilities of incorrect detections are chosen, and has nothing to do with the observations. However the rate at which the $\lambda_k$ approaches the thresholds depends on how the information is accumulating, for example, how far statistically the two models are apart.

Denote the average sample number (ASN) by $E(N \mid H_0)$ when $H_0$ is true and $E(N \mid H_1)$ when $H_1$ is true. Note that, for a nonsequential detector, ASN is fixed. It has shown that, with i.i.d. observations, the SPRT satisfies the optimality property that $E(N \mid H_0)$ and $E(N \mid H_1)$ are minimized among all sequential or nonsequential statistical tests, with finite ASN and with error probabilities no greater than $\alpha$ and $\beta$ under $H_0$ and $H_1$, respectively. The optimality of the SPRT states that for any other
test with $\alpha' \leq \alpha$ and $\beta' \leq \beta$

which implies

$E(N \mid H_0) \leq E(N' \mid H_0)$ and $E(N \mid H_1) \leq E(N' \mid H_1)$

The key assumptions used to prove the optimality by Wald are:

(1) There are two hypotheses and

(2) Samples are drawn from i.i.d process

### 2.3.2 SPRT in Development

Generally the detection problem, with an unknown sample size, has been regarded as being very difficult because the search for an optimal solution usually becomes computationally intractable. Simply stated, it requires a stopping rule and a decision rule. The underlying objective is to minimize the expected number of data points needed for a reliable decision, where the reliability of the decision is determined by the false alarm rate and missed alarm rate, which are specified.

Much effort has been made to overcome the limitation of Wald’s SPRT. Armitage (’50) modified Wald’s SPRT in order to extend the binary case to $M$-ary hypotheses problem. He proposed a sequential decision procedure for choosing one of $M$ hypotheses when the observation number is not fixed. This procedure consists of constructing a number of likelihood ratios for various pairs of hypotheses, and then comparing these ratios against each prescribed threshold. He proposed relationships between the thresholds and the various probabilities of the incorrect decision, but the minimum time detection property of the algorithm has yet to be established.

Liu and Blostein (’92) showed that the binary SPRT has the minimum time detection property when the samples are drawn from an independent but nonstationary random process by varying the thresholds. He gave a procedure to update the
thresholds and found the lower bound of the expected number of samples needed, however this method is available only when the signals are periodic.

Baram (‘77, ‘78) studied a convergence property of the test in the sense of the information metric. He showed that even if the true underlying model does not belong to the set of hypothesized models, the likelihood ratio converges to the model which is closest to the true model in the model set. This is achieved as long as the residual sequences are ergodic and the covariance matrices are finite positive definite. These two conditions are satisfied, for a Gauss-Markov Process, under certain controllability, observability and stability assumptions. The information metric used to show the convergence was

$$E_{true} \log \frac{p(y^k|H_i)}{p(y^k|H_j)}$$

where expectations are with respect to the true model and $i$, $j$ are any index within the model set. Note the Kullback information (i.e relative entropy) does not satisfy the metric requirement because it does not satisfy the required symmetry property.

Despite some of the attractive properties, there has been some problems in the SPRT: (1) although the sample size needed in SPRT is finite with probability 1, a bound in the number of samples has not been found. The SPRT saves samples by making quick decisions when the hypothesis is clear. But on the other hand, if the observed data is not consistent with any of the statistical model, or the models are statistically close to each other, the SPRT may require a large number of samples. (2) The SPRT needs accurate knowledge of the underlying p.d.f. In many cases, a Gaussian assumption is used for simplicity, but it does not yield good performance if the true density is far from the assumed density. (3) The theory regarding the optimality of the SPRT in terms of the minimum time detection has been established only for the binary detection case with independent samples. (4) Accurate knowledge of the signal (or deterministic part) is required in carrying out the test. In practical situations, the actual values of the signal may not be known because an estimated value is used in the design stage or the model employed can not be perfect, yielding unknown perturbations from the nominal behavior.
2.3.3 M-ary SPRT and FDI

In spite of these disadvantages, the SPRT has been used in many applications because of its simplicity and the lack of other alternative methods. Because in most cases, the binary SPRT is not sufficient for FDI, it had to be extended. For example, inspired by Amitage's ('50) work Zhang ('89) studied the $M$-ary SPRT and applied it to the failure detection and identification problem. Note, in the extension of Binary SPRT to $M$-ary problems, the minimum time detection property is not guaranteed. $M$-ary SPRT is briefly reviewed next.

Suppose there are $M$ hypotheses $H_1, H_2, ..., H_M$. Let $I = \{(i,j) \mid 1 \leq i \leq M, 1 \leq j \leq M, i \neq j\}$ and

$$\lambda^{i,j}_k = \log \frac{p(y^k \mid H_i)}{p(y^k \mid H_j)} \quad i,j \in I$$

Then there are $M(M - 1)/2$ log-likelihood ratios $\lambda_k$ for various pairs of hypotheses. Using the equality $\lambda^{i,j}_k = \lambda^{1,1}_k - \lambda^{i,1}_k$, all $M(M - 1)/2$ log-likelihood ratios $\lambda^{i,j}_k$ can be determined once the $M - 1$ independent basic log-likelihood ratios $\{\lambda^{1,1}_k, \lambda^{2,1}_k, ..., \lambda^{M,1}_k\}$ are determined. Given $M(M - 1)$ thresholds $T_{i,j}, (i, j) \in I$, assuming $T_{i,j} = -T_{j,i}$, the test is continued until, for some $i$, an hypothesis $H_i$ is found such that $\lambda^{i,j}_k \geq T_{i,j}$ holds (or equivalently $\lambda^{i,i}_k \leq -T_{i,j}$) for all $j \in \{1, 2, ..., M\}, i \neq j$. If $H_i$ is the true hypothesis, then $\lambda^{i,j}_k$ would increase on average as the test goes on for all $1 \leq l \leq i - 1$, but $\lambda^{i,i}_k$ will decrease towards the negative threshold $-T_{i,j}$ for all $j = i + 1 ... M$.

Suppose $\beta_{i,j}$ is the probability of accepting $H_i$ when $H_j$ is true. The modified $M$-ary SPRT accepts $H_i$ if

$$\log \frac{p(y^k \mid H_i)}{p(y^k \mid H_j)} + \lambda^{i,j}_0 \geq T_{i,j}$$

$$\Leftrightarrow \frac{p(y^k \mid H_i)}{p(y^k \mid H_j)} \geq \exp(T_{i,j} - \lambda^{i,j}_0)$$

holds for $1 \leq j \leq M, j \neq i$ where $\lambda^{i,j}_0$ is the initial value of the log-likelihood ratio based on prior information about $H_i$ and $H_j$. Then the desired threshold becomes

$$T_{i,j} \leq \lambda^{i,j}_0 + \log \frac{\beta_{i,i}}{\beta_{i,j}}$$
If $T_{i,j}$ are sufficiently large, then the probability of making a correct decision is close to one. However, for large thresholds, the time to make a decision increases. Therefore there is a trade-off between the detection time and the probability of making an incorrect decision. A sample test algorithm proposed by Zhang is shown next.

1. Calculate the threshold according to the desired probability of incorrect decisions.
2. At sampling time $k$, take an observation on $y$.
3. For $i = 1$ to $M$, compute $\lambda_k^{i,1}$.
   If $\lambda_k^{i,1} \leq -T_{1,i}$ for all $i = 1, \ldots, M$, then accept $H_1$ and terminate the test.
5. For $i = j + 1$ to $M$, compute $\lambda_k^{i,j}$ and check
   * if $\lambda_k^{i,j} \leq -T_{i,j}$ for all $i = j + 1, \ldots, M$ and $\lambda_k^{i,l} \geq T_{j,l}$ for all $l = j - 1, \ldots, 1$
     then accept $H_j$ and terminate the test.
   * Otherwise
     - if $j < M$ return to 5 with $j = j + 1$
     - otherwise return to 2 with $k = k + 1$

For more details and examples see Zhang('89) and Amitage('50). There are two important observations to note. First, although the $M$-ary SPRT has demonstrated good detection performance when used in failure detection problems, the comparison schemes become complex as the number of hypotheses is increased. In fact, it requires $M(M - 1)/2$ comparisons in order to reach an appropriate decision. Second, as the number of hypotheses increases, the test performance decreases and computational load increases. In other words, the termination conditions are more difficult to satisfy as the number of hypotheses increases. Also the computational load increases as the number of hypotheses increases. These features are quite reasonable because inherently, the SPRT evaluates the most plausible hypothesis against all other possible hypotheses in order to satisfy each of the incorrect alarm rates. This unfavorable fea-
ture can not be overcome even if we employ other statistical methods like the MMAD or the GLR methods. So there is a trade-off between the number of hypotheses and the test performance (i.e. how fast the test is completed). One practical approach is to eliminate fault models that are less likely to happen from the set of all possible faults. and only consider the models which are relatively more likely to happen. But in many cases it will be necessary to consider many fault models, and even if a fewer number of faults is used, there will be a need for quick detection in order to send appropriate control commands in a timely manner to avoid potentially dangerous situations. This generally makes the detection job more difficult.

2.4 Signal Selection for Detection

Based on these observations there is a need to increase the power of the test and to find a way to increase the test performance by using the inputs to the system in an appropriate way. Zhang proposed two methods, introducing an open loop signal synthesized in the frequency domain and a closed loop signal synthesized in the time domain. His work can be summarized as follows.

\[
u_k = |\text{Mag}| \cdot \text{sign}(\sin(freq \cdot k + \phi))
\]

where \(freq = \arg \left(\max_{freq} \sum_{i=2}^{M} w_i \cdot J^{i-1}\right)\)

s.t

\[
w_2 + w_3 + \cdots + w_M = 1
\]

\[
w_i \geq 0 \quad \text{for all } i
\]

\[
J^{i-1} = D(p(y_{k+1}|z^k, u_k, H_i) \parallel p(y_{k+1}|z^k, u_k, H_1))
\]

Here, \(u_k\) is the auxiliary input signal at the \(k\)-th time, \(|\text{Mag}|\) is the maximum allowable magnitude, \(\phi\) is on angle and \(J^{i-1}\) is the performance function in frequency domain, indicating how much the signal with \(freq\) can discriminate the mean values of the future observation between the \(i\)-th and 1st hypothesized model. Here, \(w\) represents a fixed weighting factor, and this algorithm is implemented off-line. For more details refer to Zhang ('89). The other method, which is a closed loop signal design procedure based on a time domain approach, can be summarized as follows.
\[ u_k = \arg \left( \max_{u_k} \sum_{i=2}^{M} p(H_i \mid z^k, u_k) \cdot J^{i,1} \right) \]

where \( J^{i,1} = D(p(y_{k+1} \mid z^k, u_k, H_1) \| p(y_{k+1} \mid z^k, u_k, H_1)) \).

Note in the open loop synthesis, the performance index uses the average conditional relative entropy, but in the closed loop synthesis, the conditional relative entropy is used.

Although some improvements in performance can be obtained, there are some disadvantages. First suppose fault \( i \) occurred. If the auxiliary signal is obtained by maximizing a linear combination of independent basis functions \( J^{2,1}, J^{3,1}, \ldots, J^{M,1} \) instead of general \( J^{i,j} \), it may take a long time to discriminate one model from the other candidates unless \( j = 1 \). SPRT needs to discriminate between the hypotheses and select the strongest hypothesis model against the null hypothesis as well as all other possible hypotheses. If any one of these is not satisfied, i.e. no thresholds are reached, a decision is deferred until all the threshold conditions are met. Generally \( J^{i,j} \) can not be represented with a linear combination of \( \{J^{2,1}, J^{3,1}, \ldots, J^{M,1}\} \) with positive weights. Therefore if the auxiliary input signal is required, then there is a better approach. This will be discussed in Chapter 4.

Although the approach proposed by Zhang initially provided some good results, the methods are appropriate for linear Gaussian system and extension to more general non-linear, non-Gaussian system is not easy. Nonlinear and non-Gaussian systems are more general, have abundant complexities, and can be used to describe complex dynamics and statistical behaviors.
Figure 2.1: The Multiple Model Adaptation Detector (or Bayesian-Kalman Filter)
Chapter 3

A New Formulation

3.1 Input-Output Predictive Model

Consider the following input-output predictive model

\[ y_{k+1} = f_{k+1}(y^k, u^k) + n_{k+1} \]

In this equation, \( y^k \) and \( u^k \) denote \( \{y_0, y_1, \ldots y_k\} \) and \( \{u_0, u_1, \ldots u_k\} \), respectively, where \( y_k \in \mathbb{R}^p \) is the output vector, \( u_k \in \mathbb{R}^m \) is the input vector, \( n_k \in \mathbb{R}^p \) is the error or noise vector and \( f_k(\cdot, \cdot) : \mathbb{Z}^+ \times \mathbb{R}^{k+p} \times \mathbb{R}^{k+m} \rightarrow \mathbb{R}^p, k = 0, 1, \ldots \) is the one step deterministic transition map. The noise process \( \{n_k\}_{k=0}^{\infty} \) is assumed to be i.i.d. with zero mean. Thus the conditional ensemble average is

\[ E(y_{k+1} \mid y^k, u^k) = f_{k+1}(y^k, u^k). \]

Accordingly, \( f_{k+1}(y^k, u^k) \) is the ‘best’ one step ahead predictor in the mean square sense.

Linear input-output maps, e.g., ARMAX or linear state space models, can be converted to this predictive model (See the Appendix A for more details). For nonlinear maps, it might be possible to obtain the desired predictive model from input-output identification procedure rather than from the direct derivation from each nonlinear state space model. Note that the model interchangeability is not guaranteed in non-linear systems. There are important advantages in the usage of the predictive...
model. First, the output prediction is attained directly from the deterministic part of the model, and avoids unnecessary complexity. Second, the prediction can be used as a valuable feature vector in the detection. For example, the models might be composed of different orders of ARMAX models, or different orders of non-linear input-output models. This predictive model together with additive noises will be used in the remainder of this work.

3.2 Formulation

Conventional Bayesian methods have the optimal detection property when the sample size is fixed, but can not determine when to terminate the test when the sample size is a RV. On the other hand, the SPRT provides a framework concerning when to stop the test and which hypothesis to choose, but the optimality is proved only for two memoryless hypotheses case. These two statistical hypothesis test methods have been basis in many modern statistical detection/classification methods. However, the optimal method for the multiple memory hypotheses test with the random sample size has not been addressed in the detection fields.

For example, Willsky’s approach (Willsky 76) was based on the statistical hypothesis test method, where each posterior probability for each hypothesis is computed by using a bank of Kalman filters and a detection alarm is made when a posterior probability goes higher than the prescribed threshold. However, in his formulation, relations for the probabilities of incorrect decisions are not involved, hence the optimality can not be issued. Basseville’s method (Basseville 88) can only be applied to the binary switching detection problem. Zhang’s method (Zhang 91) is applicable to M-ary detection problem, but it needs $M(M - 1)/2$ comparisons to make a decision, and the test optimality was not proved. For these reasons, we will formulate a different detection mechanism where decision conditions are adjustable by the probabilities of incorrect detection.

Suppose there are $M$ hypothesized models representing each anomaly mode of a dynamic system and assume that these hypothesized models can be given as the
following $M$ predictive models.

$$H_1 : y_{k+1}^{(1)} = f_{k+1}^{(1)}(y^k, u^k) + n_{k+1}^{(1)}$$

$$\vdots$$

$$H_M : y_{k+1}^{(M)} = f_{k+1}^{(M)}(y^k, u^k) + n_{k+1}^{(M)}$$

Denote the observation set $\Omega = \mathbb{R}^{p \times k}$, the set of all one sided $\mathbb{R}^p$-valued sequences with increasing dimension. Our objective is to find the optimal partition rule of the observation set $\Omega$ into sets $\{\Omega_0, \Omega_1, \ldots, \Omega_M\}$ such that the followings are hold.

- if $y^k \in \Omega_0$ then defer decision
- if $y^k \in \Omega_1$ then decide $H_1$
  \[ \vdots \]
- if $y^k \in \Omega_M$ then decide $H_M$

Equivalently, our objective is to find the \textit{sequential decision rule}, a pair of sequences $(\phi, \delta)$, where $\phi = \{\phi_k\}_{k=0}^\infty$ is a sequence for the stopping rule $\phi_k : \mathbb{R}^{p \times k} \to [0, 1]$, and $\delta = \{\delta_k\}_{k=0}^\infty$ is a sequence for the terminal decision rule $\delta_k : \mathbb{R}^{p \times k} \to E^M$ for each $k \geq 0$. Here $E^M$ is a set of $M$ orthonormal basis vectors in the $M$-dimensional Euclidean space, that is $E_i \in E^M$, for $i = 1, \ldots, M$.

The sequential decision rule $(\phi, \delta)$ operates as follows: Given an observation sequence $\{y_k\}_{k=1}^\infty$, the rule $(\phi, \delta)$ makes the decision $\delta_N(y^N)$ where $N$ is the stopping time defined by $N = \min\{k \mid \phi_k(y^k) = 1\}$. That is $\phi$ tells us when to stop taking samples and make a decision (i.e., when $\phi_k(y^k) = 0$ we take another sample, and when $\phi_k(y^k) = 1$ we stop collecting samples and make a decision.) In this way the number of samples, $N$, is a RV since it depends on the (random) data sequence. The terminal decision rule $\delta$ tells us what decision to make when we stop sampling. To address the optimal $M$-ary sequential decision rule into the Bayesian framework, we can consider the following extended Bayesian problem by adding an additional cost to the sample size.

$$R_1(\phi, \delta) = C_1 E\{\|\delta_N(y^N) - E_1\|_{L_1} \mid H_1\} + CE\{N \mid H_1\}$$

$$R_2(\phi, \delta) = C_2 E\{\|\delta_N(y^N) - E_2\|_{L_1} \mid H_2\} + CE\{N \mid H_2\}$$

$$\vdots$$

$$R_M(\phi, \delta) = C_M E\{\|\delta_N(y^N) - E_M\|_{L_1} \mid H_M\} + CE\{N \mid H_M\}$$
where $C > 0$ is the cost per samples, $C_1, \ldots, C_M > 0$ are the costs allocated for incorrect decision. $N$ is the stopping time, and $\| \cdot \|_{L_1}$ denotes the $L_1$ norm (or can be others). The total Bayes’ risk is thus given by

$$r(\phi, \delta) = p(H_1)R_1(\phi, \delta) + \cdots + p(H_M)R_M(\phi, \delta)$$

where $p(H_1), \ldots, p(H_M)$ denote prior probabilities. We may follow the Bayesian version of the sequential rule for the $M$-ary memory system detection problem given the probabilities of incorrect decision by minimizing the total Bayes’ risk as follows.

$$\min_{\phi, \delta} r(\phi, \delta)$$

$$s.t \quad \sum_{i=1}^{M} p(H_i) = 1$$

$$\phi_0 = 0$$

Here $\phi_0 = 0$ means that the test does not stop before at least one observation is taken. This problem has been addressed only for the simple binary memoryless system detection with i.i.d. assumptions in the samples. Actually, Bayesian version of the Binary SPRT was first considered by Ferguson(1967) where it was proved that the truncated likelihood ratio test was the solution. For more details refer to Poor(88).

Therefore, instead of trying to solve the complicated $M$-ary optimization problem, we propose a heuristic algorithm based on the following three observations: First, the test structure whose optimality was proven under the i.i.d. assumption can be applied to memory system. In this case the optimal property may not be conserved. However the parameters of the test (i.e. thresholds) do not depend on the i.i.d. assumption. (This will be clear in later.) Second, once the test stops, the hypothesis that is to be selected will be the one which has the largest a posteriori. Otherwise the test can not stop. Hence we can deduce the terminal decision rule. Third, in order to stop the test, the strongest hypothesis should surpass each of the other hypotheses in evidence with certain minimum amounts which can support the given probabilities of an incorrect detection. Based on these observations the following algorithm is proposed.
Algorithm 1  Find at each time step
\[
i^* = \arg \max_{1 \leq i \leq M} \underbrace{\log \frac{p(H_i | y^k, u^{k-1})}{p(H_j | y^k, u^{k-1})}}_{\text{decision rule}}
\]
\[
s.t. \quad \frac{\log \frac{p(H_{i^*} | y^k, u^{k-1})}{p(H_j | y^k, u^{k-1})}}{T_{i^*,j}} \geq 1 \quad \text{for all } 1 \leq j \leq M, \ j \neq i^*
\]

\[
\left\{
\begin{array}{l}
\text{if the solution exists} \\
\quad \text{then declare } H_{i^*} \quad (i.e., \ \phi_k(y^k) = 1, \ \delta_k(y^k) = E_{i^*}) \\
\text{otherwise} \\
\quad \text{defer the decision and continue the test process} \\
\quad \quad (i.e., \ \phi_k(y^k) = 0, \ \delta_k(y^k) = \text{arbitrary})
\end{array}
\right.
\]

where \(T_{i,j} (1 \leq i \leq M, \ 1 \leq j \leq M, \ i \neq j)\) are test parameters.

The test parameters enable the decision condition to be adjustable by the prescribed probabilities of incorrect decision. These parameters can be determined by the following theorem.

Theorem 1  Suppose \(\beta_{i,j}\) is the probability of accepting \(H_i\) when \(H_j\) is true. Then the following inequalities are hold.
\[
\frac{\beta_{i,j}}{\beta_{i,i}} \leq \exp(-T_{i,j} + \frac{p(H_i)}{p(H_j)})
\]

Proof.  Suppose that \(H_i\) is the hypothesis with the largest posterior probability at time step \(k\), then \(\phi_k(y^k) = 1\) if the following condition is satisfied for all \(1 \leq j \leq M, \ j \neq i\).
\[
\log \frac{p(H_i | y^k, u^{k-1})}{p(H_j | y^k, u^{k-1})} \geq T_{i,j}
\]
\[
\begin{align*}
\Rightarrow & \quad \frac{\log \frac{p(y^k | u^{k-1}, H_i)}{p(y^k | u^{k-1}, H_j)}}{\frac{p(y^k | u^{k-1}, H_i)}{p(H_i)}} + \frac{\log \frac{p(H_i)}{p(H_j)}}{\frac{p(H_i)}{p(H_j)}} \geq T_{i,j} \\
\text{(Bayes' Eq)}
\end{align*}
\]

\[
\Rightarrow \quad p(y^k | u^{k-1}, H_i) \geq p(y^k | u^{k-1}, H_j) \cdot \exp(T_{i,j} - \log \frac{p(H_i)}{p(H_j)})
\]
Let $\Omega_0, \Omega_1, \Omega_2, \ldots, \Omega_M$ denote the mutually exclusive decision boundaries over the observation space $y^k$. Therefore

\[
y^k \in \Omega_0 \Rightarrow \phi_k(y^k) = 0 \quad \text{and} \quad \delta_k(y^k) = \text{arbitrary}
y^k \in \Omega_1 \Rightarrow \phi_k(y^k) = 1 \quad \text{and} \quad \delta_k(y^k) = E_1
\vdots
\vdots
y^k \in \Omega_M \Rightarrow \phi_k(y^k) = 1 \quad \text{and} \quad \delta_k(y^k) = E_M
\]

Integrating BHS of the equation 3.2 over the ensemble of $y^k \in \Omega_i$ yields the probability of deciding $H_i$ when $H_j$, $1 \leq j \leq M$, $j \neq i$ is true. That is

\[
\int_{\Omega_i} p(y^k \mid u^{k-1}, H_i)dy^k \geq \int_{\Omega_i} p(y^k \mid u^{k-1}, H_j)dy^k \cdot \exp(T_{i,j} - \log \frac{p(H_i)}{p(H_j)})
\]

\[
= \Pr(\text{stop at } k \text{ and decide } H_i \mid H_i) \geq \Pr(\text{stop at } k \text{ and decide } H_i \mid H_j)
\]

for all $j$, $1 \leq j \leq M$, $j \neq i$. Summing over all $k$, for $1 \leq k \leq \infty$ yields.

\[
\sum_{k=1}^{\infty} \int_{\Omega_i} p(y^k \mid u^{k-1}, H_i)dy^k \geq \sum_{k=1}^{\infty} \int_{\Omega_i} p(y^k \mid u^{k-1}, H_j)dy^k \cdot \exp(T_{i,j} - \log \frac{p(H_i)}{p(H_j)})
\]

\[
= \Pr(\text{decide } H_i \mid H_i) = \beta_{ii} \geq \Pr(\text{decide } H_i \mid H_j) = \frac{\beta_{ij}}{\beta_{ii}}
\]

for all $j$, $1 \leq j \leq M$, $j \neq i$. Therefore.

\[
\Leftrightarrow \beta_{ii} \geq \beta_{ij} \cdot \exp(T_{i,j} - \log \frac{p(H_i)}{p(H_j)})
\]

\[
\Leftrightarrow \frac{\beta_{ij}}{\beta_{ii}} \leq \exp(-T_{i,j} + \log \frac{p(H_i)}{p(H_j)})
\]

for all $1 \leq j \leq M$, $j \neq i$, where $\beta_{ii} = 1 - \sum_{i \neq i} \beta_{ii}$. □

Note that the proof does not require the i.i.d. assumption, so the theorem is valid for dependent and/or non identically distributed observations. The equation 3.1 can be decomposed as follows.

\[
\left( \sum_{i=1}^{k} \log \frac{p(y_i \mid y^{i-1}, u^{i-1}, H_i)}{p(y_i \mid y^{i-1}, u^{i-1}, H_j)} \right) + \log \frac{p(H_i)}{p(H_j)} \geq T_{i,j} \quad (3.3)
\]

Note that the log-likelihood-ratio is a RV and therefore it has a mean and a variance. If the mean and the variance of this RV are small compared to the threshold $T_{i,j}$ for
all $j, 1 \leq j \leq M, j \neq i$, then the required sample size is large and the overshoot (i.e. overshoot over the thresholds) is small on average at the first time the inequalities 3.3 are satisfied. This observation can enable us to use the following approximation about the thresholds $T_{i,j}$ in terms of the given probabilities of incorrect decisions. That is
\[
\frac{\beta_{ij}}{1 - \sum_{l=1,l \neq i}^{M} \beta_{il}} \approx \exp(-T_{i,j} + \log \frac{p(H_i)}{p(H_j)})
\]
\[
\Rightarrow T_{i,j} \approx \log \frac{p(H_i)}{p(H_j)} - \log \frac{\beta_{ij}}{1 - \sum_{l=1,l \neq i}^{M} \beta_{il}}
\]
for all $1 \leq j \leq M, j \neq i$. By this way, given a set of probabilities of making incorrect decisions, the approximated values of the thresholds can be computed. If the $T_{i,j}$ are sufficiently large, then the probability of making a correct decision is close to one. However, for large thresholds, it takes a long time for the procedure to make a decision. Therefore there is a trade-off between the detection time and the probability of making an incorrect decision.

The basic mechanism sets the thresholds of the detector in such a way that, when the cumulative sum reaches those threshold values, each probability of an incorrect detection is smaller than each of the prescribed ones, based on evidence. Note that the thresholds are independent from the observations. But the rate at which the cumulative sum reaches the thresholds depends on how quickly the evidence (or information) is growing with time, in other words, the statistical separation of the two models. It is interesting to know how the information is growing between two hypotheses at each sample. The following lemma shows this more clearly.

**Lemma 1** Let $B$ be Baram's measure of information and $D$ be Kullback's measure of information (or relative entropy). Let $p_{k+1|k}^{(i)} = p(y_{k+1} \mid y^k, u^k, H_i)$. Then the following relation holds.

\[
B(p_{k+1|k}^{(i)} \parallel p_{k+1|k}^{(j)}) = D(p_{k+1|k}^{(true)} \parallel p_{k+1|k}^{(i)}) - D(p_{k+1|k}^{(true)} \parallel p_{k+1|k}^{(j)})
\]
Proof. From the definition,

\[
B(p^{(i)}_{k+1|k} \parallel p^{(j)}_{k+1|k}) = E_{true} \log \frac{p^{(i)}_{k+1|k}}{p^{(j)}_{k+1|k}}
\]

\[
= \int \log \frac{p^{(i)}_{k+1|k}}{p^{(j)}_{k+1|k}} dP_{true}
\]

\[
= \int \log \frac{p^{(true)}_{k+1|k}}{p^{(true)}_{k+1|k}} dP_{true} - \int \log \frac{p^{(true)}_{k+1|k}}{p^{(true)}_{k+1|k}} dP_{true}
\]

\[
= D(p^{(true)}_{k+1|k} \parallel p^{(i)}_{k+1|k}) - D(p^{(true)}_{k+1|k} \parallel p^{(j)}_{k+1|k}) \quad \square
\]

For a related result refer to Cover (91). The above lemma shows that the mean information increment based on observations up to time \( k \) is simply the difference between the two distances: The distance between the \( j \)-th distribution and the \( i \)-th distribution and the distance between the \( i \)-th distribution and the true distribution. Because the true model is not known, we can not estimate the true model's future output. If strictly \( p^{(true)}_{k+1|k} = p^{(i)}_{k+1|k} \) or \( p^{(true)}_{k+1|k} = p^{(j)}_{k+1|k} \) then we get maximum information. Even if it is not so, the information is accumulated based on the difference of each of the relative entropies. and after a certain amount of information is gathered, the prescribed thresholds will be crossed. (i.e. if the true model is closer to the \( i \)-th one, than the \( j \)-th one, then the positive information is accumulated in the mean sense in proportion to the relative difference between those. On the other hand, if the true model is closer to the \( j \)-th one than the \( i \)-th one, then the negative information is gained in the mean sense in proportion to the relative difference between those.) Therefore the proposed detection mechanism systematically works by choosing the largest a posterior and checking whether the prescribed amount of evidence (or information) has accumulated against all of the other candidate hypotheses.

This method is similar to Willsky's Multiple Model Adaptation Detector. However rather than checking whether the largest a posterior is greater than a prescribed threshold, this framework compares the ratio of the largest a posterior to
all other hypotheses and tests whether all of those ratios are greater than the corresponding prescribed thresholds. This enables us to embed the probabilities of making incorrect decisions into the framework, and this is the key idea. Figure 3.1 shows the decision boundaries over the space of log-posterior-ratios when three hypotheses are joined. Here X-axis is $\log\{p(H_2 \mid y^k, u^{k-1})/p(H_1 \mid y^k, u^{k-1})\}$ and Y-axis is $\log\{p(H_3 \mid y^k, u^{k-1})/p(H_1 \mid y^k, u^{k-1})\}$. The minimum time detection property was not proved. No minimum time detection algorithm for the M-ary memory system has been developed to date.

Figure 3.1: A decision boundary
Chapter 4

Signal Selection

In the proposed detection method, the overall performance depends on how each of the binary detection performs. Furthermore, the performance of each binary detection depends on the speed at which the information is accumulating. We can speed up the rate at which information is growing if we select an input signal for the system efficiently.

If there are $M$ hypotheses then, at each sampling time, there will be $M-1$ log-posterior ratios (i.e., the ratio of the largest a posterior to all the other posteriors). The objective is to obtain information such that the $M-1$ ratios can be easily discriminated, but this is not likely to happen without intervention. Therefore it is desirable to reduce the amount of redundant information.

In this chapter a methodology for detection purpose by an auxiliary input will be proposed. Then this methodology will be extended for the case of $M$ models using the notion of trade-off.
4.1 Binary Detection Case

Suppose at time step \( k \), \( H_i \) is the hypothesis with the largest a posterior and we are comparing it to the posterior for the model \( H_j \). If the inequalities.

\[
\log p(H_i | y^k, u^{k-1}) > \log p(H_j | y^k, u^{k-1})
\]

\[
\geq \log \frac{p(y^k | u^{k-1}, H_i)}{p(y^k | u^{k-1}, H_j)} \cdot \frac{p(H_i)}{p(H_j)} > 0
\]

hold then the hypothesis \( H_i \) is preferred more than the hypothesis \( H_j \). Since the prior distribution are not affected by the observations, \( \log P(y^k | u^{k-1}, H_i) \) can be regarded to be a measure of the information in the observations \( y^k \), with knowledge of \( u^{k-1} \) for favoring the hypothesis for model \( H_i \). The difference

\[
\log P(y^k | u^{k-1}, H_i) - \log P(y^k | u^{k-1}, H_j) = \log \frac{P(y^k | u^{k-1}, H_i)}{P(y^k | u^{k-1}, H_j)}
\]

is then a measure of the information in \( y^k \) for favoring \( H_i \) against \( H_j \). Similarly, the difference

\[
\log \frac{P(y^k | u^{k-1}, H_i)}{P(y^k | u^{k-1}, H_j)} - \log \frac{P(y^{k-1} | u^{k-2}, H_i)}{P(y^{k-1} | u^{k-2}, H_j)} = \log \frac{P(y_k | y^{k-1}, u^{k-1}, H_i)}{P(y_k | y^{k-1}, u^{k-1}, H_j)}
\]

between the information favoring \( H_i \) against \( H_j \) in \( y^k \) and the information favoring \( H_i \) against \( H_j \) in \( u^{k-1} \) can be interpreted as a measure of the information favoring \( H_i \) against \( H_j \) in the single observation \( y_k \).

Generally, because the models are not perfect, the amount of information added in favor of \( H_i \) against \( H_j \) by the observation \( y_k \) in the average will follow Baram's mean information:

\[
B_{k+1} = E(\log \frac{P(y_{k+1}| y^k, u^k, H_i)}{P(y_{k+1}| y^k, u^k, H_j)} | y^k, u^k, H_{true})
\]

If the true probability measure is known, then the accumulation speed of information
can be increased by maximizing Baram’s information measure $B_{k+1}$ with respect to the input sequence $\{u_k\}_{k=0}^k = u^k$. Because the underlying probability measures are unknown, the Baram’s information measure cannot be used in synthesizing the input signal. Instead, the Baram’s mean is replaced with the conditional relative entropies. Some possible candidates that can be maximized with respect to input signal $u_k$ are

**Type 1:**  
$$ J_{k+1}^{i,j} = \frac{1}{2} (D_{k+1/k}^{i,j} + D_{k+1/k}^{j,i}) $$

**Type 2:**  
$$ J_{k+1}^{i,j} = p(H_i|y^k, u^{k-1}) D_{k+1/k}^{i,j} + p(H_j|y^k, u^{k-1}) D_{k+1/k}^{j,i} $$

where $D_{k+1/k}^{i,j}$ is the conditional relative entropy defined as

$$ D_{k+1/k}^{i,j} = E(\log \frac{P(y_{k+1}|y^k, u^k, H_i)}{P(y_{k+1}|y^k, u^k, H_j)} | y^k, u^k, H_i) $$

$$ = E(\log \frac{p(y_{k+1}|y^k, u^k, H_i)}{p(y_{k+1}|y^k, u^k, H_j)} | y^k, u^k, H_i) $$

Note that $D_{k+1/k}^{i,j} \neq D_{k+1/k}^{j,i}$. Therefore it is necessary to include both terms in the performance function. Even if the $i$-th a posteriori is greater than the $j$-th a posteriori, it is not certain that the $i$-th hypothesis is the true hypothesis. It is reasonable to wait until the test is terminated. Type 1 is a simple average of the two relative entropies. Type 2 is a weighted average where the weights are the posterior distribution at time $k$. What index should be used will depend on the particular system tested and the designer’s choice. A signal that maximizes the log-likelihood ratio also maximizes the log-posterior ratio, and this can be shown by adding a constant term $\log \frac{P(H_i)}{P(H_j)}$. Hence a signal that maximizes a linear combination of the log likelihood also maximizes a linear combination of the log-posterior ratio.

If we use a predictive model with additive noise then we can represent its dynamic behavior as two parts: a deterministic part and a noise part with zero mean. The evolution and the mean value is determined by the deterministic part while the noise part accounts for the uncertainty. In other words, the mean value of the future distribution is controlled by deterministic part and its other statistical properties (i.e., moments) are controlled by the noise part. We cannot control all the statistics of the prediction because the true distribution of the noise is unknown. But the distance between the mean values of the two densities can be controlled by the current input.
signal. By controlling the mean values of the future output distribution, the rate at which useful information is obtained can be increased or decreased, and this is the basic idea to the use of input signals to enhance the performance of the detection system.

Generally $u_k$ is designed according to some performance criteria, like regulation, but there may be some flexibility in how this signal is implemented. The input signal can be perturbed from its nominal value via the direction which is favorable to the detection purpose, e.g. to separate posterior distribution to enhance the detection. Denote the nominal value of $u_k$ as $\bar{u}_k$ and the perturbation as $\tilde{u}_k$ such that

$$u_k = \bar{u}_k + \tilde{u}_k.$$ 

From a performance perspective, if the perturbation is too large, this may be an unfavorable situation, but if the perturbation is small then it may not seriously deteriorate the quality of the control but may still be beneficial to detection. Recognizing that a good control performance comes only after a correct diagnosis, there is a trade-off between the short term regulation provided by the numerical control and long term benefits of proper diagnosis.

Because large bounds of $\tilde{u}_k$ are unfavorable to the controller and system, $\tilde{u}_k$ should be constrained by either the maximum allowable output perturbation $\Delta y_{\text{max}}$ from the nominal expected output value $\bar{y}_{k+1}$ or by the maximum allowable input perturbation $\Delta u_{\text{max}}$ from the nominal input $\bar{u}_k$, or even by both of them. So we can set the maximum allowable range for the perturbation signal $\tilde{u}_k$ to

$$\tilde{u}_k \in S = \min\{\Delta u_{\text{max}}, \frac{\partial y_{k+1}}{\partial u_k}^{-1} (\Delta y_{\text{max}})\}$$ 

The second term is the inverse of the Jacobian of the input-output transition map evaluated at $\Delta y_{\text{max}}$. Additional constraints might be required depending on the problem. The input signal selection problem can be formulated following optimization problem.

$$\tilde{u}_k = \arg \max_{u_k} J_{k+1}^{ij}$$

s.t. $\tilde{u}_k \in S$
Unfortunately, there are some cases that an input signal may not help with the detection problem. For example, if the Jacobian of the one step transition map is the same for two models, then we can not increase the gap between the two distributions using the input signal. Hence it is important to know when we can use probing signal for detection purposes.

**Lemma 2** Suppose the deterministic part of two models $H_i$ and $H_j$ are not identically the same. If the following two conditions hold for some $h \geq 1$, then there exist some input that can control the gap between the two future distributions.

1. Either $\frac{\partial}{\partial u_k} y_{k+h}^{(i)} \neq 0$ or $\frac{\partial}{\partial u_k} y_{k+h}^{(j)} \neq 0$
2. $\frac{\partial}{\partial u_k} y_{k+h}^{(i)} \neq \frac{\partial}{\partial u_k} y_{k+h}^{(j)}$

These are sufficient conditions for the existence of a probing input signal. In this case we have to change the output first and then let the future distributions change according to the system’s dynamic behavior. This requires an increased time horizon.

**Proposition 1** Suppose the deterministic part of the two models $H_i$ and $H_j$ are not identically same. Then the following input signal can control the gap between the two future distributions.

$$\tilde{u}_k = \arg \max_{u_k} J_{k+h_{i,j}}^{i,j}$$

s.t. $\tilde{u}_k \in S$

where $h_{i,j}$ is the minimum integer satisfying the above Lemma 1, and $J_{k+h_{i,j}}^{i,j}$ is defined as follows.

Type 1: $J_{k+h_{i,j}}^{i,j} = \frac{1}{2}(D_{k+h_{i,j}/k+k+h_{i,j}-1}^{i,j} + D_{k+k+h_{i,j}/k+k+h_{i,j}-1}^{i,j})$

Type 2: $J_{k+h_{i,j}}^{i,j} = p(H_i|y^k, u^{k-1})D_{k+h_{i,j}/k+k+h_{i,j}-1}^{i,j}$

$$+ p(H_j|y^k, u^{k-1})D_{k+h_{i,j}/k+k+h_{i,j}-1}^{i,j}$$

where $D_{k+h_{i,j}/k+k+h_{i,j}-1}^{i,j}$ is the conditional relative entropy defined as

$$D_{k+h_{i,j}/k+k+h_{i,j}-1}^{i,j} = \mathbb{E}(\log \frac{p(y_{k+h_{i,j}}|y^k, u^{k+h_{i,j}-1}, H_i)}{p(y_{k+h_{i,j}}|y^k, u^{k+h_{i,j}-1}, H_j)} | y^k, u^{k+h_{i,j}-1}, H_i)$$
Generally unless the system is modeled by a linear Gaussian I/O map, the optimization problem can not be solved exactly. However a simple numerical solution can be coped with obtaining a solution in real time (e.g., gradient based method). If the performance function over the input space is highly nonlinear and the allowable search space is large then we may need several iterations to obtain an approximate solution. If the maximum allowable perturbation \( \tilde{u}_{k,\text{max}} \) is small and the performance function is not highly nonlinear in its feasible space of \( \tilde{u}_k \), we can propose an input signal as follows.

\[
\tilde{u}_k = \tilde{u}_{k,\text{max}} \cdot \frac{\nabla u_k f^{i,j}_{k+h_{i,j}}}{\| \nabla u_k f^{i,j}_{k+h_{i,j}} \|}
\]

where \( \tilde{u}_{k,\text{max}} \in S \) is the maximum allowable perturbation

Assume \( h_{i,j} = 1 \). The amount of net information added at each time due to this perturbed input signal is

\[
\Delta B(\tilde{u}_k)
\equiv B(p^{(i)}_{k+1/k}(\tilde{u}_k + \tilde{u}_k)) - B(p^{(i)}_{k+1/k}(\tilde{u}_k)) - B(p^{(i)}_{k+1/k}(\tilde{u}_k)) - B(p^{(i)}_{k+1/k}(\tilde{u}_k))
\]

where \( p^{(i)}_{k+1/k}(\tilde{u}_k + \tilde{u}_k) = E\{p(y_{k+1}|y^k, u^{k-1}, \tilde{u}_k + \tilde{u}_k, \theta^{(i)}, H_i)\} \}

Thus the contribution is obtained as follows.

\[
\Delta B(\tilde{u}_k) \times 100 \%
\]

If the true model \( H_{\text{true}} \) can be represented as a parametric perturbation \( \Delta \theta \) then the amount of information added due to the input signal is

\[
\Delta B(\tilde{u}_k; \Delta \theta)
\equiv E\{\log \frac{p(y_{k+1}|y^k, u^{k-1}, \tilde{u}_k + \tilde{u}_k, \theta^{(i)}, H_i)}{p(y_{k+1}|y^k, u^{k-1}, \tilde{u}_k; \theta^{(i)}, H_i)} | y^k, u^{k-1}, \tilde{u}_k + \tilde{u}_k, \theta^{(i)} + \Delta \theta^{(i)}, H_{\text{true}}\}
\]

\[
- E\{\log \frac{p(y_{k+1}|y^k, u^{k-1}, \tilde{u}_k; \theta^{(i)}, H_i)}{p(y_{k+1}|y^k, u^{k-1}, \tilde{u}_k; \theta^{(i)}, H_i)} | y^k, u^{k-1}, \tilde{u}_k; \theta^{(i)} + \Delta \theta^{(i)}, H_{\text{true}}\}
\]

Note the probing signal will be effective as long as \( \Delta B(\tilde{u}_k; \Delta \theta) > 0 \)
4.2 Input Signal Selection for Binary Detection of Linear Gaussian Processes

In this section, a simple example is presented to clarify and illustrate the ideas developed in the previous section. Suppose we are generating a single to accelerate the detection between the following two hypothesized linear Gaussian models.

\[ H_i : y_{k+1} = A_k^{(i)} y_k + B_k^{(i)} u_k + e_k^{(i)} \]
\[ H_j : y_{k+1} = A_k^{(j)} y_k + B_k^{(j)} u_k + e_k^{(j)} \]

where \( y_k \in \mathbb{R}^n \) is the observation vector, \( u_k \in \mathbb{R}^m \) is the input vector, \( e_k^{(i)} \in \mathbb{R}^n \), \( e_k^{(j)} \in \mathbb{R}^n \) are the noise vectors and \( \{e_k^{(i)}\}_{k=0}^\infty \), \( \{e_k^{(j)}\}_{k=0}^\infty \) are two i.i.d. process with

\[ e_k^{(i)} \sim N(0, R_i) \]
\[ e_k^{(j)} \sim N(0, R_j) \]

Assume the parameters of the models \( A_k^{(i)}, B_k^{(i)}, A_k^{(j)} \) and \( B_k^{(j)} \) are perfectly known. The observations are then generated by one of the hypothesized models. Assume \( B_k^{(i)} \neq B_k^{(j)} \). At time step \( k \), the conditional p.d.f. of the new observation is given by one of the followings.

\[ y_k^{(i)} \sim N(A_k^{(i)} y_k + B_k^{(i)} u_k, R_i) \]
\[ y_k^{(j)} \sim N(A_k^{(j)} y_k + B_k^{(j)} u_k, R_j) \]

Given \( \bar{u}_k \), the objective is to find \( \bar{u}_k \) such that the maximum accumulation speed of the information can be obtained. To be more specific

\[ \bar{u}_k = \arg \max_{u_k} \frac{J_{k+1}^{j,i}(\bar{u}_k + \bar{u}_k) = \frac{1}{2}(D_{k+1}^{j,i}(u_k) + D_{k+1}^{j,i}(u_k))}{\frac{1}{2}(D_{k+1}^{j,i}(u_k) + D_{k+1}^{j,i}(u_k))} \]
\[ s.t \quad |\bar{u}_k| \leq \bar{u}_{k,\text{max}} \]

One approach to solving this optimization problem, the Steepest Ascent Method, is considered here.

\[ \bar{u}_k = \bar{u}_{k,\text{max}} \cdot \frac{\nabla_{u_k} J_{k+1}^{j,i}(\bar{u}_k)}{\left\| \nabla_{u_k} J_{k+1}^{j,i}(\bar{u}_k) \right\|} \]
The relative entropy is can be found as follows.

\[ D_{k+1}^{i,j}(u_k) \]

\[ = D\{p(y_{k+1}|y^k, u^k, H_i) \parallel p(y_{k+1}|y^k, u^k, H_j)\} \]

\[ = E\{\log \frac{(2\pi)^{-\frac{d}{2}} |R_i|^{-\frac{1}{2}} \exp(-\frac{1}{2}(y_{k+1} - \tilde{y}_{k+1|k}^{(i)})^T R_i^{-1}(y_{k+1} - \tilde{y}_{k+1|k}^{(i)}))}{(2\pi)^{-\frac{d}{2}} |R_j|^{-\frac{1}{2}} \exp(-\frac{1}{2}(y_{k+1} - \tilde{y}_{k+1|k}^{(j)})^T R_j^{-1}(y_{k+1} - \tilde{y}_{k+1|k}^{(j)}))}\} \]

\[ = \frac{1}{2} \log \frac{|R_j|}{|R_i|} - \frac{1}{2} E\{(y_{k+1} - \tilde{y}_{k+1|k}^{(i)})^T R_j^{-1}(y_{k+1} - \tilde{y}_{k+1|k}^{(i)})|y^k, u^k, H_i\} \]

\[ + \frac{1}{2} E\{(y_{k+1} - \tilde{y}_{k+1|k}^{(j)})^T R_j^{-1}(y_{k+1} - \tilde{y}_{k+1|k}^{(j)}))|y^k, u^k, H_i\} \]

\[ = \frac{1}{2} \log \frac{|R_j|}{|R_i|} - \frac{1}{2} tr R_j^{-1} R_i + \frac{1}{2} \| (A_k^{(i)} - A_k^{(j)}) y_k + (B_k^{(i)} - B_k^{(j)}) u_k \|_{R_j^{-1}}^2 + \frac{1}{2} tr R_j^{-1} R_i \]

Thus, the performance function is can be obtained as follows.

\[ J_{k+1}^{i,j}(u_k) \]

\[ = \frac{1}{2} \{D_{k+1}^{i,j}(u_k) + D_{k+1}^{j,i}(u_k)\} \]

\[ = \frac{1}{4} \{ \| (A_k^{(i)} - A_k^{(j)}) y_k + (B_k^{(i)} - B_k^{(j)}) u_k \|_{R_j^{-1}}^2 \}

\[ + \| (A_k^{(i)} - A_k^{(j)}) y_k + (B_k^{(i)} - B_k^{(j)}) u_k \|_{R_i^{-1}}^2 \} + C \]

\[ = \frac{1}{2} \{ \| (A_k^{(i)} - A_k^{(j)}) y_k + (B_k^{(i)} - B_k^{(j)}) u_k \|_{(R_j^{-1} + R_i^{-1})/2}^2 \} + C \]

Therefore the gradient can be is

\[ \nabla_{u_k} J_{k+1}^{i,j}(\bar{u}_k) = (B_k^{(i)} - B_k^{(j)})^T \left( \frac{R_j^{-1} + R_i^{-1}}{2} \right) \{ (A_k^{(i)} - A_k^{(j)}) y_k + (B_k^{(i)} - B_k^{(j)}) \bar{u}_k \} \]

Accordingly, the optimal input \( \bar{u}_k \) is found by substituting the gradient to the Steepest Ascent equation.
Special Case 1: Even if $\tilde{u}_k = 0$ we can use
\[
\tilde{u}_k = \tilde{u}_{k,\text{max}} \cdot \frac{\nabla_{u_k} J_{k+1}^{i,j}(0)}{\left\| \nabla_{u_k} J_{k+1}^{i,j}(0) \right\|}
\]

Special case 2: If $\tilde{u}_k = 0$ and $A^{(i)} - A^{(j)} = 0$ then $\nabla \tilde{J} = 0$. Thus the gradient based approach can not be used. One feasible method is to use the eigenspace of the covariance matrix. In the following equation
\[
\max_{\tilde{u}_k} J_{k+1}^{i,j}(0 + \tilde{u}_k) = \max_{\tilde{u}_k} \frac{1}{2} \| (B_k^{(i)} - B_k^{(j)}) \tilde{u}_k \|_{(R_{i}^{-1} + R_{j}^{-1})/2}^2
\]
denote $\frac{1}{2}(B_k^{(i)} - B_k^{(j)})^T(R_{i}^{-1} + R_{j}^{-1})^{-1}(B_k^{(i)} - B_k^{(j)}) = K$. Since $R_i$ and $R_j$ are positive definite so are $R_{i}^{-1}$ and $R_{j}^{-1}$. Hence $K$ is positive definite and symmetric. Therefore the eigenvalues are all positive. Since $K$ is symmetric, it can be decomposed as follows.
\[
K = \sum_{k=1}^{n} \lambda_k u_k v_k^T
\]
Here $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > 0$ and $v_1, v_2, \ldots, v_n$ are the eigenvalues and corresponding orthonormal eigenvectors of $K$. Accordingly, the following inequality is existed.
\[
\tilde{u}_k^T K \tilde{u}_k \leq \lambda_1 \| \tilde{u}_k \|^2
\]
and thus the maximum is achieved iff $\tilde{u}_k$ is in the direction of the eigenvector corresponding to maximum eigenvalue, $\lambda_1$. Therefore the optimal signal is
\[
\tilde{u}_k = \tilde{u}_{k,\text{max}} v_1
\]
where $v_1$ is the unit eigenvector of $K$ whose eigenvalue is maximum. Note this direction is the steepest direction in the eigenspace of $K$.

Special case 3: If $B_k^{(i)} = B_k^{(j)}$ a different time horizon should be used as follows.
\[
\tilde{u}_k = \arg \max_{\tilde{u}_k} J_{k+2}^{i,j}(\tilde{u}_k + \tilde{u}_k) = \frac{1}{2} (D_{k+2}^{i,j}(u_k) + D_{k+2}^{i,i}(u_k))
\]
\[
s.t \quad |\tilde{u}_k| \leq \tilde{u}_{k,\text{max}}
\]
The procedure to find $\bar{u}_k$ is similar to the one step horizon case. Since the conditional entropy is

$$D_{k+2}^{i,j}(u_k) = \frac{1}{2} \| (A_k^{(i)} - A_k^{(j)}) y_k + (A_k^{(i)} B_k^{(i)} - A_k^{(j)} B_k^{(j)}) u_k \|^2_{R_k^{-1}} + C_1$$

the performance function is

$$J_{k+2}^{i,j}(u_k) = \frac{1}{2} \{ \| (A_k^{(i)} - A_k^{(j)}) y_k + (A_k^{(i)} B_k^{(i)} - A_k^{(j)} B_k^{(j)}) u_k \|^2_{(R_k^{-1} + R_k^{-1})/2} + C_2 \}.
$$

Hence

$$\nabla_{u_k} J_{k+2}^{i,j}(u_k) = \left( A_k^{(i)} B_k^{(i)} - A_k^{(j)} B_k^{(j)} \right)^T \left( \frac{R_k^{-1} + R_k^{-1}}{2} \right) \cdot \left\{ (A_k^{(i)} - A_k^{(j)}) y_k + (A_k^{(i)} B_k^{(i)} - A_k^{(j)} B_k^{(j)}) \bar{u}_k \right\}.$$

Therefore the optimal input $\bar{u}_k$ is found by substituting the gradient to the Steepest Ascent equation.

$$\bar{u}_k = \bar{u}_{k,\text{max}} - \frac{\nabla_{u_k} J_{k+2}^{i,j}(u_k)}{\| \nabla_{u_k} J_{k+2}^{i,j}(u_k) \|}.$$

### 4.3 M-ary detection case

Suppose there are three hypotheses $H_1, H_2$ and $H_3$. At time step $k$, the information available is $z^k = \{ y^k, u^{k-1} \}$. Denote $\lambda_k^{2,1}, \lambda_k^{3,1}, \lambda_k^{3,2}$ the log posterior ratios $\log\{p(H_2|z^k)/p(H_1|z^k)\}, \log\{p(H_3|z^k)/p(H_1|z^k)\}$ and $\log\{p(H_3|z^k)/p(H_2|z^k)\}$ respectively. A three hypotheses case is considered first, but this will be generalized later.

The trajectory of $\lambda_k$ can be visualized in the log-posterior-ratio space. See Figure 4.1, where the $x$-axis is $\lambda_k^{2,1}$ and the $y$-axis is $\lambda_k^{3,1}$. Suppose $\lambda_k$ lies in the $D$ region, then more information (or evidence) about $\lambda_k^{3,2}$ will be required than that of $\lambda_k^{2,1}$ or $\lambda_k^{3,1}$; at least for a few time steps from $k + 1$. The reason is that $H_2$ or $H_3$ will be more likely than $H_1$, and it is ambiguous between $H_2$ or $H_3$ because we have accumulated more information about $H_2/H_1$ and $H_3/H_1$ than $H_3/H_2$.

Note that if only the information (or evidence) about $H_3/H_1$ and $H_3/H_1$ increase, then no matter how much such information increases, the detector’s stopping rule can not be activated. In order to obtain a full information set to make a decision, it is necessary to reach the given thresholds quickly. Thus the information about
$H_3/H_2$ is the critical information to accomplish this. By this way, the decision making process can be efficiently speeded up by using a small input signal $\tilde{u}_k$. Moreover, by systematically making the mean values of each distribution farther apart, protections from the outliers distribution can be performed too.

On the other hand, suppose $\lambda_k$ is in region $B$. Then information about $H_2/H_1$ will be more required than that of $H_3/H_1$, $H_3/H_2$. If $\lambda_k$ is in region $A$, then the signal dependency on the region is not so critical as the situations when $\lambda_k$ is in the regions $B, C$ or $D$.

Based on this observation, the formulation for the input signal selection problem can be performed by the following optimization problem.

$$\tilde{u}_k^* = \arg \max_{\tilde{u}_k} P(H_1|z^k)P(H_2|z^k) \cdot J_{k+h_{2,1}}^{2,1} + P(H_1|z^k)P(H_3|z^k) \cdot J_{k+h_{3,1}}^{3,1}$$

$$+ P(H_2|z^k)P(H_3|z^k) \cdot J_{k+h_{3,2}}^{3,2}$$

The pairwise products of two posteriors are used as weights, and this makes the optimal value of the performance functional smooth with respect to changes in the posteriors. Smoothness is important because it is difficult to find a separating surface between the regions $A, B, C$ and $D$. This can be generalized into the $M$ hypotheses case as follows.

$$\tilde{u}_k^* = \arg \max_{u_k} \sum_{i=1}^{M} \sum_{i\neq j}^{M} P(H_i|z^k) \cdot P(H_j|z^k) \cdot J_{k+h_{i,j}}^{i,j}$$

Although this gives a meaningful way to design a probing input signal $u_k$, it requires $M(M-1)/2$ computations to evaluate the performance function. Actually the proposed detection mechanism only requires information about $\{\lambda^*_{i,1}, \lambda^*_{i,2}, \ldots, \lambda^*_{i,M}\} = \{\lambda^*_{i,i}^*\}$ only, where $i^*$ denotes the index of the largest a posterior. Therefore the following selection criteria would be computationally more reasonable.

$$\tilde{u}_k^* = \arg \max_{u_k} \sum_{j=1, j\neq i^*}^{M} P(H_j) \cdot J_{k+h_{i,j}}^{i,j}$$

This criteria requires $M-1$ computations, but even this can be problematic in practical implementation. We might need a selection criteria which is composed of just a
single performance index. For this reason, the following proposition can be used for the probing signal generation.

\[ \tilde{u}_k^* = \arg \max_{u_k} J_{i^*, j^*}^{i^*, j^*} \]

where \[ j^* = \arg \max_{j \in J} p(H_j | z^k) \]

\[ J = \{ j \mid j \neq i^* \text{ and } p(H_{i^*} | z^k) < p(H_j | z^k) \cdot T_{i^*, j} \} \]

This input signal \( u_k^* \) will probe the system in such a way that the largest a posteriori distribution and the second largest which does not satisfy the given constraint are separated. This input signal will then help to reduce the ambiguity between these two hypotheses and speed up the detection process, by systematically suppressing the strongest incorrect hypothesis.
Figure 4.1: A log posterior ratio space.
Chapter 5

Change Detection in Dynamic Patterns and Time Estimation

5.1 Change Detection in M-ary Detection

In most cases, no matter what detection method is used, the initial hypothesis which is detected is usually the one that indicates normal mode of the dynamic system. That means if we start some machine then usually it starts with normal behavior. In this case, rather than terminate the detection, we want to continue the test so that abnormal operating behavior can be identified if it occurs. This requires the design of the detection mechanism that can, systematically and in some optimal fashion, detect a change from a known hypothesis to unknown hypothesis.

Most of the conventional detection methods accomplish this task by continuously running, using the same detection mechanism. Although most sequential detection mechanisms are designed based on the assumption that they are initialized with zero or minimal information (e.g., priors), the detection algorithm must continue to work even if normal behavior is detected.

For example, in the MMA method, we set a fixed threshold and let the termination rule be active whenever the threshold is penetrated by some a posterior. We also usually set lower bounds for the posteriors so that they don’t become too small. This modification is often primary, since if posteriors get too large or too small.
the detection time can become very large. (If a posterior goes approximately to zero and the algorithm is not reset to its priors, it may take a long time for it to reach a prescribed threshold, even if a failure occurs). With this modification, the MMA method has been successfully applied to change detection.

However, there is one important thing that is still missing. If the detection algorithm uses the same decision rule and thresholds that are used to initialize the procedure, the conventional mechanism may require waiting for a long time before a change is detected. In fact, at the time when a change occurs, the a posteriori corresponding to the true hypothesis will most likely begin from a low initial value near the lower bound, and the a posterior corresponding to the normal hypothesis will start from a high value, which makes the detection difficult. The major difficulty is, of course, that we don’t know when the change occurs, given only partial observations of the dynamic system.

One may ask why should the test’s terminal conditions be changed because the algorithm can still operate with improper alarm levels? Possible answers include: (1) the relationship between detection and the probabilities of incorrect detections which was tightly set are lost (2) in the binary detection case, the optimal detection property is lost. (3) trade off between the probability of incorrect detection in the initial stage and that of change detection in subsequent stages leads to the dilemma: if we design the thresholds in the initial stage then the change detection would be prolonged, and if we design the thresholds for change detection stage then the algorithm can have an unexpected large wrong alarm rates in the initial stage, (4) besides for correct detection, estimating the time of model switching is important because it supplies important information for the controller. Hence it is the objectives of this chapter to develop a generalized concept that provides an approach to the detection of the times where the model switches in a multiple hypothesis case. Our objectives can be explained as follows.

Given the sequence \( \{y_k\}_{k=0}^{\infty} \) and the a posterior functions \( p(y^k | H_i) \) \( i = 1, 2, \ldots M \), the objective is to decide the possible change from a known hypothesis\(^1\) to

\(^1\)For a moment, we assume that the normal operating hypothesis is \( H_1 \)
some unknown hypothesis $H_i$ as reliable and quickly as possible.

$$H_1 \rightarrow H_i \quad \text{for some } i \neq 1$$

Evidently, we can estimate the time when the model switching occurs. because $p(H_i \mid y^k)$ should be decreasing and $p(H_{true} \mid y^k)$ should be increasing from the time model switch occurs. Since the occurrence of the change is uncertain, which means we don’t know the time of occurrence of the switch, a sequential type decision rule is required.

In this section we consider this problem, by estimating the time at which the change started and adaptively adjusting the decision rule and the thresholds to equalize the probability of an incorrect decision between the initial stage and the change detection stage. For a moment. we shall assume the initial hypothesis is $H_1$. Actually this assumption is not necessary and we will generalize this later. This assumption will, however, avoid unnecessary complexity in explanatory which follows.

Define $y^k = \{y_0, y_1, ..., y_k\}$ and $y^{k,N} = y^k - y^{N-1} = \{y_N, y_{N+1}, ..., y_k\}$. Suppose that $\varphi_k = 1$ and $\delta_k = E_1$. This means that, at time step $k$, the terminal decision rule is active and the decision rule selected $H_1$. In this case we conclude that the current hypothesized model is $H_1$. If $\varphi_k = 0$, then we would be selecting a different test set. In order to make a test set, we first estimate the time of model switching and the most plausible hypothesis, except $H_1$. We do this by decomposing the observation set into two subsets. The underlying idea is based on the observation that if a model switch occurs then $p(H_1 \mid y^k)$ will decrease and $p(H_{true} \mid y^k)$ will increase right after the model switch. The selection is obtained by choosing the arguments which maximize the likelihood ratio

$$(\ast, \tilde{N}) = \max_{2 \leq i \leq M, 1 \leq N < k} \frac{\log \frac{p(y^{k,N+1} \mid y^N, H_i)}{p(y^{k,N+1} \mid y^N, H_1)}}$$

(5.1)

where $\ast$ denotes the candidate hypothesis and $\tilde{N}$ denotes the time. Here the relation $(\ast, \tilde{N})$ does not mean that at time $\tilde{N}$ the hypothesis changed to $H_\ast$. The correct meaning is that if a model switching did occur then it would be close to the estimation $(\ast, \tilde{N})$. However, we are not sure whether a model switch has occurred or not. (If a
switch did occur, the likelihood of the true hypothesis should be increasing right after the switch, but this is not a sufficient condition.) To confirm this we check whether the new observation set, collected from the estimated time $\hat{N}$ up to $k$, supports the hypothesis $H_*$ or not, based on the updated information. If an updated terminal decision rule is activated (i.e. $\varphi_k = 1$) then we set $\delta_k = E_*$ and declare that the model switching from $H_1$ to $H_*$ occurred at time $\hat{N}$. Otherwise we disregard ($*, \hat{N}$) and continue the next test based on the conventional method.

To evaluate the ratio of conditional density functions, we might guess that it can easily be computed using posteriors at time $k$ and $\hat{N}$. The following lemma suggests an easy way to evaluate the ratio.

**Lemma 3** For hypothesis model $H_i$ and $H_j$ with nonzero priors, the following equalities hold.

$$\frac{p(y^{k,N+1} \mid y^N, H_i)}{p(y^{k,N+1} \mid y^N, H_j)} = \frac{p(H_i \mid y^k)}{p(H_j \mid y^k)} \left( \frac{p(H_i \mid y^N)}{p(H_j \mid y^N)} \right)^{-1}$$

**Proof.** We start by evaluating the posterior ratio between the same models with different observation set.

$$\frac{p(H_i \mid y^k)}{p(H_i \mid y^N)} = \frac{p(y^k \mid H_i)}{p(y^N \mid H_i)} \frac{p(H_i)}{p(H_i)}$$

**Bayes’ rule**

$$= \frac{p(y^N \mid H_i) p(H_i)}{p(y_1 \mid H_i) p(y_2 \mid y_1, H_i) \cdots p(y_k \mid y^{k-1}, H_i)}$$

**Chain rule**

$$= p(y_{N+1} \mid y^N, H_i) p(y_{N+2} \mid y^{N+1}, H_i) \cdots p(y_k \mid y^{k-1}, H_i)$$

$$= p(y^{k,N+1} \mid y^N, H_i)$$

Similarly,

$$\frac{p(H_j \mid y^k)}{p(H_j \mid y^N)} = p(y^{k,N+1} \mid y^N, H_j)$$

Hence

$$\frac{p(y^{k,N+1} \mid y^N, H_i)}{p(y^{k,N+1} \mid y^N, H_j)} = \frac{p(H_i \mid y^k)}{p(H_j \mid y^k)} \left( \frac{p(H_i \mid y^N)}{p(H_j \mid y^N)} \right)^{-1} = \frac{p(H_i \mid y^k)}{p(H_j \mid y^k)} \left( \frac{p(H_i \mid y^N)}{p(H_j \mid y^N)} \right)^{-1}$$

$\Box$
In order to implement this evaluation strategy we have to keep and update posterior evolutions. By assuming that the average delay (between the observation time and the occurrence of model switching) is a nonzero finite number, we can introduce an observation window and search over this windowed observation set. Let $W$ denote the window size, then we can use the following search set

$$k - W \leq N < k$$

in the equation 5.1.

As a confirmation procedure, we revise the test by using the same decision rule, but adaptive thresholds are used.

$$\log \frac{p(H_* \mid y^k)}{p(H_j \mid y^k)} \geq \hat{T}_{*,j} \quad \text{for all } H_1 \leq H_j \leq H_M, \ H_j \neq H_*$$

or equivalently by Lemma

$$\log \frac{p(y^{k,N} \mid H_*)}{p(y^{k,N} \mid H_j)} + \log \frac{p(H_* \mid y^{N})}{p(H_j \mid y^{N})} \geq \hat{T}_{*,j} \quad \text{for all } H_1 \leq H_j \leq H_M, \ H_j \neq H_*$$

where $\hat{T}$ denote an adaptive threshold. The Lemma in chapter 3 gives an approach to selecting its thresholds. Thus

$$\hat{T}_{*,j} = \frac{\beta_{\omega*}}{\beta_{\omega j}} + \log \frac{p(H_* \mid y^{N})}{p(H_j \mid y^{N})} \quad \text{for all } H_1 \leq H_j \leq H_M, \ H_j \neq H_*$$

Therefore the test to be performed is.

$$if \quad \frac{p(H_* \mid y^k)}{p(H_j \mid y^k)} \geq \frac{\beta_{\omega*}}{\beta_{\omega j}} \cdot \frac{p(H_* \mid y^{N})}{p(H_j \mid y^{N})} \quad \text{for all } H_1 \leq H_j \leq H_M, \ H_j \neq H_*$$

$$then \quad (\varphi_k, \delta_k) = (1, E_*)$$

$$else$$

$$(\varphi_k, \delta_k) = (0, \text{arbitrary})$$

So far, the basic switching detection mechanism has been explained. Can we guarantee that the most plausible hypothesis based on evidence framework always would have the highest a posterior at any given instants of time? The answer is
no. It depends on how initial posteriors are distributed when the switching occurs. If the desired probabilities of incorrect decision are not too small, it will not be difficult to meet the threshold condition, so when the test terminates, the decision rule selects the hypothesis with not necessarily the highest a posterior. This might not seem to be a desirable feature but it should be, because we are interested only in the change of posteriors rather than the posteriors themselves. (Strictly speaking, \( p(H \mid y^k) \) means the a posterior probability of \( H \) based on the entire observation set \( y^k = \{y_0, y_1, ..., y_k\} \).) The above test can accomplish it. If the thresholds are not set too small relative to the bounds, then the most plausible hypothesis based on the data will always have the highest a posterior or at least the second highest a posterior.

The test method proposed requires searching for \((*, \widetilde{N})\) over a two-dimensional set in equation 5.1, which may be burdensome for real time computations in some situations. Under the assumption that the most plausible hypothesis will have the highest a posterior except for \( H_1 \), we can reduce the proposed two-dimensional search to two one-dimensional searches.

\[
* = \arg \max_{2 \leq i \leq M} \ p(H_i \mid y^k)
\]

\[
\widetilde{N} = \arg \max_{k-W \leq N < k} \log \frac{p(y^{k,N+1} \mid y^N, H_\ast)}{p(y^{k,N+1} \mid y^N, H_1)}
\]

Although, ideally, the switching detection test should be performed at every instance of time, it makes a good sense to test it only after a certain amount of information has accumulated or a certain amount of time has passed, since \( H_1 \) is detected. This can reduce the unnecessary computation. The estimated time has larger error, but can be acceptable for convenience. For example the detector can start checking terminal conditions, that is after some a posterior \( p(H_i \mid y^k) \), \( H_i \neq H_1 \) overtakes the posterior for \( H_1 \).

if \( \arg \max_{1 \leq i \leq M} \ p(H_i \mid y^k) \neq 1 \)

then search \((*, \widetilde{N})\) and perform termination test \( (5.2) \)

otherwise

skip termination test

Note however, if the lower bound is too small (i.e difficult to be met) then it takes
a relatively long time for the corresponding posterior to grow sufficiently and this can cause a long delay in initiating the terminal test. However, even if we use the simplified test as above 5.2, it is much better than the conventional method. (i.e detection is occurred more quickly than the conventional method.)

### 5.2 Generalized Change Detection

In the last section, we assumed that the change was from $H_1$ to another hypothesis $H_i, i \neq 1$. In general, we also want the detector to continue working even after some failure has been detected. In many cases, we can not stop the controller after a failure, requiring self-reconfigurable and fault tolerant controller implementation.

In this section, we address a generalized change detection, which copes with the $M$-ary case and model switching is not always from $H_1$. Suppose, at the present time, hypothesis $H_s$ is detected, and we want to detect a possible change to another hypothesis, i.e

$$H_s \rightarrow H_i \quad \text{for some } 1 \leq i \leq M, i \neq s$$

This change detection problem requires replacing $H_1$ in the previous scheme with $H_s$, with some additional modifications. An estimate of the time the change (i.e failure) occurred and corresponding hypothesis is given by

$$(\ast, \bar{N}) = \arg \max_{1 \leq i \leq M, i \neq s, k-W \leq N < k} \log \frac{p(y_{i,N+1}^{k} \mid y^N, H_i)}{p(y_{k,N+1}^{k} \mid y^N, H_s)}$$

We can also use two one-dimensional search for simplicity.

$$\ast = \arg \max_{1 \leq i \leq M, i \neq s} p(H_i \mid y^k)$$

$$\bar{N} = \arg \max_{k-W \leq N < k} \log \frac{p(y_{k,N+1}^{k} \mid y^N, H_s)}{p(y_{k,N+1}^{k} \mid y^N, H_s)}$$

In both case, the adaptive termination condition to be followed is

$$\frac{p(H_s \mid y^k)}{p(H_j \mid y^k)} \geq \frac{\beta_{s,i}}{\beta_{s,j}} \cdot \frac{p(H_s \mid y^{\bar{N}})}{p(H_j \mid y^{\bar{N}})} \quad \text{for all } H_1 \leq H_j \leq H_M, H_j \neq H_s.$$
And for simplicity, we can defer testing the termination condition until some a posterior \(p(H_i \mid y^k), H_i \neq H_s\) becomes a leading posterior as follows.

\[
\text{if } \arg \max_{1 \leq i \leq M} p(H_i \mid y^k) \neq s
\]

\[
\text{then search } (\ast, \widehat{N}) \text{ and perform termination test}
\]

\[
\text{otherwise skip termination test}
\]

We briefly summarize the overall algorithm next.

Algorithm for Change Detection

\[
k = 0
\]

\[
\text{step1 : } k = k + 1
\]

\[
\text{compute } p(H_i \mid y^k) \text{ for } 1 \leq i \leq M \text{ with some lower bounds}
\]

\[
\text{set } \ast = \arg \max_{1 \leq i \leq M} p(H_i \mid y^k)
\]

\[
\text{if } \frac{p(H_\ast \mid y^k)}{p(H_j \mid y^k)} \geq \beta_{s\ast} \frac{p(H_\ast)}{p(H_j)} \text{ for all } 1 \leq j \leq M, j \neq \ast.
\]

\[
\text{then declare } H_\ast, \text{ set } (\phi_k, \delta_k) = (1, E_\ast), s = \ast \text{ and go to step2}
\]

\[
\text{else set } (\phi_k, \delta_k) = (0, \text{arbitrary}) \text{ and repeat step1}
\]

\[
\text{step2 : } k = k + 1
\]

\[
\text{compute } p(H_i \mid y^k) \text{ for } 1 \leq i \leq M \text{ with some lower bounds}
\]

\[
\text{if } \frac{p(H_s \mid y^k)}{p(H_j \mid y^k)} \geq \beta_{s1} \frac{p(H_s)}{p(H_j)} \text{ for all } 1 \leq j \leq M, j \neq s.
\]

\[
\text{then declare } H_s, \text{ set } (\phi_k, \delta_k) = (1, E_s) \text{ and repeat step2}
\]

\[
\text{else}
\]

\[
\text{set } \ast = \arg \max_{1 \leq i \leq M} p(H_i \mid y^k) \text{ and}
\]

\[
\widehat{N} = \arg \max_{k \geq k \leq \widehat{N} < k} \log \frac{p(y^k, N+1 \mid y^N, H_\ast)}{p(y^k, N+1 \mid y^N, H_s)}
\]

\[
\text{if } \frac{p(H_\ast \mid y^k)}{p(H_j \mid y^k)} \geq \beta_{s\ast} \frac{p(H_\ast \mid \widehat{N})}{p(H_j \mid \widehat{N})} \text{ for all } 1 \leq j \leq M, j \neq \ast.
\]

\[
\text{then declare } H_\ast, \text{ set } (\phi_k, \delta_k) = (1, E_\ast), s = \ast \text{ and}
\]

\[
\text{repeat step2}
\]
else set \((\phi_k, \delta_k) = (0, \text{arbitrary})\) and repeat step2

---

**Simplified Change Detection Algorithm**

\[
k = 0
\]

*step1: \(k = k + 1\)*

Compute \(p(H_i \mid y^k)\) for \(1 \leq i \leq M\) with some lower bounds

set \(s = \arg \max_{1 \leq i \leq M} p(H_i \mid y^k)\)

if \(\frac{p(H_* \mid y^k)}{p(H_j \mid y^k)} \geq \frac{\beta_{s|j} p(H_s)}{\beta_{s|j} p(H_j)}\) for all \(1 \leq j \leq M, j \neq *\).

then declare \(H_*\), set \((\phi_k, \delta_k) = (1, E_*)\), \(s = *\) and go to step2

else set \((\phi_k, \delta_k) = (0, \text{arbitrary})\) and repeat step1

*step2: \(k = k + 1\)*

Compute \(p(H_i \mid y^k)\) for \(1 \leq i \leq M\) with some lower bounds

if \(\frac{p(H_* \mid y^k)}{p(H_j \mid y^k)} \geq \frac{\beta_{s|j} p(H_s)}{\beta_{s|j} p(H_j)}\) for all \(1 \leq j \leq M, j \neq s\).

then declare \(H_*\), set \((\phi_k, \delta_k) = (1, E_*)\) and repeat step2

else

if \(\arg \max_{1 \leq i \leq M} p(H_i \mid y^k) \neq s\),

set \(s = \arg \max_{1 \leq i \leq M} p(H_i \mid y^k)\) and

\[
\overline{N} = \arg \max_{k - W \leq N < k} \log \frac{p(y^{k,N+1} \mid y^N, H_*)}{p(y^{k,N+1} \mid y^N, H_*)}
\]

if \(\frac{p(H_* \mid y^N)}{p(H_j \mid y^N)} \geq \frac{\beta_{s|j} p(H_* \mid y^N)}{\beta_{s|j} p(H_j \mid y^N)}\) for all \(1 \leq j \leq M, j \neq *\).

then declare \(H_*\), set \((\phi_k, \delta_k) = (1, E_*)\), \(s = *\)

and repeat step2

else set \((\phi_k, \delta_k) = (0, \text{arbitrary})\) and repeat step2

else set \((\phi_k, \delta_k) = (0, \text{arbitrary})\) and repeat step2
So far, the detection of changes in systems using a statistic information (hypothesis) approach has been discussed. Alternatively, this sequential change detection problem can be interpreted as a pattern classification where the objective is the detection of dynamic patterns or the classification of dynamic patterns. Classical pattern recognition is defined on a sample space where the samples are independent and detections are made at every time step. The objective is to select one in $M$ classifications, at each time (or with fixed number of samples). Classification of dynamic patterns is better defined on the observation space where samples are dynamically correlated, and the classification of samples at every time instance is not mandatory (i.e., the number of sample is a RV too). This increases the degrees of freedom in the problem formulation. In this case, our objectives are to design a termination decision rule for one of $M$-classification and to estimate the time of switching. This problem, being generally regarded as more difficult than classical pattern recognition, has not been sufficiently studied, despite its importance and wide applications. Throughout this section, we have reflected on some important features and developed an implementation scheme based on statistical information theory, and we conclude this chapter by emphasizing that further studies are needed.
Chapter 6

Robust Detection

In practical situations. there are uncertainties that would disturb the detection system. Generally, when we say robustness, there are two meanings: one is robustness in the presence of modeling uncertainties which can result from various reasons: modeling error, model perturbation due to changes in the surroundings and unmodeled dynamics excited by unwanted disturbances, etc. The other meaning is robustness in the presence of statistical uncertainties, for example, incorrectness in density models and the occasional occurrence of low probability events. With regard to detection, both sources of error are important because they can reduce the detection performance seriously. These undesirable uncertainties can occur in both the normal and failure modes of the dynamic system. This chapter discusses the robust detection in the presence of model uncertainties.

6.1 Robust Detection in the Presence of Model Uncertainties

Generally, uncertainties can be modeled mathematically in two ways: parametrically and non-parametrically. In this thesis, parametric perturbations are used to model these uncertainties. Suppose the following $M$ models $H_1, H_2, ..., H_M$ were
obtained by using parametric identification methods.

\[ H_1: \quad y_{k+1} = f^{(1)}(y^k, u^k; \theta^{(1)}) + e^{(1)}_{k+1} \]
\[ \vdots \quad \vdots \]
\[ H_M: \quad y_{k+1} = f^{(M)}(y^k, u^k; \theta^{(M)}) + e^{(M)}_{k+1} \]

(6.1)

where for each \( i, \ 1 \leq i \leq M \), \( \theta^{(i)} = (\theta_{1}^{(i)}, \theta_{2}^{(i)}, \ldots, \theta_{m(i)}^{(i)})^T \in R^{m(i)} \) denotes fixed parameter vectors, \( f^{(i)}(\cdot, \cdot; \theta^{(i)}): R^{d(i)} \to R^{d2(i)} \) denotes discrete time continuous valued functions (possibly nonlinear), and \( e^{(i)}_{k+1} \) is an zero mean i.i.d. noise with \( p(e^{(i)}_{k+1}) = N(0, R^{(i)}) \). The parametrically perturbed models are given by:

\[ H_1: \quad y_{k+1} = f^{(1)}(y^k, u^k; \theta^{(1)} + \Delta \theta^{(1)}_k) + e^{(1)}_{k+1} \]
\[ \vdots \quad \vdots \]
\[ H_M: \quad y_{k+1} = f^{(M)}(y^k, u^k; \theta^{(M)} + \Delta \theta^{(M)}_k) + e^{(M)}_{k+1} \]

(6.2)

where \( \Delta \theta^{(i)}_k = (\Delta \theta^{(i)}_{1,k}, \Delta \theta^{(i)}_{2,k}, \ldots, \Delta \theta^{(i)}_{m(i),k})^T \in R^{m(i)} \) denotes parametric perturbations in the parameter vectors for \( 1 \leq i \leq M \). Suppose these perturbations are subject to the constraint sets \( \Delta \theta^{(i)}_{1,k} \in G^{(i)}_1, \Delta \theta^{(i)}_{2,k} \in G^{(i)}_2, \ldots, \Delta \theta^{(i)}_{m(i),k} \in G^{(i)}_{m(i)} \) for \( 1 \leq i \leq M \) which are problem dependent. We assume these constraints are fixed with respect to time. A simple example which is commonly used is

\[ G^{(i)}_j = \{ \Delta \theta^{(i)}_j \mid -\overline{\Delta \theta^{(i)}_j} \leq \Delta \theta^{(i)}_j \leq \overline{\Delta \theta^{(i)}_j} \} \quad \text{for } 1 \leq i \leq M, \ 1 \leq j \leq m(i) \]

In this thesis, parametric changes are modeled as random variables. A variable which can not be predicted, but whose statistical behavior can be estimated. By such modeling, a detection problem in the presence of uncertainties can be formulated within the framework of statistical information. Thus, our objective is to find the conditional density functions. This type of approach has not been extensively studied in the detection problem.

The basic random variables are \( \{ \Delta \theta^{(i)}_{1,k}, \Delta \theta^{(i)}_{2,k}, \ldots, \Delta \theta^{(i)}_{m(i),k}, e^{(i)}_k \} \) for \( 1 \leq i \leq M \), and they are assumed to be i.i.d. Accordingly,

\[
cov(\Delta \theta^{(i)}_k, e^{(i)}_k) = \begin{pmatrix}
\cov(\Delta \theta^{(i)}_k, e^{(i)}_k) & 0 \\
0 & \cov(e^{(i)}_k)
\end{pmatrix}
\]
where \( \text{cov}(\Delta \theta_k^{(i)}) = \text{diag}\{\text{cov}(\Delta \theta_{1,k}^{(i)}), \text{cov}(\Delta \theta_{2,k}^{(i)}), ..., \text{cov}(\Delta \theta_{m(i),k}^{(i)})\} \). For parameters which are certain (i.e., \( \Delta \theta_{l,k}^{(i)} \approx 0 \) for some \( l \)), \( \text{cov}(\Delta \theta_{l,k}^{(i)}) = 0 \) can be applied. Two possible such distributions are the **uniform** distribution and the **Gaussian** distribution as follows.

\[
p(\Delta \theta_k^{(i)}) = \text{unif}(\Delta \theta^{(i)})
\]

where \( \bar{\Delta \theta}^{(i)} = (\bar{\Delta \theta}_1^{(i)}, \bar{\Delta \theta}_2^{(i)}, ..., \bar{\Delta \theta}_{m(i)}^{(i)}) \)

\[
p(\Delta \theta_k^{(i)}) = N(0, Q^{(i)})
\]

where \( Q^{(i)} = \text{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_{m(i)}^2) \)

The uniform distribution means that the uncertainties are equal within parametric bounds. On the other hand, the Gaussian distribution means that the parametric uncertainties are weighted more favorably toward the mean values. In this thesis the latter distribution is considered and the corresponding variances are assumed to be small.

As mentioned, our objective is to find the conditional p.d.f. One way to do this is to apply the Taylor series and treat the first few terms. Assuming \( f^{(i)} \) is analytic at \( \theta^{(i)} \). To be more specific.

\[
y_{k+1} = f^{(i)}(y^k, u^k; \theta^{(i)}) + \Delta \theta_k^{(i)} + e_{k+1}^{(i)}
\]

\[
y_{k+1} = f^{(i)}(y^k, u^k; \theta^{(i)}) + \left[ \frac{\partial f^{(i)}(y^k, u^k; \theta^{(i)})}{\partial \theta} \right]^T \Delta \theta_k^{(i)}
\]

\[
+ \frac{1}{2} \Delta \theta_k^{(i)}^T \left[ \frac{\partial^2 f^{(i)}(y^k, u^k; \theta^{(i)})}{\partial \theta^2} \right] \Delta \theta_k^{(i)} + O(|| \Delta \theta_k^{(i)} ||^3) + e_{k+1}^{(i)}
\]

In the last equation, the second term and the fifth term have Gaussian distribution. However the third term and the fourth term do not. Hence the entire term does not have Gaussian distribution, and this makes it difficult to obtain the conditional p.d.f. If the perturbations are all small, then the high order terms beginning from the quadratic term can be neglected. In that case, the above equation becomes

\[
y_{k+1} = f^{(i)}(y^k, u^k; \theta^{(i)}) + \left[ \frac{\partial f^{(i)}(y^k, u^k; \theta^{(i)})}{\partial \theta} \right]^T \Delta \theta_k^{(i)} + e_{k+1}^{(i)} \tag{6.3}
\]
Note that this approximation is exact if the system dynamics \( f^{(i)} \) is linear. The RHS of the Eq. 6.3 can be decomposed into two parts: A deterministic part which determines the estimation of the future output (or mean value of the one step future output) and an uncertainty part which controls the quality of that estimation. A great benefit in the usage of this approximation is that the conditional p.d.f. becomes a Gaussian distribution. This enables us to obtain the conditional p.d.f. as follows.

\[
p(y_{k+1}| y^k, u^k) = N(m_k^{(i)}, \Gamma_k^{(i)})
\]

where

\[
m_k^{(i)} = f(y^k, u^k; \theta^{(i)})
\]

\[
\Gamma_k^{(i)} = \left[ \frac{\partial f^{(i)}(y^k, u^k; \theta^{(i)})}{\partial \theta} \right] \left[ \text{cov}\{\Delta \theta_k^{(i)}\} \right] \left[ \frac{\partial f(y^k, u^k; \theta^{(i)})}{\partial \theta} \right]^T + \text{cov}\{e_k\}
\]

for \( 1 \leq i \leq M \). In this expression, \( m_k^{(i)} \) and \( \Gamma_k^{(i)} \) are obtained by taking conditional expectation and conditional covariance respectively, over the joint p.d.f. space of \( \{\Delta \theta_k^{(i)}, e_k\} \). Note that if \( \Delta \theta_k^{(i)} \approx 0 \) then \( \Gamma_k^{(i)} \) becomes \( \text{cov}\{e_k\} \).

### 6.2 An Example

In this section, a simple example is shown using the first order linear input-output time series with Gaussian additive noise model. Suppose that the \( i \)-th model can be represented as follows.

\[
y_{k+1} = (A_k^{(i)} + \Delta A_k^{(i)})y_k + (B_k^{(i)} + \Delta B_k^{(i)})u_k + e_k^{(i)}
\]

Here \( y_k = (y_{k,1}, \ldots, y_{k,n})^T \in \mathbb{R}^n \) is a \( n \)-dimensional output vector, \( u_k = (u_{k,1}, \ldots, u_{k,m})^T \in \mathbb{R}^m \) is an \( m \)-dimensional input vector, \( e_k^{(i)} \in \mathbb{R}^n \) is a white noise process with \( p(e_k^{(i)}) = N(0, R_k) \), \( A_k^{(i)}, B_k^{(i)} \) are the nominal parameter matrices, and \( \Delta A_k^{(i)}, \Delta B_k^{(i)} \) are the matrix valued perturbations. As explained, this parametric perturbation is modeled statistically by using the i.i.d. Gaussian distributions. To be more specific, the
following probabilistic representation is obtained.

\[
p(\triangle A_k^{(i)}) = \begin{pmatrix}
p(\triangle A_{1,1,k}^{(i)}) & \cdots & p(\triangle A_{1,n,k}^{(i)}) \\
\vdots & \ddots & \vdots \\
p(\triangle A_{n,1,k}^{(i)}) & \cdots & p(\triangle A_{n,n,k}^{(i)})
\end{pmatrix}
= \begin{pmatrix}
N(0, S_{1,1}^{(i)}) & \cdots & N(0, S_{1,n}^{(i)}) \\
\vdots & \ddots & \vdots \\
N(0, S_{n,1}^{(i)}) & \cdots & N(0, S_{n,n}^{(i)})
\end{pmatrix}
\]

\[
p(\triangle B_k^{(i)}) = \begin{pmatrix}
p(\triangle B_{1,1,k}^{(i)}) & \cdots & p(\triangle B_{1,m,k}^{(i)}) \\
\vdots & \ddots & \vdots \\
p(\triangle B_{n,1,k}^{(i)}) & \cdots & p(\triangle B_{n,m,k}^{(i)})
\end{pmatrix}
= \begin{pmatrix}
N(0, T_{1,1}^{(i)}) & \cdots & N(0, T_{1,m}^{(i)}) \\
\vdots & \ddots & \vdots \\
N(0, T_{n,1}^{(i)}) & \cdots & N(0, T_{n,m}^{(i)})
\end{pmatrix}
\]

The basic random variables are \(\{\triangle A_k^{(i)}, \triangle B_k^{(i)}, e_k\}_{k=0}^{\infty}\). Thus the linear model can be decomposed into two parts: a deterministic part and an uncertainty part as follows.

\[
y_{k+1} = \overline{A}^{(i)} y_k + \overline{B}^{(i)} u_k + \triangle A_k^{(i)} y_k + \triangle B_k^{(i)} u_k + e_{k+1}^{(i)}
\]

Accordingly, the conditional p.d.f. of \(y_{k+1}\) given \(y^k, u^k\) has a Gaussian distribution with time-varying mean and time-varying covariance as follows.

\[
p(y_{k+1} | y^k, u^k) = N(m_k^{(i)}, \Gamma_k^{(i)})
\]

where

\[
m_k^{(i)} = \overline{A}^{(i)} y_k + \overline{B}^{(i)} u_k
\]

\[
\Gamma_k^{(i)} = \text{diag}\{S_{1,1}^{(i)} \cdots S_{1,n}^{(i)}
\begin{pmatrix}
y_{k,1}^2 \\
\vdots \\
y_{k,n}^2
\end{pmatrix}
+ \begin{pmatrix}
T_{1,1}^{(i)} \cdots T_{1,m}^{(i)} \\
\vdots \\
T_{n,1}^{(i)} \cdots T_{n,m}^{(i)}
\end{pmatrix}
\begin{pmatrix}
u_{k,1}^2 \\
\vdots \\
u_{k,m}^2
\end{pmatrix}\} + R_i
\]

This model requires computing the means and the variances at every sampling instant. Note that the overall statistically represented uncertainties are not stationary since its amount depends on the \(y_k\) and \(u_k\). If such uncertainties are large, then it is reasonable to have much confidence on that single observation, but if such uncertainties are small, then much emphasis should be put to that single observation. This desirable feature is implied in the proposed detection method through the covariance structure \(\Gamma_k^{(i)}\).
6.3 Probing Signal Selection in the Presence of Uncertainty

In this section two questions will be discussed: First, is the probing input signal helpful in detection even if the model is not accurate? Second, if the parametric models are described by Gaussian random perturbation, how can the probing signal be synthesized?

Suppose there are $M$ hypotheses $H_1, H_2, \ldots, H_M$ with corresponding parametric models as in the Eq. 6.2. If the probing signal generation method for binary detection can be performed, then the extension to the $M$-ary detection problem can be performed by the idea shown in Chapter 4.

Suppose the probing signal is to be selected to discriminate the following two hypothesized models $H_i$ and $H_j$.

$$
H_i : y_{k+1} = A^{(i)} y_k + B^{(i)} u_k + e_{k+1}^{(i)} \\
H_j : y_{k+1} = A^{(j)} y_k + B^{(j)} u_k + e_{k+1}^{(j)}
$$

Here $y_k \in \mathbb{R}^n$ is a output vector, $u_k \in \mathbb{R}^m$ is an input vector, and $e_{k+1}^{(i)} \in \mathbb{R}^n$ is a white noise vector with $p(e_{k+1}^{(i)}) = N(0, R_t)$. Suppose at time $k$, the true hypothesis is $H_i$ and its dynamic behavior can be mathematically described by using the following statistically perturbed model.

$$
y_{k+1} = (A^{(i)} + \Delta A^{(i)}) y_k + (B^{(i)} + \Delta B^{(i)}) u_k + e_{k+1}^{(i)} \\
p(e_{k+1}^{(i)}) = N(0, R_{tr})
$$

Our concern is the question whether the probing signal is effective or not, in the presence of unknown uncertainty. The probing signal will enhance the performance of the detector as long as the following increment in Baram's information is positive.

$$
\Delta B_{k+1}^{ij}(\tilde{u}_k; \Delta \theta) = B_{k+1}^{ij}(\tilde{u}_k; \Delta \theta) - B_{k+1}^{ij}(\tilde{u}_k = 0; \Delta \theta) > 0
$$

where $B_{k+1}^{ij}(\tilde{u}_k; \Delta \theta)$ denotes

$$
E\{\log \frac{p(y_{k+1} | y^k, u^{k-1}, \tilde{u}_k + \tilde{u}_k; \theta^{(i)}, H_i)}{p(y_{k+1} | y^k, u^{k-1}, \tilde{u}_k + \tilde{u}_k; \theta^{(j)}, H_j)} | y^k, u^{k-1}, \tilde{u}_k + \tilde{u}_k; \theta^{(i)} + \Delta \theta^{(i)}, H_i\}.
$$
This can be calculated as follows.

\[
B_{k+1}^{i,j}(\bar{u}_k; \Delta \theta) = E\left\{ \log \frac{(2\pi)^{-\frac{d}{2}} |R_i|^{-\frac{1}{2}} \exp(-\frac{1}{2}(y_{k+1} - \hat{y}_{k+1})^T R_i^{-1}(y_{k+1} - \hat{y}_{k+1}))}{(2\pi)^{-\frac{d}{2}} |R_i|^{-\frac{1}{2}} \exp(-\frac{1}{2}(y_{k+1} - \hat{y}_{k+1})^T R_j^{-1}(y_{k+1} - \hat{y}_{k+1}))} |y_k, u^k, \theta^{(i)}, \Delta \theta^{(i)}, H_i \right\}
\]

\[
= \frac{1}{2} \log \frac{|R_j|}{|R_i|} - \frac{1}{2} \left\| (A^{(i)} + \Delta A^{(i)}_k)y_k + (B^{(i)} + \Delta B^{(i)}_k)u_k + e^{(i)}_{k+1} - A^{(i)}y_k - B^{(i)}u_k \right\|_{R_i^{-1}}^2 \left\| y_k, u_k \right\|
\]
\[
+ \frac{1}{2} E \left\{ \left\| (A^{(i)} + \Delta A^{(i)}_k)y_k + (B^{(i)} + \Delta B^{(i)}_k)u_k + e^{(i)}_{k+1} - A^{(j)}y_k - B^{(j)}u_k \right\|_{R_j^{-1}}^2 \left\| y_k, u_k \right\| \right\}
\]

\[
= \frac{1}{2} \log \frac{|R_j|}{|R_i|} - \frac{1}{2} \left\| \Delta A^{(i)}_k y_k + \Delta B^{(i)}_k u_k \right\|_{R_i^{-1}}^2 - \frac{1}{2} E(e^{(i)}_k^T R_i^{-1} e^{(i)}_k)
\]
\[
+ \frac{1}{2} \left\| \Delta A^{(i)}_k y_k + \Delta B^{(i)}_k u_k \right\|_{R_j^{-1}}^2 - \frac{1}{2} E(e^{(i)}_k^T R_j^{-1} e^{(i)}_k)
\]
\[
+ \frac{1}{2} \left\| (A^{(i)} - A^{(j)})y_k + (B^{(i)} - B^{(j)})u_k \right\|_{R_j^{-1}}^2
\]
\[
+ \left( \Delta A^{(i)}_k y_k + \Delta B^{(i)}_k u_k \right)^T R_j^{-1} \left( (A^{(i)} - A^{(j)})y_k + (B^{(i)} - B^{(j)})u_k \right)
\]

Let \( \bar{A}^{i,j} = A^{(i)} - A^{(j)} \) and let \( \bar{B}^{i,j} = B^{(i)} - B^{(j)} \) then the above equation becomes

\[
\frac{1}{2} \log \frac{|R_j|}{|R_i|} + \frac{1}{2} \left\| \Delta A^{(i)}_k y_k + \Delta B^{(i)}_k u_k \right\|_{R_i^{-1}}^2 - \frac{1}{2} \left\{ \text{tr}(R_i^{-1} R_j) - \text{tr}(R_i^{-1} R_i) \right\}
\]
\[
+ \frac{1}{2} \left\| \bar{A}^{i,j} y_k + \bar{B}^{i,j} u_k \right\|_{R_j^{-1}}^2 + \left( \Delta A^{(i)}_k y_k + \Delta B^{(i)}_k u_k \right)^T R_j^{-1} \left( \bar{A}^{i,j} y_k + \bar{B}^{i,j} u_k \right)
\]

where \( u_k = \bar{u}_k + \bar{u}_k \). Without loss of generality, assume that \( \bar{u}_k = 0 \). Then

\[
\Delta B_{k+1}^{i,j}(\bar{u}_k; \Delta \theta)
\]

\[
= B_{k+1}^{i,j}(\bar{u}_k = \bar{u}_k; \Delta \theta) - B_{k+1}^{i,j}(\bar{u}_k = 0; \Delta \theta)
\]
\[
\begin{align*}
&= \frac{1}{2} \left\| \Delta A_k^{(i)} y_k + \Delta B_k^{(i)} \bar{u}_k \right\|_{R_j^{-1}}^2 - \frac{1}{2} \left\| \Delta A_k^{(i)} y_k \right\|_{R_j^{-1}}^2 \\
&\quad + \frac{1}{2} \left\| \tilde{A}^i_j y_k + \tilde{B}^i_j \bar{u}_k \right\|_{R_j^{-1}}^2 - \frac{1}{2} \left\| \tilde{A}^i_j y_k \right\|_{R_j^{-1}}^2 \\
&\quad + \left( \Delta A_k^{(i)} y_k + \Delta B_k^{(i)} \bar{u}_k \right)^T R_j^{-1} \left( \tilde{A}^i_j y_k + \tilde{B}^i_j \bar{u}_k \right) - \left( \Delta A_k^{(i)} y_k \right)^T R_j^{-1} \left( \tilde{A}^i_j y_k \right)
\end{align*}
\]

This quantity must be positive in order for the probing signal to be effective. This probing signal was designed without the consideration of the modeling errors. The exact derivation for the conditions of \( \Delta A_k^{(i)}, \Delta B_k^{(i)} \) to guarantee \( \Delta B_{k+1}^{i,j}(\bar{u}_k; \Delta \theta) > 0 \) will not follow. One may ask whether any \( \bar{u}_k \) can be helpful for detection. (In other words whether any \( \bar{u}_k \) increases \( \Delta B_{k+1}^{i,j} \)) The answer is no. This can be easily seen even if there is no uncertainty and can be verified by letting \( \Delta A_k^{(i)} = 0, \Delta B_k^{(i)} = 0 \). Thus,

\[
\left[ \Delta B_{k+1}^{i,j} \right]_{\Delta A_k^{(i)} = 0, \Delta B_k^{(i)} = 0} = \frac{1}{2} \left\| \tilde{A}^i_j y_k + \tilde{B}^i_j \bar{u}_k \right\|_{R_j^{-1}}^2 - \frac{1}{2} \left\| \tilde{A}^i_j y_k \right\|_{R_j^{-1}}^2
\]

where the sign can be positive or negative depending upon \( \bar{u}_k \). Hence we can deduce that an arbitrary signal may decrease the detection performance even if there is no uncertainty. If there are some uncertainties then this chance will increase. On the other hand, if there is no modeling uncertainty then the probing signal which was `optimally' designed, in the sense of relative entropy, will always increase the separation performance. If there are some uncertainties then such benefits may be or may not be expected depending upon the amount of modeling uncertainties.

Next, the generation of the probing signal knowing the statistically represented modeling uncertainties is concerned. If the covariance of the statistical perturbation is known, then the probing signal can be designed effectively. Suppose at time step \( k \), we have two p.d.f. profiles.

\[
\begin{align*}
H_i : \quad &p(y_{k+1} \mid y^k, u^k, H_i) = N(m_k^{(i)}, \Gamma_k^{(i)}) \\
H_j : \quad &p(y_{k+1} \mid y^k, u^k, H_j) = N(m_k^{(j)}, \Gamma_k^{(j)})
\end{align*}
\]

Here, both \( \Gamma_k^{(i)} \) and \( \Gamma_k^{(j)} \) can be time-varying. Consider the following performance
function as in the Chapter 4.

$$\tilde{u}_k = \arg \max_{u_k} J_{k+1}^{i,j}(\bar{u}_k + \tilde{u}_k) = \frac{1}{2} \left\{ D_{k+1}^{i,j}(u_k) + D_{k+1}^{i,i}(u_k) \right\}$$

$$s.t \quad |\tilde{u}_k| \leq \tilde{u}_{k,\text{max}}$$

where

$$D_{k+1}^{i,j}(u_k) = D\{p(y_{k+1}|y^k, u^k, H_i) \parallel p(y_{k+1}|y^k, u^k, H_j)\}$$

The gradient of the performance function was

$$\nabla_{u_k} J_{k+1}^{i,j}(\bar{u}_k) = \frac{1}{2} (B^{(i)} - B^{(j)})^T (R_{i}^{-1} + R_{j}^{-1}) \left\{ (A^{(i)} - A^{(j)})y_k + (B^{(i)} - B^{(j)})\bar{u}_k \right\}.$$

The optimal probing signal is easily determined by replacing the covariance portion with the new ones (i.e. $R_i \rightarrow \Gamma_k^{(i)}$ and $R_j \rightarrow \Gamma_k^{(j)}$) which are obtained at every instant of time. Thus the corresponding gradient vector at $\bar{u}_k$ is

$$\nabla_{u_k} J_{k+1}^{i,j}(\bar{u}_k) = \frac{1}{2} (B^{(i)} - B^{(j)})^T (\Gamma_k^{(i)} + \Gamma_k^{(j)})^{-1} \left\{ (A^{(i)} - A^{(j)})y_k + (B^{(i)} - B^{(j)})\bar{u}_k \right\}.$$
Chapter 7

Detection Network

We have observed how the detection task is performed among linearly hypothesized models with assumed Gaussian distributions. However in practical situations, these models might not be representative of the stochastic dynamic behavior being modeled. For some detection tasks, linearized models may give reasonable performances, but if an improved performance is needed, then there is an alternative approach. The detection structure can be developed without changing its basic working mechanism by replacing each or some of the simplified modules (i.e., dynamic prediction model and probability density function model) with improved ones. This detection mechanism will be introduced and defined as a Detection Network in the first section.

Basically, the Detection Network is a natural extension of information based detection. By detecting over the observation space, various types of models can be incorporated into its modules. Each hypothesized model can be a combination of different prediction models and probability density function (p.d.f.) models. This is a great advantage because it adds flexibility to the design and implementation of the network.

Prediction modules can be constructed by curve fitting (or surface fitting) methods with delayed coordinates. There are two types of fitting methods: parametric and nonparametric. In the parametric method, a functional structure with some unknown parameters is defined and such parameters are tuned by minimizing
some appropriate penalty function (e.g. LMS error criteria). In the nonparametric method, a kernel function is defined, applied to each sample, and interpolated. Although the nonparametric method has been widely used and has shown good performance, the parametric method will be preferred in the present work because of the need to calculate the optimal probing signal and this would require heavy online computations. (Remember that the computation of the Jacobians was necessary in real time). There are many parametric methods in universal approximators. But the following Gaussian Bases Function Networks (GBFNs) will be used.

\[ f(x) = \sum_{i=1}^{L} a_i \cdot \exp \left( -\frac{1}{2} \| x - m_i \|_{R_i^{-1}}^2 \right) \]  \hspace{1cm} (7.1)

Here \( x \) can be a scalar or a vector. This is a linear superposition of the Gauss Base Function (GBF). Each kernel’s center position \( m_l \) (\( l = 1, 2, \ldots, L \)) can be fixed (e.g. by uniformly fixing over the input domain) or can be varied depending on designer’s choice. The covariances \( R_l \) can be diversified by constraining the structure. (e.g. in multivariate case, each \( R_l \) can be independent ellipses or can be circle with same size.) This type of functional is attractive because (1) there have been many reports about its outstanding performance and, (2) each node in the network has compact support for which the output is greater than some small value \( \epsilon > 0 \). The compact support is a desirable feature since it leads to fast parameter identification procedures and is very efficient in capturing local variations in the response surface. More will be discussed in Section 2.

There are two types of approaches for p.d.f. estimation: parametric methods and nonparametric methods. In parametric p.d.f. estimation, a functional structure with some unknown parameters are defined and all parameters are identified by maximizing some appropriate performance function (e.g., Likelihood criteria). In kernel based nonparametric methods, a kernel function is defined, applied to each sample, and superimposed. Even though the nonparametric density estimation method is widely used and shows good performance, the parametric method will be preferred because less computational effort is required to calculate the optimal probing signal. Particularly GBFNs will be preferred as p.d.f. estimators in the present work for the similar reasons considered in the regression model. A linear sum of convex basis func-
tion is able to approximate any class of bounded function within arbitrary accuracy if there are many samples and an appropriate network size is chosen. In the GBFNs, the Maximum Likelihood (ML) method does not yield a closed form of solution since it provides a complicated simultaneous nonlinear equations that should be solved. To overcome this technical problem, the Expectation Maximization (EM) method will be used. The EM algorithm has been used to various applications including the Mixed Gaussian Density estimation problem. It will be used for more diverse applications in the density estimation problem in this thesis.

The attractiveness in applying a GBF is that highly complex functions are closely approximated by compositions of smooth less complicated functions. In Gaussian family, there are many classes of base functions that depends on how much a covariance structure is constrained. The most generalized GBFN (i.e a GBFN where covariances are independent ellipses) can represent any sub classes of it, but the result of parameter identification is not the same. Two reasons are, (1) the parameter space, over which the penalty function is to be minimized (or performance function is to be maximized), is non convex nonlinear, and (2) if the optimal parameters are positioned in the edge of the space, then the chance to access it through a gradient based search is very low. Furthermore, curve fittings (or surface fitting) and density estimations shows poor performance in generalization capabilities when too many parameters are used. Cramer-Rao type bounds are obtainable only when we have a large number of samples. Also, since the samples are usually contaminated with noise, over-parameterization often results in a problem. Basically, our aim is to avoid over-parameterization and under-parameterization problems. One may ask how to choose the number of independent parameters in a given structure. One may also want to know which effective measures for comparing the representational potential between models should be used. These difficult questions will be partially answered in this chapter. If a large number of samples are available in every anomaly mode (or any dynamic class), then the Cross Validation method can be used in both the regression problem and the density estimation problem, but this is not likely. In reducing the independent parameters, there are two ways: One approach is to directly reduce the number of parameters in a given structure and the other is to replace the
base function with less complex ones. (In GBFN this can be controlled this by giving
some constraints to the covariance matrix). For this reason, the GBFN approach
to the curve fitting problem will be discussed in Section 2. Also, various class of
reduced complexity GBFNs for p.d.f. estimation will be defined, and their parameter
tuning procedure will be derived by using the Maximum Likelihood (ML) method
incorporated with the Expectation Maximization (EM) algorithm in Section 3. In
this chapter, a model selection method will also be presented based on the Minimum
Description Length approach.

In the context of statistic information theory, a probing signal has been
shown to be is helpful since Chapter 4 for active detection. This signal can be de-
termined in a Detection Network. A basic idea for such probing signal selection is to
find and supply the signal that maximizes the conditional relative entropy of the two
most plausible models’ future output distributions. A simple example for probing
signal generation for a binary detection problem will be presented in the final section
in this chapter. Extension to the \(M\)-ary detection problem is achieved by using the
algorithm as developed in Chapter 4.

The main contribution in this chapter is the development of the Detection
Network. If the residual error distribution is not an independent sequence, (i.e. corre-
lated with the independent variables), then the detection network certainly will show
limitations because it was designed assuming the i.i.d. condition in the residual error
distributions. In this case we can define a likelihood function based on the joint p.d.f.
of the residual process and the independent variable’s process. This will increase
the input dimension of the p.d.f. and require more samples to perform identification
process. This is certainly possible and feasible for low dimension problems. However
this is not studied in the present work.

7.1 Detection Network

The Detection Network is defined as follows.
**Definition 1:** A Detection Network is a hybrid information processing structure composed of two parts: (1) a Generalized Posterior Network where posterior probabilities for each hypothesized models are calculated. (2) a Sequential Decision Module which is composed of a stopping rule and a terminal decision rule.

Figure 7.1 shows a schematic diagram of the Detection Network. The proposed Detection Network requires the Generalized Posterior Network and this is defined as follows.

![Diagram of Detection Network](image)

Figure 7.1: Detection Network
Definition 2: Suppose that there are $M$ hypothesized models $H_1, H_2, \ldots, H_M$. Let $y_k \in \mathbb{R}^{d_1}$, $u_k \in \mathbb{R}^{d_2}$. Denote $y^k = \{y_k, y_{0}\}$, $u^k = \{u_k, u_{0}\}$, $y^{k,k-n} = \{y_k, \ldots, y_{k-n}\}$, and $u^{k,k-m} = \{u_k, \ldots, u_{k-m}\}$. A Generalized Posterior Network $\mathcal{N}$, a distributed information processing structure, is a MIMO discrete time real valued function $\mathcal{N}: \{y_k, u_k\} \rightarrow p(H_i \mid y^k, u^{k-1})$, i.e. $\mathbb{R}^{d_1+d_2} \rightarrow \mathbb{R}^1$. consisting of pairs of local information processing modules $\{\mathcal{P}_i, \mathcal{Q}_i\}_{i=1}^M$. Here, a prediction module (or a sub network) $\mathcal{P}_i : \mathbb{R}^{(n+1)d_1+(m+1)d_2} \rightarrow \mathbb{R}^{d_1}$ computes $\tilde{y}_k^{(i)} = E\{y_k \mid y^{k-1}, u^{k-1}, H_i\}$, and a density estimation module (or a sub network) $\mathcal{Q}_i : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^1$ computes $p(y_k - \tilde{y}_k^{(i)} \mid H_i)$. Delay units and a posterior computation module are combined such that, given an observation $y_k$ the following posterior probability is recursively obtained

$$p(H_i \mid y^k, u^{k-1}) = \frac{L_{k}^{(i)} p(H_i \mid y^{k-1}, u^{k-2})}{\sum_{j=1}^{M} L_{k}^{(j)} p(H_j \mid y^{k-1}, u^{k-2})}$$

where $L_{k}^{(i)} = \mathcal{Q}_i(y_k - \mathcal{P}_i(y^{k-1,k-n-1}, u^{k-1,k-m-1}))$

for each $i = 1..M$ and each $k = 1, 2, \ldots$

Figure 7.2 shows a schematic diagram for the Generalized Posterior Network.

In the Detection Network, each prediction module $\mathcal{P}$ can be either a linear model or a nonlinear model, and each density estimation module $\mathcal{Q}$ can be a simple Gaussian model or an other p.d.f. model. How well this detection network performs depends on how well each of the module does its job.

Suppose the $M$ hypothesized models are

$$H_1 : \quad y_{k+1}^{(1)} = f^{(1)}(y^k, u^k) + n_k^{(1)}$$
$$\vdots$$
$$H_M : \quad y_{k+1}^{(M)} = f^{(M)}(y^k, u^k) + n_k^{(2)}$$

where $\{n_k\}$ is an i.i.d. process. In the detection network, the predictive module computes $E\{y_{k+1}^{(i)} \mid y^k, u^k\} = f^{(i)}(y^k, u^k)$ but there will be some modeling error. say $\epsilon(y^k, u^k)$. Hence $y_{k+1}^{(i)} - \mathcal{P}_i(I_{k,n,m}) = \epsilon^{(i)}(y^k, u^k) + n_k^{(i)}$. Strictly speaking, the modeling
error is not an independent process since it may depend on the past outputs and inputs. If such modeling error is dominated by i.i.d. noise process \( n_k^{(i)} \), then we can approximate it as another i.i.d. process \( \{e_k\} \).

7.2 Prediction Module with GBFN

To obtain the predictive module in the function fitting approach, the number of required independent variables in the set of past outputs and inputs \( \{y_k, \ldots, y_0, u_k, \ldots, u_0\} \) should be identified. This is a difficult problem which is called ‘model order selection problem’ in identifications or the ‘input dimension selection (or reduction) problem’ in neural networks. If too few independent variables are selected then it may result in a large error variance and the i.i.d. property is destroyed. On the other hand, if too many independent variables are selected then the increased input dimension will demand a lot of training samples for parameter tuning. In the linear model identification problem, the model order selection problem and the model complexity selection problem come together and the solution is the same because there is a one-to-one correspondence between input variables and parameters. But generally, this is not here. This input dimension selection/reduction problem will not be addressed in this work. Once the class of independent variable is chosen, the delaying operators are applied to obtain the necessary delayed coordinates. For more details see (Narendra, 90).

A network can express functions of several variables as a composition of basic functions. In order to approximate arbitrary functions by using a composition of basic functions, the network should be a universal approximator in the sense that for any such function \( f^* \) there exists a sequence of network functions \( f_n \) that converges uniformly to \( f^* \). Two important classes are the Multi Layered Perceptron (MLP) and the Radial Bases Function (RBF) networks.

The first one is the class of two layer network with an unrestricted number of sigmoidal units in the first layer and a linear unit in the second layer. The functional
form of this network is
\[ \sum_j a_j h(\sum_k w_{jk} x_k + w_{j0}) \]
where \(a, w\) are parameters to be determined and \(h(x)\) is the sigmoid function. (e.g. \(h(x) = \tanh(x)\)) The universal approximation property was proved by Cybenko (89).

The second one is the class of two layer networks with Gaussian type function units in the first layer and a linear unit in the second layer. The functional form is
\[ \sum_j a_j g(x; \theta_j) \]
where \(a, \theta\) are parameters to be determined and \(g(x; \theta_j)\) is a smooth convex function (e.g., the Gaussian function).

These two functions are tuned to approximate the desired function by adjusting the parameters. To adjust the parameters, first a penalty function is set up which represents the functional distance and such penalty function is minimized over the admissible parameters space. The most popularly used cost function is the following LMS criterion.
\[ \frac{1}{2N} \sum_{k=1}^N \| t_k - y_k \|_W^2 \]
Here \(t_k, y_k\) are the training sample vectors and network output vectors respectively. and \(N\) is the number of training samples. The Euclidean norm defines the metric. A weighting matrix \(W\) can be involved and this is determined by designers. Certainly this type of convex penalty function has been extensively used. In particular, for problems where the underlying structure is linear, a convex quadratic optimization problem is obtained and thus local optima ensures the global optima.

As a way to minimize the criteria function over the parameter space, some numerical search algorithms can be applied. There are two types of methods. Methods that use gradient information only (e.g., the Steepest Descent Method or some Adaptive Method) and methods that use gradients and second order derivatives as well (e.g., the Newton Method). The Conjugate gradient method can be applied with or without second order gradient information. (The Fletcher-Reeves method and the Polak-Ribiere method avoid the need of second order gradient but it is computationally more expensive than the usual first order methods.) Generally, second order
gradient methods are computationally expensive, but once computed they make the convergence to the goal much faster than the first order gradient based methods.

If the criteria is a nonconvex function with respect to the parameter space then the algorithm can terminate at a local minimum. However such local minima are not the same. If the parameters space is large and the criteria function is highly nonlinear, then the chance to fall in an unacceptable local minimum increases. Hence, unless the criteria function is convex with respect to parameters, the search process should be tried several times with different initial condition.

Although the model tuned by minimizing the above criteria has shown acceptable performances, some problems have been identified. The over-fitting problem is unfavorable since it basically limits the generalization performances of the Neural Network. Over-fitting or under-fitting can often occur, because the training samples are corrupted by noise, and one does not know how many parameters are required. If the parameters are too few then under-fitting can happen and, if parameters are too many then the redundant parameters tend to be adjusted by the noise.

Generally, there are two ways in the MLP: the search methods and the complex penalty function methods. In the search methods, the correct structure was obtained by changing the network structure (i.e., checking the activities of nodes and pruning unnecessary ones), or by generating nodes as required. Structural changes stop when an "appropriate" structure is obtained. The Optimal Brain Surgeon (Hassibi et al. 93, 94), Optimal Brain Damage (Le Cun et al. 90) and Pruning using Principal Components (Levin et al. 94) are all pruning methods. The basic common idea of the pruning approach is the usage of the local curvature in the error surface to make a trade-off between the network complexity and the training performance\(^1\). On the other hand, the Growing Cell Structure (Fritzke 94) belongs to the node generation approach. Although these methods have shown a systematic approach, they

\(^1\)Basically, pruning the nodes can be understood as perturbations in the weight parameters. A local model of the error surface is constructed by applying a Taylor series at the operating point. By this way, the information about the possible pruning is obtained by the local curvature of the error surface, for analytically predicting the effect of such perturbations in synaptic weights. The Hessian matrix is assumed to be diagonal in the OBD method, and is assumed to be real symmetric in the OBS method. These methods are studied for MLP.
can not determine ‘when to stop the pruning or generalization’.

Another way is to use the complexity penalty function. (Hinton. 87)

\[ \text{Total Cost} = \text{Fitting Error Cost} + \lambda \cdot \text{Complexity Cost} \]
\[ = \frac{1}{2N} \sum_{k=1}^{N} \| t_k - y_k \|^2_W + \lambda \cdot \sum_i w_i^2 \]

The basic mechanism is that the above complexity criterion forces the distribution of parameters to be of a Gaussian type centered at zero. Since many parameters are close to zero (or tied around zero), the effect of the number of parameters decreases. Nowlan proposed a complexity cost term based on the multi-modal distributions and made this distribution be adjustable. Once the dependencies among the parameters occur, then the network loses certain degrees of freedom and flexibilities, in other words the complexity is reduced compared to the fully independent parameter case. For this reason, the number of parameters is not an adequate measure of the complexity and the determination of the effective number of parameters is needed. (Moody. 92. 94).

Even though the complexity penalty term is used, specifying the trade-off parameter (or weight) \( \lambda \) is not an easy problem. Trial and error based methods show that the minimization is not so sensitive to the parameter \( \lambda \). However, if \( \lambda \) is too large then the under-fitting phenomenon can occur, and if \( \lambda \) is too small then the over-fitting phenomenon can occur. Many approaches have been studied for the systematic selection of \( \lambda \) (i.e., Bayesian approach (Mackey. 92) and Minimum Description Principle approach (Nowlan 91; Nowlan and Hinton 92)).

In RBFN, similar approaches have been studied in both nonparametric and parametric kernel regression. (Poggio and Girosi, 90). A stabilizer \( P \) (i.e., \( \| Pf \|^2 = \sum_{k=0}^{\infty} a_k \| D^k f(x) \|^2 \) for some \( a_k \)) can be used to obtain a good regularized interpolation. For example, if one chooses \( a_k = \sigma^{2k}/(k!2^k) \) then the Gaussian type kernel regression minimizes the following interpolation function.

\[ \frac{1}{2N} \sum_{k=1}^{N} \| t_k - f(x_k) \|^2_W + \frac{1}{2} \| Pf \|^2 \]

Even though the MLP and the RBFN can perform the same task, their working behavior is quite different. Since the sigmoid function does not have compact
support, weight adjustments to each node cause global changes in the functional mapping. This may be an unfavorable thing since local errors can produce a global change. In Gaussian networks each node has compact support for which the output is greater than some value $\epsilon > 0$.

Several researchers have investigated Gaussian Bases Function Network (Broomhead and Lowe, 88; Moody and Darken, 89; Poggio and Girosi, 90; Ghahramani and Jordan 94). However a fundamental question concerning how to choose the basis function has not been answered. One may want to know whether a composition of many `simple' functional elements is better than a composition of relatively small number of complex functional elements. For example, the Gaussian base function with arbitrary covariance matrices will have more degrees of freedom than those with diagonal covariance matrices. The functional complexity can be controlled by the selection of the basis functions as well as by the numbers of kernels. How one can choose the basis functions and how many kernels are necessary are certainly difficult questions that have not been satisfactorily answered, and are important because they directly effect the generalization performance of the network.

In this chapter, parametric regression functions based on various Gaussian kernels are defined and their tuning method is shown. The optimal class of basis function and the optimal number of parameters will be problem dependent. However, given any two tuned GBFNs, the preference should be possible. The Cross Validation test method will be used in simulations in Chapter 8.

The parametric GBFN can be represented as follows.

$$y_k = \sum_{i=1}^{M} a_i G(\|x_k - m_i\|^2_{\Sigma^{-1}})$$

$$G(\|x_k - m_i\|^2_{\Sigma^{-1}}) = \exp\left\{ -\frac{1}{2} (x_k - m_i)^T \Sigma^{-1} (x_k - m_i) \right\}$$

where $M$ is number of kernels employed in the prediction module, $x_k \in R^d$ and $y_k \in R^1$. Let $\Phi$ denote the set of independent parameters. The family to be considered are as follows.
Family 1:  (Fixed Centers with the same Circles)
\[ a_1, ..., a_M \text{ are parameters} \]
\[ m_1, ..., m_M \text{ are fixed} \]
\[ \Sigma_{l-1} = \cdot \Sigma_{M-1} = vI \quad (l = 1..M) \text{ is a parameter} \]

Family 2:  (Variable Centers with Independent Circles)
\[ a_1, ..., a_M \text{ are parameters} \]
\[ m_1, ..., m_M \text{ are parameters} \]
\[ \Sigma_{l-1} = v_l I \quad (l = 1..M) \text{ are parameters} \]

Family 3:  (Variable Centers with Independent Orthogonal Ellipses)
\[ a_1, ..., a_M \text{ are parameters} \]
\[ m_1, ..., m_M \text{ are parameters} \]
\[ \Sigma_{l-1} = \text{diag}(v_{l1}, ..., v_{ld}) \quad (l = 1..M) \text{ are parameters} \]

Family 4:  (Variable Centers with Independent General Ellipses)
\[ a_1, ..., a_M \text{ are parameters} \]
\[ m_1, ..., m_M \text{ are parameters} \]
\[ \Sigma_{l-1} = \text{real symmetric matrix} \quad (l = 1..M) \text{ are parameters} \]

The following LMS criterion function is considered and the first order gradient based numerical parameter search method is used.

\[ E = \frac{1}{2} \sum_{k \in C} \|t_k - y_k\|^2 \]

Here \( C \) is the class set to be considered. Then knowing the gradient of error surface with respect to the parameters is sufficient for the search procedure. For example if the Steepest Descent method is used then the parameter update law is

\[ \hat{\theta}_{t+1} = \hat{\theta}_t - \eta \frac{\partial E(\hat{\theta}_t)}{\partial \theta} \]
The following lemma can be used for the gradient information for each class given above. Most of the results have been previously derived by other authors.

**Lemma 4**: Suppose $E = (1/2) \sum_{k \in C} (t_k - y_k)^2$. Let

$$\Sigma_l^{-1} = \begin{pmatrix} v_{11} & \cdots & v_{1d} \\ \vdots & \ddots & \vdots \\ v_{dl} & \cdots & v_{dd} \end{pmatrix} \quad \text{and} \quad \frac{\partial E}{\partial \Sigma_l^{-1}} = \begin{pmatrix} \frac{\partial E}{\partial v_{11}} & \cdots & \frac{\partial E}{\partial v_{1d}} \\ \vdots & \ddots & \vdots \\ \frac{\partial E}{\partial v_{dl}} & \cdots & \frac{\partial E}{\partial v_{dd}} \end{pmatrix}$$

where $\Sigma_l^{-1} = (\Sigma_l^{-1})^T$. For each family defined in 7.2, the gradients with respect to the corresponding independent parameters are:

**Family 1**: (Fixed Centers with same Circles)

$$\frac{\partial E}{\partial a_{l}} = - \sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|^2_{\Sigma_l^{-1}})$$

$$\frac{\partial E}{\partial \Sigma_l^{-1}} = \frac{a_l}{2} \sum_{k \in C} \left\{ (t_k - y_k) \sum_{i=1}^{L(i)} G(\|x_k - m_i\|^2_{\Sigma_l^{-1}}) \|x_k - m_l\|^2 \right\} l$$

**Family 2**: (Variable Centers with Independent Circles)

$$\frac{\partial E}{\partial a_{l}} = - \sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|^2_{\Sigma_l^{-1}})$$

$$\frac{\partial E}{\partial m_{l}} = -a_l \sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \Sigma_l^{-1}(x_k - m_l^{(i)})$$

$$\frac{\partial E}{\partial \Sigma_l^{-1}} = \frac{a_l}{2} \sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \|x_k - m_l\|^2 l$$

**Family 3**: (Variable Centers with Independent Orthogonal Ellipses)

$$\frac{\partial E}{\partial a_{l}} \text{ and } \frac{\partial E}{\partial m_{l}} \text{ are the same as those of Class2}$$

$$\frac{\partial E}{\partial \Sigma_l^{-1}} = \frac{a_l}{2} \sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \cdot \Lambda(x_k - m_l)$$

where $\Lambda(\theta)$ denotes a matrix such that

$$\Lambda(\theta)_{ij} = \delta_{ij} \theta_i \theta_j \text{ for } \theta = (\theta_1, ..., \theta_d)^T$$
Family 4: (Variable Centers with Independent General Ellipses)

\[
\frac{\partial E}{\partial \Sigma_l} = \frac{\partial E}{\partial m_l} = \frac{1}{2} \sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|_{\Sigma_l}^2) \left\{ (x_k - m_l)(x_k - m_l)^T \right\}
\]

Proof. : Family 1: (Fixed Centers with same Circles)

In this class, \( \Phi = \{a_1, a_2, \ldots, a_M, \Sigma^{-1}\} \) where \( \Sigma^{-1} = vI \) for some constant \( v \).

Therefore

\[
y_k = \sum_{l=1}^{M} a_l G(\|x_k - m_l\|_{\Sigma^{-1}}^2)
= \sum_{l=1}^{M} a_l \exp \left\{ -\frac{1}{2} (x_k - m_l)^T (vI)(x_k - m_l) \right\}
\tag{7.3}
\]

By differentiating Eq. 7.3 with respect to \( a_l \), we obtain \( \frac{\partial y_k}{\partial a_l} = G(\|x_k - m_l\|_{\Sigma^{-1}}^2) \).

Thus

\[
\frac{\partial E}{\partial a_l} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial a_l} = -\sum_{k \in C} (t_k - y_k) G(\|x_k - m_l\|_{\Sigma^{-1}}^2)
\]

By differentiating Eq. 7.3 with respect to \( v \), we get

\[
\frac{\partial y_k}{\partial v} = -\sum_{l=1}^{M} a_l G(\|x_k - m_l\|_{\Sigma^{-1}}^2) \left\{ \frac{1}{2} (x_k - m_l)^T (x_k - m_l) \right\}
\]

\[
\Rightarrow \frac{\partial y_k}{\partial \Sigma^{-1}} = \frac{\partial y_k}{\partial v} I
= -\frac{1}{2} \sum_{l=1}^{M} a_l G(\|x_k - m_l\|_{\Sigma^{-1}}^2) \| x_k - m_l \|^2 I
\]

Hence

\[
\frac{\partial E}{\partial \Sigma^{-1}} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \Sigma^{-1}} = -\sum_{l=1}^{M} (t_k - y_k) \frac{\partial y_k}{\partial \Sigma^{-1}}
= \frac{1}{2} \sum_{k \in C} \left\{ (t_k - y_k) \sum_{l=1}^{M} a_l G(\|x_k - m_l\|_{\Sigma^{-1}}^2) \| x_k - m_l \|^2 \right\} I
\]

Family 2: (Variable Centers with Independent Circles)
In this class, \( \Phi = \{ a_1, ..., a_M, m_1, ..., m_M, \Sigma_1^{-1}, ..., \Sigma_M^{-1} \} \), where \( \Sigma_1^{-1}, ..., \Sigma_M^{-1} \) are \( v_1 I, ..., v_M I \) respectively, for some constants \( v_1, ..., v_M \). Therefore

\[
y_k = \sum_{l=1}^{M} a_l G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) = \sum_{l=1}^{M} a_l \exp \left\{ -\frac{1}{2} (x_k - m_l)^T (v_l I) (x_k - m_l) \right\}
\]

(7.4)

By differentiating Eq. 7.4 with respect to \( a_l \) we obtain \( \frac{\partial y_k}{\partial a_l} = G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \).

Thus

\[
\frac{\partial E}{\partial a_l} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial a_l} = - \sum_{k \in C} (t_k - y_k)G(\|x_k - m_l\|^2_{\Sigma_l^{-1}})
\]

By differentiating Eq. 7.4 with respect to \( m_l \) and using the fact that \( \frac{d}{d\theta} (\theta^T A \theta) = 2A\theta \) we obtain \( \frac{\partial y_k}{\partial m_l} = a_l G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \Sigma_l^{-1} (x_k - m_l) \). Thus

\[
\frac{\partial E}{\partial m_l} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial m_l} = -a_l \sum_{k \in C} (t_k - y_k)G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \Sigma_l^{-1} (x_k - m_l)
\]

Also by differentiating Eq. 7.4 with respect to \( v_l \), we obtain

\[
\frac{\partial y_k}{\partial v_l} = -a_l G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \left\{ \frac{1}{2} (x_k - m_l)^T (x_k - m_l) \right\}
\]

\[
\Rightarrow \frac{\partial y_k}{\partial \Sigma_l^{-1}} = -\frac{a_l}{2} G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \| x_k - m_l \|^2 I
\]

hence we have

\[
\frac{\partial E}{\partial \Sigma_l^{-1}} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \Sigma_l^{-1}} = - \sum_{k \in C} (t_k - y_k) \frac{\partial y_k}{\partial \Sigma_l^{-1}}
\]

\[
= \frac{a_l}{2} \left\{ \sum_{k \in C} (t_k - y_k)G(\|x_k - m_l\|^2_{\Sigma_l^{-1}}) \| x_k - m_l \|^2 \right\} I
\]

Family 3: (Variable Centers with Independent Orthogonal Ellipses)

In this class, \( \Phi = \{ a_1, ..., a_M, m_1, ..., m_M, \Sigma_1^{-1}, ..., \Sigma_M^{-1} \} \) where \( \Sigma_1^{-1}, ..., \Sigma_M^{-1} \) are diagonal matrices. Hence

\[
y_k = \sum_{l=1}^{M} a_l G(\|x_k - m_l\|^2_{\Sigma_l^{-1}})
\]

(7.5)

\( \partial E/\partial a_l \) and \( \partial E/\partial m_l \) are obtained by the same procedure as in Family 2. By differentiating Eq. 7.5 with respect to \( \Sigma_l^{-1} \) and using the fact that \( \frac{d}{d\theta} (\theta^T D \theta) = \Lambda(\theta) \).
(where \(\Lambda(\theta)\) denotes a diagonal matrix such that \([\Lambda(\theta)]_{ij} = \delta_{ij}\theta_i\theta_j\) for \(\theta = (\theta_1, \ldots, \theta_d)^T\)). We obtained

\[
\frac{\partial y_k}{\partial \Sigma_i^{-1}} = -\frac{a_i}{2} G(\|x_k - m_i\|_{\Sigma_i^{-1}}^2) \cdot \Lambda(x_k - m_i)
\]

Hence

\[
\frac{\partial E}{\partial \Sigma_i^{-1}} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \Sigma_i^{-1}} = -\sum_{k \in C} (t_k - y_k) \frac{\partial y_k}{\partial \Sigma_i^{-1}}
\]

\[
= \frac{a_i}{2} \sum_{k \in C} (t_k - y_k) \cdot G(\|x_k - m_i\|_{\Sigma_i^{-1}}^2) \cdot \Lambda(x_k - m_i)
\]

**Family 4: (Variable Centers with Independent General Ellipses)**

In this case, \(\Phi = \{a_1, \ldots, a_M, m_1, \ldots, m_M, \Sigma_1^{-1}, \ldots, \Sigma_M^{-1}\}\) and

\[
y_k = \sum_{i=1}^{M} a_i G(\|x_k - m_i\|_{\Sigma_i^{-1}}^2) \tag{7.6}
\]

\(\partial E/\partial a_i\) and \(\partial E/\partial m_i\) are obtained by the same procedure used in Family 2. By differentiating Eq. 7.6 with respect to \(\Sigma_i^{-1}\) and using the fact \(\frac{\partial}{\partial \Lambda}(\theta^T A \theta) = \theta \theta^T\), we obtain

\[
\frac{\partial y_k}{\partial \Sigma_i^{-1}} = -\frac{1}{2} a_i G(\|x_k - m_i\|_{\Sigma_i^{-1}}^2) (x_k - m_i)(x_k - m_i)^T
\]

Thus

\[
\frac{\partial E}{\partial \Sigma_i^{-1}} = \sum_{k \in C} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \Sigma_i^{-1}} = -\sum_{k \in C} (t_k - y_k) \frac{\partial y_k}{\partial \Sigma_i^{-1}}
\]

\[
= \frac{a_i}{2} \sum_{k \in C} (t_k - y_k) G(\|x_k - m_i\|_{\Sigma_i^{-1}}^2) \{ (x_k - m_i)(x_k - m_i)^T \}
\]

If there is a limitation in the computation and the samples are drawn from a highly complexed function corrupted by a large noise, then the search method may not be a good approach. The tuning process with many parameters (say more than 100) requires much computational time when the search method is used. In this case the linear approach (i.e. Family 1 with fixed bandwidth) can be an acceptable approach because it reduces the computational time by using \(y = Ax\) type linear approach. In other words, parameters are obtained by the simple projection over the \(M\) dimensional linear subspace spanned by the fixed Gaussian Basis Functions.
7.3 Density Estimation Module with GBFN

A long time has passed since the parametric and the non-parametric approaches has competed and today it is understood that they compensate each other rather than rival each other. Each method has advantages and disadvantages. One of advantages in parametric p.d.f. estimation is its efficiency, but a clear disadvantage is that the true model class can not be known. A nonparametric kernel based density estimator has the form

\[ \hat{p}(x) = \frac{1}{(nh^d)} \sum_{j=1}^{n} K \left( \frac{x - t_j}{h} \right) \quad x \in \mathbb{R}^d \]

with constraints

\[ K(x) \geq 0, \quad \int_{\mathbb{R}^d} K(x)dx = 1 \]

where \( N \) is the number of training samples. The choice of kernel function \( K \) and the bandwidth \( h > 0 \) determines the performance of the estimator. The advantages in the kernel based non-parametric p.d.f. estimation is that ones are free from the difficulty in model selection. Mathematical consistencies\(^2\) are guaranteed for many classes. Devroye (1983), using the \( L_1 \) approach, proved that the above estimator will be a strongly consistent estimator of true p.d.f. iff \( h_n \to 0 \) and \( nh^d \to \infty \) as \( n \to \infty \). A good summary paper and book about the nonparametric p.d.f. estimation are produced by Izenman (91) and Silverman (86), respectively. There are some disadvantages of the approach. First, since the bandwidth parameter is unknown there is a bias versus variance dilemma\(^3\). Second when samples are not sufficient, the estimated results can be poor.

---

\(^2\)In nonparametric density estimation, consistency is defined in the following way. We form a sequence of regions \( R(1),R(2),... \) containing \( x \), the first region is to be used with one sample, the second one with two, and so on. Let \( V(n) \) be the volume of \( R(n) \), \( k(n) \) be the number of samples falling in \( R(n) \), and \( p(x;n) \) be the n-th estimate for \( p(x) \) such that \( p(x;n)=k(n)/(nV(n)) \). To show that \( p(x;n) \) is to converge to \( p(x) \), three things are required to be shown: (1) \( \lim V(n)=0 \) (2) \( \lim k(n)=\text{infinity} \) (3) \( \lim (k(n))/n=0 \) where \( \lim \) is taken such that \( n \) goes infinity.

\(^3\)The nonparametric method is basically a parameter-free approach. In kernel based methods, which are popularly being used, a kernel function is defined, and applied to each sample. Then the p.d.f. estimator is obtained simply by adding up all the evaluation of the kernels applied. The histogram method, Nearest Neighbor method and various convex kernel methods belong to this class. However a parameter still exists which must be determined or estimated from the data: that is a 'bandwidth'. Choosing the bandwidth is a difficult problem because of the Bias v.s. Variance problem. Specifically, if the size of bandwidth decreases then bias errors decrease but the variance
In our detection problem there are two important factors to be considered. First, given an observation, the computation time for the evaluation of the p.d.f. and its gradients should not be too large since the computation the optimal proving signal is necessary. Second, the p.d.f. surface should be smooth since a gradient computation is required in computing the proving signal. Because of these two factors, the parametric approach is the most appropriate for the detection network.

As a parameter tuning method the Maximum Likelihood (ML) method is used. It is a known fact that the ML method minimizes the relative entropy between the true p.d.f. \( p_{\text{true}}(x) \) and estimated p.d.f. \( p(x \mid \theta) \) if the model class is correctly chosen.

\[
\theta_{\text{opt}} = \min_{\theta} \int p_{\text{true}}(x) \log \frac{p_{\text{true}}(x)}{p(x \mid \theta)} \, dx
= \max_{\theta} \int p_{\text{true}}(x) \log p(x \mid \theta) \, dx
= \max_{\theta} \mathbb{E}_{p_{\text{true}}(x)} \log p(x \mid \theta)
\geq \max_{\theta} \mathcal{L}(\theta)
\]

The true log-likelihood \( \mathcal{L}(\theta) \) can not be calculated because \( p_{\text{true}}(x) \) is unknown. But can be estimated using empirical log-likelihood \( \hat{\mathcal{L}}_n(\theta) \), by which the law of large numbers is given by

\[
\hat{\mathcal{L}}_n(\theta) = \frac{1}{n} \sum_{k=1}^{n} \log p(x_k \mid \theta)
\]

This empirical log likelihood will be used in this work to estimate the necessary errors increase. On the other hand, if the size of bandwidth increases then the variance errors decrease but bias errors increase. This is the dilemma. One can assume that the population has a Gaussian distribution and try to find the optimal bandwidth, but the result may not be good because if the true density is not unimodal then the bandwidth will be too large. Consequently the optimal bandwidth depends on the true p.d.f. which is unknown. To overcome this problem, the Cross Validation method has been popularly used. The basic algorithm is to remove a single sample, say \( x(i) \) from the data, compute the density estimator \( f(i) \) at that \( x(i) \) using the remaining \( n-1 \) samples, and then choose the bandwidth to optimize some criterion (e.g. likelihood or least squares error) involving all estimators \( f(i) \). However, this methods does not give smooth results unless there are sufficiently many samples, otherwise the results are sensitive to outliers. There is an adaptive version where the bandwidth is not a constant but a function of the pre-p.d.f estimation. The basic idea is to make the bandwidth large in regions where samples are sparse and to make it small in regions where the samples are dense. This work was done by Abramson. If we use this method then we can reduce the errors, but we need another estimator for the pre-p.d.f. Although this method is useful, it too can not avoid the trade-off between sufficient number of samples and computation time. If samples are few then the results are too rough, and if the sample are many the computation time is long for our application.
Among parametric approaches the mixed Gaussian p.d.f. is attractive, since Gaussian kernel is familiar to us and it has theoretical background too. It has been known that a linear combination of convex kernels can represent any function within arbitrary small error and has local compact support.

In this section, a learning method for the estimation of p.d.f. in the Mixed Gaussian Density function using EM algorithm is introduced. This EM algorithm, since Dempster et al. (77) first found, has been used for mixed parametric density estimation problem in statistics and incomplete data classification problem in pattern recognitions. The EM algorithm estimates the parameters iteratively, starting from some initial guess. Each iteration consists of an Expectation (E) step, which finds the expectation of the complete-data log likelihood given the values of the observed variables and the current estimate of the parameters, and a Maximization (M) step, which re-estimates the parameters under the assumption that the distribution found in E step is correct. Although it has been referred by many authors, the contents written by mathematicians are too difficult for the engineers and the reviews by the engineers are not concrete and too rough. Motivated by these observations, we derived here again.

Suppose there are observation sequences \( \{y_k\}_{k=1}^N \) collected from a dynamic model which corresponds to some anomaly mode. Assume the prediction module is available and it outputs \( \hat{y}_k \). Denote \( x_k = y_k - \hat{y}_k \) and regard this is another observation. Suppose \( \hat{y}_k = E\{y_k \mid y^{k-1}, u^{k-1}\} \) and our observations \( x^n = \{x_i\}_{i=1}^N \) are i.i.d process and hence stationary and ergodic. The objective is to estimate the density function by using a parametric family particularly the finite number of Gaussian functions. Although our samples are drawn from a single population \( K \), one can assume that the density consists of \( M \) components \( K_1, K_2, ..., K_M \), of which the p.d.fs are \( p_1(x \mid \phi_1), p_2(x \mid \phi_2), ..., p_M(x \mid \phi_M) \) and the mixing proportions are \( a_1, a_2, ..., a_M \). Hence the p.d.f model of the observation \( x_i \in K \) is

\[
q(x_i \mid \Phi) = \sum_{j=1}^{M} a_j p_j(x_i \mid \phi_j)
\]  

(7.7)
under the constraints
\[ \sum_{j=1}^{M} a_j = 1 \quad \text{and} \quad a_j \geq 0 \quad (j = 1, \ldots, M) \]
where \( \phi_j \ (j = 1, \ldots, M) \) is the set of associated parameters for each kernel and \( \Phi = \{a_1, \ldots, a_M, \phi_1, \ldots, \phi_M\} \) is a set of total parameters. Assume that this is a minimal set of the independent parameters to encode the given parametric family. Note the \( a_i \) satisfies the above conditions, and hence if they are known, \( a_i \) can be regarded as the a priori probability that an observation comes from a component \( K_i \). The Gaussian kernels are defined as follows.

\[ p_j(x_i \mid \phi_j) = (2\pi)^{-d/2}(\det \Sigma_j)^{-1/2} \exp \left\{ -\frac{1}{2}(x_i - m_j)^T \Sigma_j^{-1}(x_i - m_j) \right\} \]  

(7.8)

The log-likelihood of the mixed density function is

\[ \mathcal{L}(\Phi \mid x^n) = \sum_{i=1}^{N} \log \sum_{j=1}^{M} a_j p_j(x_i \mid \phi_j) \]  

(7.9)

Then the Maximum Likelihood Estimation leads the following results.

\[ \hat{a}_j = \frac{1}{N} \sum_{i=1}^{N} h_{ij} \]

\[ \hat{m}_j = \frac{\sum_{i=1}^{N} h_{ij} x_i}{h_{ij}} \]

\[ \hat{\Sigma}_j = \frac{\sum_{i=1}^{N} h_{ij} (x_i - \hat{m}_j)(x_i - \hat{m}_j)^T}{\sum_{i=1}^{N} h_{ij}} \]

where

\[ h_{ij} = \frac{a_j \cdot (\det \hat{\Sigma}_j)^{-1/2} \exp \left\{ -\frac{1}{2}(x_i - \hat{m}_j)^T \hat{\Sigma}_j^{-1}(x_i - \hat{m}_j) \right\}}{\sum_{t=1}^{M} a_t \cdot (\det \hat{\Sigma}_t)^{-1/2} \exp \left\{ -\frac{1}{2}(x_i - \hat{m}_t)^T \hat{\Sigma}_t^{-1}(x_i - \hat{m}_t) \right\}} \]

This leads to a set of coupled nonlinear equations which can not be easily solved. Instead of solving these nonlinear coupled equations, the EM algorithm is considered. Note if the mixing proportion are known then this problem becomes very easy. Dempster et al. (77) has proposed the EM algorithm which is capable of handling the missing data problems and Redner et al. (84) applied this to mixed density estimation problem where the mixing proportions were treated as missing data.
Generating an observation $x_j$ can be regarded as a two step procedure. First select one component of the mixture and then draw a sample from the distribution of that component. An indicator variable is introduced by defining an i.i.d RV $z_i = (z_{i1}, z_{i2}, \ldots, z_{iM})^T$ where

$$z_{ij} = \begin{cases} 1 & x_i \in K_j \\ 0 & \text{otherwise} \end{cases} \quad (i = 1..N, \ j = 1..M) \quad (7.10)$$

$z^n = \{z_i\}_{i=1}^n$ has a multinomial distribution consisting of a single draw on $M$ categories with selection probabilities $a_1, a_2, \ldots, a_M$, respectively. Note $\{x^n, z^n\}$ forms a complete data set. The conditional p.d.f. of this indicator variable is

$$q(z_i | \Phi) = \sum_{j=1}^M z_{ij} a_j \quad (7.11)$$

Using the chain rule,

$$q(x_i, z_i | \Phi) = q(z_i | \Phi) \cdot q(x_i | z_i, \Phi)$$

The log-likelihood function for the joint distribution of $x$ and $z$ (this is called 'complete data log likelihood') is defined by

$$L_c(\Phi | x^n, z^n) = \sum_{i=1}^n \log q(x_i, z_i | \Phi) = \sum_{i=1}^n (\log q(z_i | \Phi) + \log q(x_i | z_i, \Phi)) \quad (7.12)$$

The first term in RHS of Eq. 7.12 can be obtained from Eq. 7.11

$$\log q(z_i | \Phi) = \log \sum_{j=1}^M z_{ij} a_j = \sum_{j=1}^M z_{ij} \log a_j \quad (7.13)$$

and since there is only one non-zero term in $z_i = (z_{i1}, z_{i2}, \ldots, z_{iM})$, the second term in RHS of Eq. 7.12 is

$$\log q(x_i | z_i, \Phi) = \sum_{j=1}^M z_{ij} \log p_j(x_i | \phi_j) \quad (7.14)$$

Hence the log likelihood function for the joint distribution (or complete data log likelihood) is obtained by substituting Eq. 7.13 and Eq. 7.14 to Eq. 7.12

$$L_c(\Phi | x^n, z^n) = \sum_{i=1}^N \left( \sum_{j=1}^M z_{ij} \log a_j + \sum_{j=1}^M z_{ij} \log p_j(x_i | \phi_j) \right) = \sum_{j=1}^M \sum_{i=1}^N z_{ij} (\log a_j + \log p_j(x_i | \phi_j))$$
The crucial thing to note is that in the complete data log likelihood, the summation moved to the outside of the log function and this decoupling makes it easy to derive the optimal parameter estimator.

(Dampster, 77) showed that the incomplete data likelihood function \( \mathcal{L}(\Phi) \) can be maximized by iterating the following two steps where the complete data log likelihood function is used:

\[
E \textit{ step} : \ Q(\Phi \mid \Phi_k) = E\{\mathcal{L}_c(\Phi \mid x^N, z^N) \mid x^N, \Phi_k\} \\
M \textit{ step} : \ \Phi_{k+1} = \arg \max_{\Phi} Q(\Phi \mid \Phi_k)
\]

where \( k \) is the iteration step and \( \Phi_k = \{\hat{a}_1^k, ..., \hat{a}_M^k, \hat{\phi}_1^k, ..., \hat{\phi}_N^k\} \) is a set of parameters estimated at \( k \)-th step. The EM algorithm estimates the parameters iteratively, starting from some initial guess \( \Phi_0 \). Each iteration consists of an Expectation (E) step, which finds the expectation of the complete-data log likelihood given the values of the observed variables and the current estimate of the parameters, and a Maximization (M) step, which re-estimates the parameters under the assumption that the distribution found in the E step is correct.

For the E step, the expectation of the complete data log likelihood function conditioned on the \( k \)-th step’s estimate is found as follows.

\[
Q(\Phi \mid \Phi_k) = E\left\{ \sum_{j=1}^M \sum_{i=1}^N z_{ij} \left( \log a_j + \log p_j(x_i \mid \phi_j) \right) \mid x^N, \Phi_k \right\}
= \sum_{j=1}^M \sum_{i=1}^N E\{z_{ij} \mid x_i, \Phi_k\} \left( \log a_j + \log p_j(x_i \mid \phi_j) \right)
= \sum_{j=1}^M \sum_{i=1}^N \hat{h}_{ij}^{k+1} \left( \log a_j + \log p_j(x_i \mid \phi_j) \right)
\]  

(7.15)

Here \( \hat{h}_{ij}^{k+1} = E\{z_{ij} \mid x_i, \Phi_k\} \) and this can be regarded as the posterior probability distribution of a Gaussian component \( j \) generated by a data sample \( x_i \) and conditioned on the estimated parameter \( \Phi_k \). Thus,

\[
\hat{h}_{ij}^{k+1} = \frac{\hat{a}_j^k p_j(x_i \mid \hat{\phi}_j^k)}{\sum_{j=1}^M \hat{a}_j^k p_j(x_i \mid \hat{\phi}_j^k)}
\]  

(7.16)

where \( \hat{a}_j^k \) is regarded as a prior since \( \Phi_k \) is known.

Applying the M step to the result of 7.15 leads to the following decomposed optimization problem.

\[
\max_{\Phi} Q(\Phi \mid \Phi_k)
\]
\[
\begin{align*}
&= \max_{\psi} \left( \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log a_j + \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log p_j(x_i | \phi_j) \right) \\
&= \left\{ \begin{array}{l}
\max_{a_1, \ldots, a_M} \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log a_j \\
st \quad a_1 + a_2 + \ldots + a_M = 1
\end{array} \right\} \quad \text{and} \quad \left\{ \begin{array}{l}
\max_{\phi_1, \ldots, \phi_M} \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log p_j(x_i | \phi_j) \\
\end{array} \right\}
\end{align*}
\] (7.17)

where \( \{\hat{a}_1^{k+1}, \ldots, \hat{a}_M^{k+1}\} \) are determined from the first constrained maximization problem and \( \{\hat{\phi}_1^{k+1}, \ldots, \hat{\phi}_M^{k+1}\} = \{\hat{m}_1^{k+1}, \hat{\Sigma}_1^{k+1}, \ldots, \hat{m}_M^{k+1}, \hat{\Sigma}_M^{k+1}\} \) are determined from the second maximization problem. Since parameters are assumed to be mutually independent. this problem can be further be decomposed into \( M \) independent optimization problems. Hence Eq. 7.17 leads to the following optimization problem.

\[
P_1 : \left\{ \begin{array}{l}
\max_{a_1, \ldots, a_M} \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log a_j \\
st \quad a_1 + a_2 + \ldots + a_M = 1
\end{array} \right\} \quad \text{and} \quad P_2 : \left\{ \begin{array}{l}
\max_{\phi_i} \sum_{i=1}^{N} h_{ii}^{k+1} \log p_i(x_i | \phi_i) \\
\quad \vdots \\
\max_{\phi_M} \sum_{i=1}^{N} h_{iM}^{k+1} \log p_M(x_i | \phi_M)
\end{array} \right\}
\] (7.18)

The \( P_1 \) optimization can be solved using the Lagrangian method

\[
L = \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log a_j + \lambda(1 - a_1 - a_2 - \ldots - a_M)
\]

Note in the first maximization problem \( h_{ij}^{k+1} \) is a known constant, so the equation is linear combination of concave functions. This maximization problem has a unique solution. So the necessary condition \( \partial L / \partial a_j = 0 \) for \( j = 1..M \) yields \( \lambda = N \) and hence

\[
\hat{a}_j^{k+1} = \frac{1}{N} \sum_{i=1}^{N} h_{ij}^{k+1} \quad (j = 1..M)
\] (7.19)

In the \( P_2 \) optimization problem, the necessary condition yields

\[
\sum_{i=1}^{N} \left\{ h_{ij}^{k+1} \frac{\partial}{\partial \phi_j} \log p_j(x_i | \phi_j) \right\} = 0 \quad (j = 1..M)
\] (7.20)

where \( \phi_j = \{m_j, \Sigma_j\} \). Note this optimization problem also has unique solution. In Eq. 7.20, \( \log p_j(x_i | \phi_j) \) is obtained as follows.

\[
\log p_j(x_i | \phi_j) = -\frac{1}{2} \log \det \Sigma_j - \frac{d}{2} \log(2\pi) - \frac{1}{2}(x_i - m_j)^T \Sigma_j^{-1}(x_i - m_j)
\] (7.21)
Thus the differentiation with respect to the parameter \( m_j \) yields

\[
\frac{\partial}{\partial m_j} \log p_j(x_i | \phi_j) = \Sigma_j^{-1}(x_i - m_j) \tag{7.22}
\]

And by using the facts that \((\partial/\partial S) \log \det S = (S^{-1})^T\) (see Appendix A4), \((\partial/\partial S)c^TS^{-1}c = -(c^TS^{-1}c)^T\) (see Appendix A4), and \(\Sigma_j^T = \Sigma_j\), the differentiation with respect to the parameter \( \Sigma_j \) yields

\[
\frac{\partial \log p_j(x_i | \phi_j)}{\partial \Sigma_j} = -\frac{1}{2}\Sigma_j^{-1} + \frac{1}{2}\Sigma_j^{-1}(x_i - m_j)(x_i - m_j)^T \Sigma_j^{-1} \tag{7.23}
\]

Substituting Eq. 7.22 and Eq. 7.23 into Eq. 7.20 and rearranging yields the following results.

\[
\tilde{m}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij}^{k+1}x_i}{\sum_{i=1}^{N} h_{ij}^{k+1}} \quad (j = 1..M) \tag{7.24}
\]

\[
\tilde{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij}^{k+1}(x_i - \tilde{m}_j^{k+1})(x_i - \tilde{m}_j^{k+1})^T}{\sum_{i=1}^{N} h_{ij}^{k+1}} \quad (j = 1..M) \tag{7.25}
\]

where \( h_{ij}^{k+1} \) was defined in Eq. 7.16.

Some remarks are in order\(^4\). First, the incomplete data log-likelihood function is non-decreasing (i.e. \( \mathcal{L}(\Phi_{k+1} | x^n) \geq \mathcal{L}(\Phi_k | x^n) \)), hence if \( \mathcal{L}(\Phi_k | x^n) \) is bounded it converges to some \( \mathcal{L}_* \). This was turned out to be a local minimum or saddle point. However, convergence of \( \mathcal{L}(\Phi_k | x^n) \) to \( \mathcal{L}_* \) does not imply the convergence of \( \Phi_k \) to an unique \( \Phi_* \).

Second, the convergence with the EM algorithm is generally quite slow. It does not directly yield the Fisher Information matrix which can be used to estimate the confidence interval around \( \tilde{\Phi} \). Instead, if one uses the Newton Rapson method, which incorporates information on the Hessian matrix, the speed of the converge can be improved, but it does not guarantee that \( \mathcal{L}(\Phi_k | x^n) \) is decreasing. Also, because the Newton Raphson method requires second order derivatives and inverses, it is computationally expensive.

Third, the performance of the estimation procedure is sensitive to the initial guess \( \Phi_0 \). The sequence of estimator may diverge if \( \Phi_0 \) is chosen too close to the edges on the feasible set. Singular solutions can also exist and the likelihood function

\(^4\)These remarks are well explained and summarized in McLachlan
becomes unbounded. For example, if $\hat{\alpha}_j^k = x_i$ for some $i, j, k$ then $\hat{\Sigma}_j^{k+1}$ in 7.25 is zero and hence inverse dose not exist and the likelihood function is unbounded. There is also the possibility of multiple local minimum. Thus many iterations with different iterations with different choices of initial data should be performed.

**Fourth**, concerning identifiability, the set of parameters are identifiable but not each individual parameter. A density $p(x \mid \theta)$ is said to be identifiable if $\theta_1 \neq \theta_2$ implies that there exists a $x$ such that $p(x \mid \theta_1) \neq p(x \mid \theta_2)$. Therefore to be identifiable, a distinct value of $\theta$ determines a distinct member of the family. Note that in this mixed type problem, the parameters can be permutable without changing the overall density function.

An important thing to note is that if the number of samples is not large, the solution supplied by this Mixed Gaussian Network is often not adequate and is not well representative of the population because it leads to over-parameterization. For each kernel in a $d$-dimensional sample, there are $1 + d + d(d + 1)/2$ parameters involved.

In this sense reducing the number parameters, e.g., using simplified kernels, seems to be an interesting approach and this will be examined in this thesis. In the remainder of this section, the derivation for the unsupervised learning law (still using ML+EM method) for several Gaussian families are performed.

Define the GBFN as follows.

$$q(x_k \mid \Phi) = \sum_{j=1}^{M} a_j p_j(x_k \mid \phi_j)$$

$$p_j(x_k \mid \phi_j) = (2\pi)^{-d/2} (\det \Sigma_j)^{-1/2} \exp \left\{ -\frac{1}{2} (x_k - m_j)^T \Sigma_j^{-1} (x_k - m_j) \right\}$$

where $\phi_j = (m_j, \Sigma_j), x_k \in R^d, \Phi$ is the set of independent parameters, and $M$ is the number of kernels employed. The classes we are going to consider are

---

$^5$M is not number of hypotheses. We use here M for notational convenience.
Family 1: (Fixed Centers with the same Circles)

\[ m_1, ..., m_M \text{ are fixed} \]
\[ a_1, ..., a_M \text{ are parameters} \]
\[ \Sigma_1 = \cdots = \Sigma_M = cI \text{ is a parameter} \]

Family 2: (Variable centers with the same Circles)

\[ m_1, ..., m_M \text{ are parameters} \]
\[ a_1, ..., a_M \text{ are parameters} \]
\[ \Sigma_1 = \cdots = \Sigma_M = cI \text{ is a parameter} \]

Family 3: (Variable centers with the same Ellipses)

\[ m_1, ..., m_M \text{ are parameters} \]
\[ a_1, ..., a_M \text{ are parameters} \]
\[ \Sigma_1 = \cdots = \Sigma_M = \Sigma \text{ is a parameter} \]

Family 4: (Variable centers with independent Circles)

\[ m_1, ..., m_M \text{ are parameters} \]
\[ a_1, ..., a_M \text{ are parameters} \]
\[ \Sigma_j = c_j I_d \quad (j = 1..M) \text{ are parameters} \]

Lemma 5 For each family defined above, the update law using EM algorithms are the following.

Family 1: (Fixed Centers with the same Circles)

\[ \tilde{z}_j^{k+1} \text{ is the same as that of Family 2} \]
\[ \tilde{\Sigma}^{k+1} \text{ is the same as that of Family 2} \]
Family 2: (Variable centers with the same Circles)
\[
\hat{a}^{k+1}_j = \frac{1}{N} \sum_{i=1}^{N} h^{k+1}_{ij} (j = 1..M) \\
\hat{m}^{k+1}_j = \frac{\sum_{i=1}^{N} h^{k+1}_{ij} x_i}{\sum_{i=1}^{N} h^{k+1}_{ij}} (j = 1..M) \\
\hat{\Sigma}^{k+1} = \frac{1}{d} \left( \frac{1}{N} \sum_{i=1}^{N} \|x_i\|^2 - \sum_{j=1}^{M} \hat{a}^{k+1}_j \|\hat{m}^{k+1}_j\|^2 \right) I_d
\]

Family 3: (Variable centers with the same Ellipses)
\[
\hat{a}^{k+1}_j \text{ and } \hat{m}^{k+1}_j \text{ are the same as those of Family 2} \\
\hat{\Sigma}^{k+1} = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T - \sum_{j=1}^{M} \hat{a}^{k+1}_j \hat{m}^{k+1}_j \hat{m}^{k+1}_j^T
\]

Family 4: (Variable centers with independent Circles)
\[
\hat{a}^{k+1}_j \text{ and } \hat{m}^{k+1}_j \text{ are the same as those of Family 2} \\
\hat{\Sigma}^{k+1}_j = \left\{ \frac{\sum_{i=1}^{N} h^{k+1}_{ij} \|x_i - \hat{m}^{k+1}_j\|^2}{d \sum_{i=1}^{N} h^{k+1}_{ij}} \right\} I_d (j = 1..M)
\]

where \[
h^{k+1}_{ij} = \frac{\hat{a}^{k}_j p(x_i \mid \hat{\phi}^k_j)}{\sum_{j=1}^{M} \hat{a}^{k}_j p(x_i \mid \hat{\phi}^k_j)} (i = 1..N, \ j = 1..M)
\]

Proof. We prove the results in the reverse orders. Also the form for the dependent parameters and the independent parameters may be different for different families.

Family 4: (Independent Centers and Independent Circles)
Denote \( \Sigma_j = c_j I_d \). In this class \( \Phi = \{a_1, ..., a_M, m_1, ..., m_M, c_1 I_d, ..., c_M I_d\} \).
Since the parameters for mixing proportions and for positions \( \{a_1, ..., a_M, m_1, ..., m_M\} \) are all mutually independent each other, the procedures shown in the generalized GBFN case can be applied without modification for these parameter estimation. Therefore the result in the Lemma is obtained.

Since the matrices \( \Sigma_j (j = 1, ..., M) \) are independent, we can use the decomposed optimization as in Eq. 7.18 and therefore the necessary condition 7.20 too.
Since,
\[
\log p_j(x_i \mid \phi_j) = -\frac{1}{2} \log \det(c_j I_d) - \frac{d}{2} \log(2\pi) - \frac{1}{2} (x_i - m_j)^T (c_j^{-1} I_d) (x_i - m_j)
\]

the differentiation with respect to the parameter \(c_j\) yields
\[
\frac{\partial \log p_j(x_i \mid \phi_j)}{\partial c_j} = -\frac{d}{2} c_j^{-1} + \frac{1}{2} c_j^{-2} (x_i - m_j)^T (x_i - m_j).
\]

Thus, substituting the above equation into the necessary condition 7.20 yields
\[
\Sigma_{j}^{k+1} = c_j I = \left\{ \frac{\sum_{i=1}^{N} h_{ij}^{k+1} \| x_i - \hat{m}_{j}^{k+1} \|^2}{d \sum_{i=1}^{N} h_{ij}^{k+1}} \right\} I \quad (j = 1..M)
\]

**Family 3: (Independent Centers and the Same Ellipses)**

Denote \(\Sigma_j = \Sigma\). In this class \(\Phi = \{a_1, \ldots, a_M, m_1, \ldots, m_M, \Sigma\}\) such that
\[
p(x \mid \Phi) = \sum_{j=1}^{M} a_j p_j(x \mid m_j, \Sigma)
\]

Parameters \(a_j, m_j (j = 1..M)\) are obtained by the same procedure applied in **Family 4**.

Since the matrices \(\Sigma_j\) are dependent, we can not use the decomposed optimization as in Eq. 7.18. However, this modification does not change the basic EM framework, so we can apply

\[
\text{E step} : Q(\Phi \mid \Phi_k) = E\{L_c(\Phi \mid x^n, z^n) \mid x^n, \Phi_k\}
\]

\[
\text{M step} : \Phi_{k+1} = \arg \max_{\Phi} Q(\Phi \mid \Phi_k)
\]

where
\[
L_c(\Phi \mid x^n, z^n) = \sum_{j=1}^{M} \sum_{i=1}^{N} z_{ij} \left( \log a_j + \log p_j(x_i \mid m_j, \Sigma) \right)
\]

and \(Q(\Phi \mid \Phi_k)\) is now defined as
\[
Q(\Phi \mid \Phi_k) = \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k} \left( \log a_j + \log p(x_i \mid m_j, \Sigma) \right)
\]

The \(M\)-step is
\[
\max_{\Phi} Q(\Phi \mid \Phi_k)
\]
\[
= \left\{ \max_{a_1, \ldots, a_M} \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log a_j \right\} \quad \text{and} \quad \left\{ \max_{m_1, \ldots, m_M} \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log p_j(x_i \mid m_j, \Sigma) \right\}
\]

Since \( m_1, \ldots, m_M \) are independent, the second part can be decomposed as

\[
= \left\{ \max_{m_1} \sum_{i=1}^{N} h_{i1}^{k+1} \log p_1(x_i \mid m_1, \Sigma) \right\} \\
\quad \vdots \\
\quad \left\{ \max_{m_M} \sum_{i=1}^{N} h_{iM}^{k+1} \log p_M(x_i \mid m_M, \Sigma) \right\} \\
\quad \left\{ \max_{m} \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log p_j(x_i \mid m_j, \Sigma) \right\}
\]

where

\[
\log p_j(x_i \mid m_j, \Sigma) = -\frac{1}{2} \log \det \Sigma - \frac{d}{2} \log(2\pi) - \frac{1}{2} (x_i - m_j)^T \Sigma^{-1} (x_i - m_j)
\]

By using the facts that \((\partial / \partial S) \log \det S = (S^{-1})^T\) (see Appendix A4), \((\partial / \partial S)c^T S^{-1} c = -(S^{-1} cc^T S^{-1})^T\) (see Appendix A4), and \(\Sigma^T = \Sigma\), the differentiation with respect to the parameter \(\Sigma\) yields

\[
\frac{\partial \log p_j(x_i \mid m_j, \Sigma)}{\partial \Sigma} = -\frac{1}{2} \Sigma^{-1} + \frac{1}{2} \Sigma^{-1} (x_i - m_j)(x_i - m_j)^T \Sigma^{-1}
\]

Thus the necessary condition for maximization is

\[
\sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}^{k+1} \frac{\partial \log p_j(x_i \mid m_j, \Sigma)}{\partial \Sigma} = 0
\]

\[
\Leftrightarrow \sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}^{k+1} \left\{-\frac{1}{2} \Sigma^{-1} + \frac{1}{2} \Sigma^{-1} (x_i - m_j)(x_i - m_j)^T \Sigma^{-1} \right\} = 0
\]

\[
\Leftrightarrow \sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}^{k+1} \Sigma = \sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}^{k+1} (x_i - m_j)(x_i - m_j)^T \quad (7.26)
\]

where

\[
\sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}^{k+1} = N
\]
and the RHS of Eq. 7.26 can be simplified as follows.

\[
RHS = \sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}^{k+1} \left\{ x_i x_i^T - 2m_j x_i^T + m_j m_j^T \right\}
\]

\[
= \sum_{i=1}^{N} x_i x_i^T \sum_{j=1}^{M} h_{ij}^{k+1} - 2 \sum_{j=1}^{M} m_j \sum_{i=1}^{N} h_{ij}^{k+1} x_i^T + \sum_{j=1}^{M} m_j m_j^T \sum_{i=1}^{N} h_{ij}^{k+1}
\]

\[
= \sum_{i=1}^{N} x_i x_i^T - \sum_{j=1}^{M} m_j m_j^T \sum_{i=1}^{N} h_{ij}^{k+1}
\]

Hence with some algebraic manipulation, the following is obtained.

\[
\hat{\Sigma}^{k+1} = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T - \sum_{j=1}^{M} \hat{a}_j^{k+1} \hat{m}_j^{k+1} \hat{m}_j^{k+1 T}
\]

**Family 2: (Independent Centers and the Same Circles)**

Denote \( \Sigma = cI \). In this case \( \Phi = \{a_1, ..., a_M, m_1, ..., m_M, cI_d \} \). The parameters \( a_j, m_j \) \((j = 1, ..., M)\) are obtained by the same procedure applied in Family 4.

Since

\[
\sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \log p_j(x_i \mid m_j, \Sigma)
\]

\[
= \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \left\{ -\frac{1}{2} \log \det(cI_d) - \frac{d}{2} \log(2\pi) - \frac{1}{2} (x_i - m_j)^T (c^{-1} I_d) (x_i - m_j) \right\}
\]

\[
= \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \left\{ -\frac{d}{2} \log c - \frac{d}{2} \log(2\pi) - \frac{1}{2c} (x_i - m_j)^T (x_i - m_j) \right\}
\]

the necessary condition for the parameter \( c \) is

\[
\sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \left\{ -\frac{d}{2c} + \frac{1}{2c^2} \|x_i - m_j\|^2 \right\} = 0
\]

Thus,

\[
c = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \|x_i - m_j\|^2}{\sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \|x_i - m_j\|^2}
\]

\[
= \frac{1}{Nd} \left( \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} \|x_i - m_j\|^2 \right)
\]

\[
= \frac{1}{Nd} \left( \sum_{i=1}^{N} x_i^T x_i \sum_{j=1}^{M} h_{ij}^{k+1} - \sum_{j=1}^{M} \sum_{i=1}^{N} h_{ij}^{k+1} m_j^T m_j \right)
\]

\[
= \frac{1}{Nd} \left( \sum_{i=1}^{N} x_i^T x_i - \sum_{j=1}^{M} (Na_j) \|m_j\|^2 \right)
\]

\[
= \frac{1}{d} \left( \sum_{i=1}^{N} \|x_i\|^2 - \sum_{j=1}^{M} \hat{a}_j^{k+1} \|m_j\|^2 \right)
\]
Therefore
$$\hat{\Theta}_{i}^{k+1} = c I_{d} = \frac{1}{d} \left( \frac{1}{N} \sum_{i=1}^{N} \| x_{i} \|^2 - \sum_{i=1}^{M} \tilde{a}_{j}^{k+1} \| \tilde{m}_{j}^{k+1} \|^2 \right) I_{d}$$

**Family 1** : *(Fixed Centers and the Same Circles)*

Denote $\Sigma = c I_{d}$. In this case $\Phi = \{ a_1, ..., a_M, c I_d \}$. The parameters $a_j$ ($j = 1..M$) are obtained by the same procedure applied in *Family 4* and $\Sigma$ is obtained by the same procedure applied in *Family 2*.

### 7.4 Model Selection Based on Minimum Description Approach

In this section, the model selection problem for p.d.f. estimation is discussed. Model selection is important but difficult. If only parameter identification is considered, then the Maximum Likelihood Method and the Least Squares Method can be used. Traditionally, a covariance is used as a measure of goodness and the Cramer-Rao inequality sets a lower bound for the error covariance of any unbiased estimator. However, these methods can not solve the model order selection problem. If we try to use the ML method between models with different numbers of parameters, then its preference will tend to increase as the number of independent parameters increases until the number of kernels becomes the same as the number of samples. This leads to a non-parametric method which is not an issue in this thesis.

It is known that the linear combination of smooth convex functions is able to approximate any class of function within arbitrary accuracy, if sufficient kernels and training samples are available. However, in practical situations, such sufficient training samples may not be available due to the cost problem. Moreover the training samples are often corrupted by noise and this requires more samples. Hence, unless a large number of samples are available, learning a function with a complicated parameterization does not make sense. When one attempts to teach the network by using few samples, the network open learns up to noise level and the results may often be unwanted patterns in the region where the training samples are sparse. This
phenomenon is due to redundant flexibilities of the network. Learning up to noise level degrades the generalization capability.

In functional approximation problems, if much data is available, then the Cross Validation type test method can be used, where a set of data is separated into three parts: training data, test data, and validation data. By this was, the model which is believed to be the most appropriate one is selected based on evidence.\(^6\) This idea is also available in the density estimation by checking the likelihood criteria instead of LMS criteria. However, if samples are not sufficient then this model selection method is not sufficient.\(^7\) Although every method will be the same, the MDL method can be useful because it does not require additional data other than the training data.

The Minimum Description Length (MDL) approach is a model selection method based on descriptive complexity. The descriptive complexity means the minimum required code length when the observed data sequences are encoded. The MDL approach can be used as a means of comparing among each of the models with different size, order or basis functions. This method asserts that for a given data sequence and a class of probabilistic models, one should choose the model that yields the shortest description length of the data. The underlying aim is to seek a good ‘summarized’ representation rather than a detailed representation.\(^8\) Concerning the complexity penalized approach, the MDL approach gives an automatic trade-off between the complexity cost and the data cost without worrying about the weighting factor \(\lambda\) in the following equation.

\[
\text{Total Cost} = \text{Data Cost} + \lambda \cdot \text{Complexity Cost}
\]

This MDL approach can be applied in unsupervised learning problems (e.g.,

---

\(^6\)This method is good when there are enough number of samples. To do this, the network parameters are found by minimizing the LMS criteria and the learning task is stopped at the time step when this network begins to learn the noise. This can be checked by using the Validation Data set. Hence the Validation Data set gives an idea about when to stop the training to avoid over-fitting. Continuing this training tends to give an over-fitting problem. Test Data set is used in selecting the most appropriate model among the candidates.

\(^7\)There is the S-fold Cross Validation method which uses data in the most efficient way. However, unless one has sufficient data this method yields poor results as well.

\(^8\)Suppose there are 10 samples drawn from the complicated p.d.f. function which is mixed with 10 kernels. Assume each sample was drawn from each kernel. In this case, a good p.d.f. estimate is the one that consists of one kernel rather than ten kernels. [Barron]
density estimation problem) and in supervised learning problems (e.g., regression problem). The MDL criteria can be briefly explained as follows.

\[ q_{opt} = \min_{q \in S} \left( \text{code length for parameters in } q \right) + \left( \text{code length for datas, given } q \right) \]

\[ \text{Curve Fitting :} \]

\[ q_{opt} = \min_{q \in S} \left( \text{code length for parameters in } q \right) + \left( \text{code length for data misfits, given } q \right) \]

Here \( S \) denotes the set of countable candidates from which one member is selected.

This MDL method is deeply rooted in Komogorov's algorithmic complexity which represents the minimum length program that can generate the given observation sequence. The length of the shortest program for a given sequence has been believed to be a measure of its intrinsic complexity.\(^9\)

One can think of candidates of models (or representation) as candidates of coding schemes. Finding a good model (representation) is equivalent to finding a good coding scheme based on information theoretic sense.

A model is regarded to be better, if it can represent the observation sequence with shorter description length. The MDL method can be understood within the Bayesian framework and this will be explained in the next section. The basic mechanism causes one to select a simple model when only a small number of samples (or realizations or evidence) are available, and to select more flexible models if many samples are available. Mathematical consistencies were proved in smooth parametric density estimation and nonparametric density estimation too, in \( L_1 \) sense (Barron and Cover 91).

In this chapter the MDL method be applied in estimating the appropriate number of the kernels in the GBFN for p.d.f. estimation. The application to density estimation problem is very important because detection problems heavily depend upon the density estimation.

\(^9\)For example, in this sense, fractal images are believed to have very low complexities.
7.4.1 An Overview

Bayesian Interpretation

The MDL approach can be well explained in the light of the Bayesian framework. Suppose one has samples \( \{X_i\}_{i=1}^n \) drawn from an unknown p.d.f. ranging over a finite countable set. Assume that the sequence is i.i.d., hence stationary and ergodic. Let \( S \) be a list of candidate p.d.f.s \( \{q_1, q_2, \ldots\} \) which is specified from a given sequence of parametric models of increasing dimension. Let \( p(q) \) be a prior p.m.f. of \( q \in S \). The objective is to select a model and estimate parameters. To be more specific, the \( q \) that maximizes the a posterior \( p(q \mid X^n) \) is to be selected in the list \( S \).

A key idea in the MDL approach is that it chooses its prior \( p(q) = 2^{-L(q)} \) where \( L(q) \) is the length of a prefix code which represents the parameters in \( q \). By choosing this prior, maximizing the a posterior probability is equivalent to minimizing the total description. To be more specific

\[
\begin{align*}
&\max_{q \in S} p(q \mid X^n) \\
= &\max_{q \in S} p(X^n \mid q) \cdot p(q) \\
= &\max_{q \in S} \log \left\{ p(X^n \mid q) \cdot p(q) \right\} \\
= &\min_{q \in S} \log \frac{1}{p(X^n \mid q) \cdot p(q)} \\
= &\min_{q \in S} \left( \sum_{i=1}^n \log \frac{1}{p(X_i \mid q)} + \log \frac{1}{p(q)} \right) \\
= &\min_{q \in S} L(X^n \mid q) + L(q) \\
= &\min_{q \in S} L(X^n, q)
\end{align*}
\]

(7.27)

Also, a similar meaning exists in the curve fitting problem. Suppose \( X^n \) is the independent variable’s data sequence and \( Y^n \) is the dependent variable’s data sequence. Let \( S \) be a list of candidate p.d.f.s \( \{q_1, q_2, \ldots\} \) of the residual sequence. Let \( p(q) \) be a prior p.m.f. of \( q \in S \) and assume \( p(q \mid X^n) = p(q) \). The objective is to
select a model and estimate parameters. To be more specific, the $q$ that maximizes the a posterior $p(q \mid X^n, Y^n)$ is to be selected in the list $S$. Then maximizing the a posterior probability becomes minimizing the total description length of the dependent variable’s data sequence conditioned on the knowledge of the independent variable’s data. Assume that $Y^n$ is independent when it is conditioned on $X^n$. Then the following relations can be obtained.

$$\max_{q \in S} p(q \mid X^n, Y^n)$$
$$= \max_{q \in S} p(Y^n \mid q, X^n) \cdot p(q \mid X^n)$$
$$= \max_{q \in S} p(Y^n \mid q, X^n) \cdot p(q)$$
$$= \max_{q \in S} \log \{ p(Y^n \mid q, X^n) \cdot p(q) \}$$
$$= \max_{q \in S} \log p(Y^n \mid q, X^n) + \log p(q)$$

(7.28)

$$= \min_{q \in S} \left( \sum_{i=1}^{n} \log \frac{1}{p(Y_i \mid q, X^n)} \right) + \log \frac{1}{p(q)} \quad \text{code length for data plus fits}$$
$$\quad \phantom{= \min_{q \in S} \left( \sum_{i=1}^{n} \log \frac{1}{p(Y_i \mid q, X^n)} \right) + \log \frac{1}{p(q)} } \quad \text{code length for parameters}$$
$$= \min_{q \in S} L(Y^n \mid q, X^n) + L(q)$$
$$= \min_{q \in S} L(Y^n, q \mid X^n)$$

Hence the MDL principle can be briefly summarized in the light of Bayesian framework as follows.

**MDL principle** = **Bayesian framework** + choice of prior to $2^{-L_{n}(q)}$

**Two Stage Code**

It was shown that minimizing the total description length is equivalent to maximizing the a posterior probability in the density estimation problem.

**Density Estimation** : \( \min_{q \in S} L(X^n, q) = \min_{q \in S} L(X^n \mid q) + L(q) \)

The best coding is the one that has the minimum total description length which is the sum of prefix codes and data codes. Here prefix codes are codes allocated for parameter representation, and data codes are codes allocated for data representation using the parameters specified by the prefix code. There are two meanings in $L(X^n \mid$
q). In an estimation's point of view, this is the negative log-likelihood function. On the other hand, in a coding theory's point of view, this is the length of data codes. Increasing the log-likelihood in estimation is equivalent to decreasing the length of the data code in coding theory. If prefix codes are accurately determined then this data code will have little redundancy in the sense of Shannon length. But if prefix codes are poorly determined then data codes will have much redundancies. Thus accurately determined prefix codes reduce the length of the data codes and these can be determined by the distribution of the data. However, such prefix codes increase their own length. Hence they should not be too detailed and there must be a trade-off between the prefix code length and the data code length.

To be more specific, suppose $\theta^{(k)} = (\theta_1, \theta_2, ..., \theta_k)^T$, $\theta_i \in R^1 \ (i = 1..k)$ are parameters in the $k$-dimensional space. Let $\pi(\theta^{(k)})$ be the distribution of the parameters, the 'prior distribution'. Denote $q(\theta^{(k)})$ a p.d.f. which is parameterized by $\theta^{(k)}$. The length of the codes for parameters in $q(\theta^{(k)})$ depends on their quantizations. Suppose $\tilde{\theta}^{(k)} = (\tilde{\theta}_1, \tilde{\theta}_2, ..., \tilde{\theta}_k)^T$, $\tilde{\theta}_i \in R^1 \ (i = 1..k)$ are the truncated parameters of the $\theta^{(k)}$ to the precision $\delta^{(k)} = (\delta_1, \delta_2, ..., \delta_k)^T$, where $\delta_i = 2^{-d_i}$ for $1 \leq i \leq k$ for some number of fractional digits $d_i$. Then the unit volume in the quantized parameter space is $dV = \prod_{i=1}^{k} \delta_i$. Thus the length of the prefix codes can approximately computed as follows.

$$L(q(\tilde{\theta}^{(k)})) = \log \frac{1}{\Pr\{\theta^{(k)} \in \text{specified unit volume}\}}$$

$$= \log \frac{1}{\pi(\tilde{\theta}^{(k)}) \cdot dV}$$

$$= \log \frac{1}{\pi(\theta^{(k)})} + \sum_{i=1}^{k} \log \frac{1}{\delta_i} \quad (7.29)$$

As $\delta_1, ..., \delta_k$ increase, the length of the prefix codes decrease. But such $\delta_1, ..., \delta_k$ should not be increased arbitrarily since they will increase the length of the data codes. Thus they must be optimally truncated and the optimal truncation was found by Rissanen (89). The above length is also influenced by the priors $\pi(\theta^{(k)})$\textsuperscript{10}, and thus

\textsuperscript{10}The meaning of 'prior' is different from that of the Bayesian interpretation. In Baysian interpretation, the prior density was used as a means to express one's prior knowledge about the true value of parameters so that small variance means a good initial knowledge. But in this case, parameters are not generated by repeatable random experiments. and the knowledge of the priors is used for a coding purpose, not for an estimation purpose. Since an important part is the complexity of the
uniformly distributed parameters (which we often implicitly assume, unless one knows the priors) consume the largest description length among all kinds of distributions. If parameters are distributed with Gaussian distribution, then the average description length is smaller than that of the uniform distribution. This explains why the complexity penalized criterion can reduce the effective number of parameters. If the Gaussian p.d.f. is applied to prior distribution \( \pi(\theta^{(k)}) \), then \( \sum_{i=1}^k \| \theta_i \|^2 \) type of LMS penalty can be obtained.

In the functional approximation problem, MDL is defined to be the minimum description length of the codes for the sequence of the dependent variable’s data conditioned on independent variable’s data. The description length is the length of the minimum codes from which dependent data are reconstructable by the knowledge of the independent data. Codes consists of two parts: codes for specifying parameters and codes for specifying data misfits. Thus the problem becomes

\[
\text{Curve Fitting : } \min_{q \in S} L(Y^n, q | X^n) = \min_{q \in S} L(Y^n | q, X^n) + L(q)
\]

Given independent data, one can reconstruct the dependent data by simply adding a data misfit to the prediction, and the prediction is obtained by the knowledge of the parameters and the independent data. In order to reduce the total code length, one should know (or assume) two things: the density function of the data misfit and the prior distribution (Uniform prior is applied if there is no specific knowledge about parameter distribution). The knowledge of these two distributions enables one to encode in a Shannon sense.

**MDL Principle**

The MDL principle is used for the evaluation of the total code length given a candidate model. This enables one to compare each of the different representation by each of the minimum required code length. The minimum required code length can be obtained by the optimal truncation of the parameters asymptotically. Rissanen model, one should be able to evaluate the length of the codes which denote parameter values. To do this parameters are truncated to a finite precision and encoded. Length of such codes can be evaluated asymptotically by knowing the ‘prior’ distribution (i.e the distribution of the parameters) in the Shannon sense. (Rissanen 89)
showed that the minimum required code length, when a model with $k$ independent parameters are used, is asymptotically given by the Eq. 7.34.

The optimal truncation, a key idea in the MDL approach, leads to the minimum description length of the data and parameters. This can be obtained by the following minimization problem.

$$
\delta^{(k)*} = \min_{\delta_1, \delta_2, \ldots, \delta_k} L(X^n, q(\delta^{(k)}))
$$

where $L(X^n, q(\delta^{(k)}))$ is obtained by the result of Eq. 7.27 and Eq. 7.29 as follows.

$$
L(X^n, q(\delta^{(k)})) = L(X^n | q(\delta^{(k)})) + L(q(\delta^{(k)}))
$$

$$
= \log \frac{1}{p(X^n | q(\delta^{(k)}))} + \log \frac{1}{\pi(\delta^{(k)})} + \sum_{j=1}^{k} \log \frac{1}{\delta_j} \quad (7.30)
$$

where the last equation is obtained by the observation that $p(X | q) = q(X)$.

Denote $L(X^n, \bar{\theta}^{(k)}) = L(X^n, q(\bar{\theta}^{(k)}))$ and $L(X^n | \bar{\theta}^{(k)}) = L(X^n | q(\bar{\theta}^{(k)}))$ to show dependencies more clearly. Denote $\theta^{(k)*}$ the maximum likelihood estimate of the parameters. The basic idea is (1) to apply the Taylor’s series to the function which denotes the total description length around the maximum likelihood point, and (2) to find the minimum with respect to the perturbation. Assuming the function is analytic, the Taylor series is

$$
L(X^n, \bar{\theta}^{(k)}) = L(X^n, \theta^{(k)*}) + \frac{1}{2} \delta^{(k)T} H(\bar{\theta}^{(k)}) \delta^{(k)} + O(|| \delta^{(k)} ||^4)
$$

where $H(\bar{\theta}^{(k)})$ denotes the curvature $\partial^2 L / \partial \theta^2 |_{\bar{\theta}^{(k)}}$, and $\delta^{(k)} = (\delta_1, \ldots, \delta_k)$ denotes the perturbation $\bar{\theta}^{(k)} - \theta^{(k)*}$. The first order gradient term is deleted since it is zero. The above equality becomes simpler by applying the Taylor theorem. That is

$$
L(X^n, \bar{\theta}^{(k)}) = L(X^n, \theta^{(k)*}) + \frac{1}{2} \delta^{(k)T} H(\bar{\theta}^{(k)}) \delta^{(k)} \quad (7.31)
$$

where $\bar{\delta}^{(k)}$ is a vector between $\theta^{(k)*}$ and $\bar{\theta}^{(k)}$. Substituting the result of Eq. 7.30 to Eq. 7.31 leads to

$$
L(X^n, \bar{\theta}^{(k)})
$$

$$
= -\log q(X^n | \theta^{(k)*}) \pi(\theta^{(k)*}) - \sum_{j=1}^{k} \log \delta_j + \frac{1}{2} \delta^{(k)T} H(\bar{\theta}^{(k)}) \delta^{(k)}
$$

$$
\Delta J(\delta^{(k)})
$$
The above equation is minimized when

\[ H(\tilde{\theta}^{(k)})\delta^{(k)} = (1/\delta_1, ..., 1/\delta_k) \] (7.32)

and this leads to the minimum \( J(\delta^{(k)}) \) as follows.

\[ J(\delta^{(k)*}) = - \log q(X^n | \theta^{(k)*})\pi(\theta^{(k)*}) - \sum_{j=1}^{k} \log \delta_j + \frac{k}{2} \] (7.33)

Assuming that \(- \log q(X^n | \theta^{(k)*})\) grows proportionally to \( n \), the elements of curvature term \( H \) are order of \( n \). Hence Eq. 7.32 implies that \( \delta_j^* = c_j/\sqrt{n} \) for some constant \( c \). and the Eq. 7.33 becomes

\[ J(\delta^{(k)*}) = - \log q(X^n | \theta^{(k)*})\pi(\theta^{(k)*}) - \sum_{j=1}^{k} \log \frac{c_j}{\sqrt{n}} + \frac{k}{2} \]

\[ = - \log q(X^n | \theta^{(k)*})\pi(\theta^{(k)*}) + \frac{k}{2} \log n + O(k) \]

where \( O(k) = (k/2) - \sum_{j} \log c_j \). Note \( O(k) \) can be neglected as \( n \) increases. This leads to the following simple criteria which should be minimized with respect to \( k \) (Rissanen 1983; Rissanen 1989)

\[ - \log \left( q(X^n | \theta^{(k)}) \cdot \pi(\theta^{(k)}) \right) + \frac{k}{2} \log n \] (7.34)

When there is no prior information, one can omit \( \pi(\theta^{(k)}) \). In order to evaluate the above criteria, one needs to search the parameters on a uniformly gridded (with \( 1/\sqrt{n} \) spacing) parameter space. If confidence interval of the estimate at the maximum likelihood point can be evaluated, then truncation within the confidence interval is a good convenient method, because it does not require a search over the gridded space. This increases the description length by only a small amount (Barron 84; Barron and Cover 91). However in the GBN, the computation for such confidence interval is difficult. Moreover, in most cases, computational simulation shows the curvature term is dominated by other terms. Thus, in this thesis, parameters will be found by the Maximum Likelihood estimation method, however such parameters will be assumed to be already optimally truncated in model selection problems for convenience.
7.4.2 Application to GBNF Density Estimation

In this subsection, the MDL criteria is applied to GBNF. This is needed since one wants to find a good estimation of the number of kernels and a good basis function with appropriate flexibility. Although GBNF has non-homogeneous parameters, one can apply the MDL by monitoring the independent parameters. This independence is the most important aspect in the MDL because Rissanen's MDL criteria is derived assuming the independence of the parameters.

Suppose samples \( \{X_i\}_{i=1}^n \) are drawn from an unknown p.d.f. Assume that the sequence is i.i.d., hence stationary and ergodic. The families of GBNFs to be considered are the same as those explained in earlier section. That is

\[
q(x | \Phi) = \sum_{l=1}^M a_l p_l(x | \phi_l)
\]

where

\[
p(x | \phi_l) = (2\pi)^{-d/2} (\det \Sigma_l)^{-1/2} \exp \left\{ -\frac{1}{2}(x - m_l)^T \Sigma_l^{-1} (x - m_l) \right\}
\]

\[
\sum_{l=1}^M a_l = 1 \quad \text{and} \quad a_l > 0 \quad \text{for} \quad 1 \leq l \leq M
\]

\[
x \in \mathbb{R}^d, \quad m_l \in \mathbb{R}^d, \quad \Sigma_l \in \mathbb{R}^{d \times d}
\]

where \( \Phi \) is the set of total independent parameters. \( \phi_l = (m_l, \Sigma_l^{-1}) \), and \( M \) is the number of kernels employed.\(^{11}\) The families to be considered are

**Class 1:** *(Fixed Centers with same Circles)*

- \( m_1, ..., m_M \) are uniformly fixed
- \( a_1, ..., a_M \) are parameters
- \( \Sigma_l^{-1} = \Sigma^{-1} = v^{-1}I \) \((l = 1..M)\) is a parameter

**Class 2:** *(Variable Centers with Same Circles)*

- \( a_1, ..., a_M \) are parameters
- \( m_l \) and \( \Sigma_l^{-1} = \Sigma^{-1} = v^{-1}I \) are parameters \((l = 1..M)\)

\(^{11}\)M is not number of hypotheses here.
Class 3: *(Variable Centers with Same Ellipses)*

\[ a_1, \ldots, a_M \text{ are parameters} \]

\[ m_l \text{ and } \sum_l^{-1} = \sum^{-1} \text{ are parameters } (l = 1 \ldots M) \]

Class 4: *(Variable Centers with Independent Circles)*

\[ a_1, \ldots, a_M \text{ are parameters} \]

\[ m_l \text{ and } \sum_l^{-1} = v_l^{-1} I_d \text{ are parameters } (l = 1 \ldots M) \]

Class 5: *(Variable Centers with Independent Ellipses)*

\[ a_1, \ldots, a_M \text{ are parameters} \]

\[ m_l \text{ and } \sum_l^{-1} \text{ are parameters } (l = 1 \ldots M) \]

The objective here is to select the best model using the MDL framework by comparing each estimator which is obtained by the Maximum Likelihood Method. Let \( F \) denote the index for representing the family. For each family, let \( S_F \) be the list of candidate p.d.f.s \( \{ q^{(1,F)}, q^{(2,F)}, \ldots, q^{(M,F)} \} \) where \( M_F \) is the maximum number of kernels to be checked in the family of \( S_F \). Denote \( \Phi^{(M,F)}_\ast \) \( (F = 1 \ldots 5) \) the set of independent parameters when \( M \) kernels are employed for each family \( F \). For each case the dimension of \( \Phi \) is

**Family 1** :
\[ \dim(\Phi^{(M,1)}) = M + 1 \]

**Family 2** :
\[ \dim(\Phi^{(M,2)}) = M + Md + 1 \]

**Family 3** :
\[ \dim(\Phi^{(M,3)}) = M + Md + d \]

**Family 4** :
\[ \dim(\Phi^{(M,4)}) = M + Md + M = M(d + 2) \]

**Family 5** :
\[ \dim(\Phi^{(M,5)}) = M + Md + Md(d + 1)/2 = M(d + 1)(d + 2)/2 \]

(7.35)

For each family, the objective is to find the total length of the two stage codes that has the minimum total description length. That is

\[
\min_{1 \leq M \leq M_F} \left\{ \frac{L(X^n | q^{(M,F)}_\ast)}{\text{total description length}} + \frac{L(q^{(M,F)}_\ast)}{\text{total description length}} \right\} \quad \text{for each } F, \ 1 \leq F \leq 5
\]

where \( q^{(M,F)}_\ast \) is the maximum likelihood estimate when \( M \) kernels in \( F \)-th family are
used. By substituting the Eq. 7.35 to 7.34 the followings are obtained.

\[
q^{(M*,1)}(M) = \min_{1 \leq M \leq M_1} \left( \frac{M + 1}{2} \log n - \sum_{i=1}^{n} q(X_i \mid \Phi^{(M,1)}) \right)
\]

\[
q^{(M*,2)}(M) = \min_{1 \leq M \leq M_2} \left( \frac{M + Md + 1}{2} \log n - \sum_{i=1}^{n} q(X_i \mid \Phi^{(M,2)}) \right)
\]

\[
q^{(M*,3)}(M) = \min_{1 \leq M \leq M_3} \left( \frac{M + Md + d}{2} \log n - \sum_{i=1}^{n} q(X_i \mid \Phi^{(M,3)}) \right)
\]

\[
q^{(M*,4)}(M) = \min_{1 \leq M \leq M_4} \left( \frac{M(d + 2)}{2} \log n - \sum_{i=1}^{n} q(X_i \mid \Phi^{(M,4)}) \right)
\]

\[
q^{(M*,5)}(M) = \min_{1 \leq M \leq M_5} \left( \frac{M(d + 1)(d + 2)}{4} \log n - \sum_{i=1}^{n} q(X_i \mid \Phi^{(M,5)}) \right)
\]

Hence, if the family is fixed, then the q can be determined.

If the optimal representation is to be selected among different families then direct comparison method can be applicable by using the above criteria. This direct comparison can be justified by the following observation. Given a number of samples, select the one that has the minimum description length in the candidate list \( \bigcup_i S_i \). Define the three stage coding system which consists of three parts: codes for specifying family, codes for specifying parameters and code for data. Then the total description length becomes

\[
L(X^n, \Phi, \mathcal{F}) = L(\mathcal{F}) + L(\Phi \mid \mathcal{F}) + L(X^n \mid \Phi, \mathcal{F})
\]

Assume no prior information about the preference for a particular family is available. Thus the length of codes specifying each family are all the same, and thus can be deleted from the criteria. Therefore this coding system becomes a two-stage coding system.\(^{12}\) The optimal representation in the MDL sense becomes

\(^{12}\) Note, if there are unequal priors of the preference for family then the criteria based on the three stage coding system cannot be simplified. This is because there can be large deviations in the length of codes for specifying each family.
\[(M^*, F^*) = \arg \min_{M,F} \left( L(\Phi_{i}^{(M,F)} \mid S_F) - \sum_{i=1}^{n} p(X_i \mid \Phi_{i}^{(M,F)}) \right) \]

### 7.4.3 Summary of Model Selection

The basic principle of the MDL method and application to the model selection problem in the GBFNs were studied. Although this method does not enable one to know the optimal number of parameters in a direct fashion, it gives a criterion by which one can compare the performances among candidates, and hence it allows for an optimal trade-off between complexity and likelihood in density estimation problems.

Among the GBFN, the desired representation must not be over-smoothed nor under-smoothed, so the best trade-off is the one that has the shortest descriptive length. Thus a simple coding concept was required. The underlying concept and philosophy were historically based on Occam’s razor and Kolmogorov’s complexity. The proposed selection method was simply picking ‘the best of each best’. If there were more than one families, then a slight extension to the minimization of the three-stage codes was needed. But by assuming that there is no specific reason to have non-uniform priors in the preference for the family, a two-stage code length comparison was used instead of a three-stage method. Exact computation for the length of the codes is impossible to find and is not needed anyway.

An important area being developed concerns the prior distribution. If one can use specific priors, then given a total number of parameters, the complexity can be decreased more than that of the uniform prior, because the uniform prior distribution has the maximum entropy which requires the largest code length. This prior concept has been introduced in the Multi-Layered Perceptron, where all parameters were homogeneous. In GBFN, this prior concept has not been introduced yet except for the linear problem class.
7.5 Probing Signal Selection

This section explains the probing signal generation in the detection network. The basic idea is the same as the approach described in Chapter 4. The following two hypothesized models are considered.

\[ H_i : \quad y_{k+1} = \mathcal{P}_i(y^k, u^k) + e_{k+1}^{(i)}, \quad p(e_k^{(i)}) = Q_i(e_k^{(i)}) \]
\[ H_j : \quad y_{k+1} = \mathcal{P}_j(y^k, u^k) + e_{k+1}^{(j)}, \quad p(e_k^{(j)}) = Q_j(e_k^{(j)}) \]

where \( y_k \in \mathbb{R}^{d_1} \) is the observation vector, \( u_k \in \mathbb{R}^{d_2} \) is the input vector. \( \{e_k^{(i)}\}_{k=0}^{\infty} \) and \( \{e_k^{(j)}\}_{k=0}^{\infty} \) are two i.i.d. zero mean noise process. \( \mathcal{P}_i, \mathcal{P}_j \) are GBFNs for prediction modules. and \( Q_i, Q_j \) are GBFNs for p.d.f. modules defined as follows

\[
Q_i(x) = \sum_{l=1}^{L^{(i)}} a_t^{(i)} \left| \Sigma_t^{(i)} \right| (2\pi)^{-d/2} \exp \left\{ -\frac{1}{2} \left\| x - m_t^{(i)} \right\|_{\Sigma_t^{(i)-1}}^2 \right\} \\
\quad \quad = \sum_{l=1}^{L^{(i)}} a_t^{(i)} G_t^{(i)}(x)
\]
\[
Q_j(x) = \sum_{l=1}^{L^{(j)}} a_t^{(j)} \left| \Sigma_t^{(j)} \right| (2\pi)^{-d/2} \exp \left\{ -\frac{1}{2} \left\| x - m_t^{(j)} \right\|_{\Sigma_t^{(j)-1}}^2 \right\} \\
\quad \quad = \sum_{l=1}^{L^{(j)}} a_t^{(j)} G_t^{(j)}(x)
\]

Thus

\[
\hat{y}_{k+1|k}^{(i)} = \mathcal{P}_i(y^k, u^k) \\
\hat{y}_{k+1|k}^{(j)} = \mathcal{P}_j(y^k, u^k)
\]  
(7.36)

Given \( \bar{u}_k \), the objective is to find \( \tilde{u}_k \) such that the maximum accumulation of the information can occur as follows.

\[
\tilde{u}_k = \arg \max_{\hat{u}_k} J^{i,j}_{k+1}(u_k) = \frac{1}{2}(D^{i,j}_{k+1}(u_k) + D^{i,j}_{k+1}(u_k)) \\
s.t. \quad |\hat{u}_k| \leq \bar{u}_{k,\text{max}}
\]

The same approach as explained in Chapter 4 requires

\[
\tilde{u}_k = \bar{u}_{k,\text{max}} \cdot \frac{\nabla_{\hat{u}_k} J^{i,j}_{k+1}(\hat{u}_k)}{\left\| \nabla_{\hat{u}_k} J^{i,j}_{k+1}(\hat{u}_k) \right\|}
\]  
(7.37)
The relative entropy can be obtained as follows.

\[ D^{i,j}_{k+1}(u_k) \]

\[ = D \left\{ p(y_{k+1}|y^k, u^k, H_i) \parallel p(y_{k+1}|y^k, u^k, H_j) \right\} \]

\[ = E \left\{ \log p(y_{k+1}|y^k, u^k, H_i) \mid y^k, u^k, H_i \right\} - E \left\{ \log p(y_{k+1}|y^k, u^k, H_j) \mid y^k, u^k, H_i \right\} \]

\[ = E \left\{ \log Q_i(y_{k+1} - \tilde{y}_{k+1|k}^{(i)}) \mid y^k, u^k, H_i \right\} - E \left\{ \log Q_j(y_{k+1} - \tilde{y}_{k+1|k}^{(j)}) \mid y^k, u^k, H_i \right\} \]

\[ = E \left\{ \log Q_i(z_{k+1|k}^{(i)}) - E \left\{ \log Q_i(y_{k+1} - \tilde{y}_{k+1|k}^{(i)} + e_{k+1}^{(i)}) \mid y^k, u^k, H_i \right\} \right\} + C \]

(7.38)

Denote

\[ z_{k+1|k}^{ij} = \tilde{y}_{k+1|k}^{(i)} - \tilde{y}_{k+1|k}^{(j)} \]

and

\[ \log(Q_j(\alpha)) = f_j(\alpha) . \]

By applying the Taylor series expansion to the result of Eq. 7.38, the followings are obtained.

\[ D^{i,j}_{k+1}(u_k) \]

\[ = -E \left\{ f_j(z_{k+1|k}^{ij} + e_{k+1}^{(i)}) \mid y^k, u^k, H_j \right\} + C \]

\[ = -E \left\{ f_j(z_{k+1|k}^{ij} \mid y^k, u^k, H_j) + e_{k+1}^{(i)} \partial f_j(z_{k+1|k}^{ij} \mid y^k, u^k, H_j) \partial z_{k+1|k}^{ij} \right\} + C \]

\[ = -f_j(z_{k+1|k}^{ij}) + E \left\{ O(\| e_{k+1}^{(i)} \|^2) \right\} + C \]

\[ \approx -f_j(z_{k+1|k}^{ij}) + C \]

\[ = -\log Q_j(z_{k+1|k}^{ij}) + C \]

Here the second order term is neglected. Hence the performance function is obtained as follows.

\[ J_{k+1}^{i,j}(u_k) = -\frac{1}{2} \log Q_j(z_{k+1|k}^{ij}) - \frac{1}{2} \log Q_i(z_{k+1|k}^{ij}) + C' \]
Suppose \( z_k^j = (z_1, \ldots, z_{d_1})^T \), then the gradient is.

\[
\nabla_u Q_j(z_k^j) = \begin{pmatrix}
\frac{\partial Q_i}{\partial u_1} \\
\vdots \\
\frac{\partial Q_i}{\partial u_{d_1}} \\
\end{pmatrix}\begin{pmatrix}
\frac{\partial z_1}{\partial z_1} \\
\vdots \\
\frac{\partial z_{d_1}}{\partial z_{d_1}} \\
\end{pmatrix} + \cdots + \begin{pmatrix}
\frac{\partial Q_i}{\partial u_1} \\
\vdots \\
\frac{\partial Q_i}{\partial u_{d_1}} \\
\end{pmatrix}\begin{pmatrix}
\frac{\partial z_1}{\partial z_1} \\
\vdots \\
\frac{\partial z_{d_1}}{\partial z_{d_1}} \\
\end{pmatrix} + \cdots + \begin{pmatrix}
\frac{\partial Q_i}{\partial u_1} \\
\vdots \\
\frac{\partial Q_i}{\partial u_{d_1}} \\
\end{pmatrix}\begin{pmatrix}
\frac{\partial z_1}{\partial z_1} \\
\vdots \\
\frac{\partial z_{d_1}}{\partial z_{d_1}} \\
\end{pmatrix}
\]

\[
= \begin{pmatrix}
\frac{\partial z_1}{\partial z_1} \\
\vdots \\
\frac{\partial z_{d_1}}{\partial z_{d_1}} \\
\end{pmatrix}\begin{pmatrix}
\frac{\partial Q_i}{\partial z_1} \\
\vdots \\
\frac{\partial Q_i}{\partial z_{d_1}} \\
\end{pmatrix}
\]

where \( J \) denotes the Jacobian matrix. Thus the gradient of the performance function with respect to the input is

\[
\nabla_u J_{k+1}^i(u_k) = -\frac{1}{2} Q_j^{-1}(z_{k+1}^j) \nabla_u Q_j(z_{k+1}^j) - \frac{1}{2} Q_i^{-1}(z_{k+1}^i) \nabla_u Q_i(z_{k+1}^i)
\]

\[
= -\frac{1}{2} Q_j^{-1}(z_{k+1}^j) \underbrace{\nabla_u Q_j(z_{k+1}^j)}_{\text{scalar}} - \frac{1}{2} Q_i^{-1}(z_{k+1}^i) \underbrace{\nabla_u Q_i(z_{k+1}^i)}_{\text{vector}}.
\]

(7.40)

The gradient of the p.d.f. is

\[
\nabla_x Q_j(z) = -\sum_{l=1}^{L(j)} d_l^{(j)}(2\pi)^{-d_1/2} |\Sigma_l^{(j)}|^{-1/2} \exp \left\{ -\frac{1}{2} \|z_l^{(j)} - m_l^{(j)}\|_{\Sigma_l^{(j)}}^{-1} \right\} \Sigma_l^{(j)}^{-1} (z_l^{(j)} - m_l^{(j)})
\]

\[
= -\sum_{l=1}^{L(j)} d_l^{(j)} G_l^{(j)}(z^{(j)}) \Sigma_l^{(j)}^{-1} (z_l^{(j)} - m_l^{(j)}).
\]

(7.41)

and the Jacobian is

\[
J_{uk}(z_{k+1}^j) = \frac{\partial z_{k+1}^j}{\partial u_k} = \frac{\partial \frac{\partial y_{k+1}^{(j)}}{\partial u_k}}{\partial \frac{\partial y_{k+1}^{(j)}}{\partial u_k}} - \frac{\partial \frac{\partial y_{k+1}^{(j)}}{\partial u_k}}{\partial \frac{\partial y_{k+1}^{(j)}}{\partial u_k}}
\]

(7.42)
Substituting the result of Eq. 7.41 and Eq. 7.42 to Eq. 7.40 yields

\[
\nabla_u J_{k+1}^{ij}(u_k) = \frac{1}{2} \mathcal{Q}_j^{-1}(z_k^{ij}) \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \left\{ \sum_{l=1}^{L(j)} a_l^{ij} \mathcal{G}_l^{(j)}(z_k^{ij}) \Sigma_l^{(j)-1}(z_k^{ij} - m_l^{(j)}) \right\}
\]

\[
- \frac{1}{2} \mathcal{Q}_i^{-1}(z_k^{ji}) \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \left\{ \sum_{l=1}^{L(i)} a_l^{ji} \mathcal{G}_l^{(i)}(z_k^{ji}) \Sigma_l^{(i)-1}(z_k^{ji} - m_l^{(i)}) \right\}
\]

(7.43)

where \( z_k^{ij} = \hat{x}_k^{(i)} - \hat{y}_k^{(j)} \) and \( z_k^{ji} = -z_k^{ij} \). Hence the optimal probing signal is obtained by substituting the result of Eq. 7.43 to Eq. 7.37.

**Special Case:** Suppose \( L(i) = L(j) = 1 \). Then this is the single Gaussian p.d.f. case. Hence, \( a_1^{(i)} = a_1^{(j)} = 1 \), \( m_1^{(i)} = m_1^{(j)} = 0 \), and \( \mathcal{Q}_i = \mathcal{G}_l^{(i)} \) and \( \mathcal{Q}_j = \mathcal{G}_l^{(j)} \).

Substituting these conditions to Eq. 7.43 yields.

\[
\nabla_u J_{k+1}^{ij}(u_k) = \frac{1}{2} \mathcal{G}^{(j)}(z_k^{ij}) \cdot \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \cdot \left\{ \mathcal{G}^{(j)}(z_k^{ij}) \cdot \Sigma^{(j)-1} \cdot z_k^{ij} \right\}
\]

\[
- \frac{1}{2} \mathcal{G}^{(i)}(z_k^{ji}) \cdot \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \cdot \left\{ \mathcal{G}^{(i)}(z_k^{ji}) \cdot \Sigma^{(i)-1} \cdot z_k^{ji} \right\}
\]

\[
= \frac{1}{2} \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \cdot \Sigma^{(j)-1} \cdot z_k^{ij} - \frac{1}{2} \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \cdot \Sigma^{(i)-1} \cdot z_k^{ji}
\]

\[
= \left( \frac{\partial \mathcal{P}_i}{\partial u_k} - \frac{\partial \mathcal{P}_j}{\partial u_k} \right) \cdot \left( \frac{\Sigma^{(j)-1} + \Sigma^{(i)-1}}{2} \right) \cdot (\mathcal{P}_i - \mathcal{P}_j)
\]

Note this becomes the same as the result in Chapter 4 when each dynamics is linear.
Figure 7.2: Generalized Posterior Network
Chapter 8

Simulation

In this chapter, simulational studies for the dynamic anomaly detection are presented. Simulational works are important because the detection time can not be analytically computed.

In Simulation 1, the proposed detection method together with the probing signal generations method are applied to the detection problem which is composed of three linear Gaussian SISO systems. In Simulation 2, the same test is applied to the detection problem which is composed of four linear Gaussian MIMO systems. In Simulation 3, the detection test is performed in the presence of modeling uncertainty. The proposed method will show a better performance if we can identify the statistical properties of the state dependent uncertainties. In Simulation 4, a variable size probing signal will be demonstrated. In simulation 5, the Detection Network is demonstrated. The test will be performed together with the Minimum Variance Control. A non-linear non-Gaussian model is assumed as one of the hypotheses. This is empirically modeled by using GBFNs for a prediction model as well as for a p.d.f. estimation model.

More details about the training procedure of the GBFN is demonstrated in Appendix B. for a predictive module and a p.d.f. estimation module, respectively. In Simulation B1, a surface fitting capability is demonstrated by using GBFNs. Noisefree data is used as training data. In Simulation B2, the same demonstration is shown for the noised data case. In Simulation B2, a p.d.f. estimation capability is demonstrated
by using a generalized GBFN. The MDL method is applied for the model selection.

8.1 Simulation 1 (SISO Linear Gaussian system)

A simulation for the dynamic anomaly detection among the following three hypothesized linear dynamic models was performed.

\[
\begin{align*}
P_1 & \quad \begin{cases}
    H_1 : & y_{k+1} = 0.5y_k + u_k + \epsilon_{k+1}^{(1)} \\
    H_2 : & y_{k+1} = 0.7y_k + 0.6u_k + \epsilon_{k+1}^{(2)} \\
    H_3 : & y_{k+1} = 0.9y_k + 0.6u_k + \epsilon_{k+1}^{(3)}
\end{cases}
\end{align*}
\]

Here \( \epsilon_k^{(i)} \) is a i.i.d. random sequence with \( \epsilon_k^{(i)} \sim N(0, 0.01^2) \) for each \( i = 1, 2, 3 \). \( u_k \) was assumed to be zero. The test conditions used are as follows.

\[
\begin{array}{cccc}
0 & 10 & 200 & 400 \\
\bullet & \bullet & t_s & t_w
\end{array}
\]

(a) Time when the test starts (\( t_s \)) = 10

(b) Time when the system switches \( H_1 \rightarrow H_2 \), (\( t_w \)) = 200

(c) Time when the test ends = 400

(d) \[
\begin{align*}
\text{Probability of detecting } H_i \\
\text{when } H_j \text{ is true, } (\beta_{ij})
\end{align*}
\]
\[
\begin{pmatrix}
0.99 & 0.005 & 0.005 \\
0.005 & 0.99 & 0.005 \\
0.005 & 0.005 & 0.99
\end{pmatrix}
\]

(e) Priors \( P(H_1) = P(H_2) = P(H_3) = 1/3 \)

(f) Thresholds (\( T_{ij} \)) = \[ \log \left( \frac{p(H_i)}{p(H_j)} \right) + \log \left( \frac{\beta_{ii}}{\beta_{ij}} \right) \]

The test is comprised of two things: initial stage detection and model switching detection. A lower bound was set to prevent the excessive time delay in the change detection stage. These lower bounds were set such that they are variable depending on which hypothesis is the strongest one. To be more specific, at time step \( k \), if \( H_i \) was of the largest a posteriori, then the lower bound for the \( j \)-th posterior was set such that

\[
\log p(H_j \mid y^k, u^{k-1}) \geq -2T_{ij}
\]
holds (in other words \( p(H_j \mid y^k, u^{k-1}) \geq (\beta_{ij}/\beta_{ii})^2 \)).

Figure 8.1 shows that the detection did not occur under the selected test condition. Figure 8.1-(a) shows the magnitude of the output process \( y_k \), and Figure 8.1-(c) shows posterior evolution, where posterior probability is slowly changing. Figure 8.1-(d) shows the detected hypothesis (here zero means that the termination law is not activated). Figure 8.2 shows the evolution of the log posterior ratios. We see that the posterior ratio is still wandering in the neutral zone. Two types of probing signals with the same power were tested. Pseudo Random Binary Signal (PRBS) and the signal which maximally discriminates between the relative entropies of the two largest posteriors. This signal will be denoted by Maximal Relative Entropy Signal (MRES). A PRBS was synthesized as

\[
u_k = 0.3 \cdot \text{sign}(v)
\]

where \( v \) is a i.i.d. RV. with \( v \sim N(0, 1) \). This signal has uniform distribution in the frequency domain, and inputting this kind of multiple frequency signal has been commonly used for the identification procedure when we do not have prior information about the system. In synthesizing the MRES, a two step horizon was used in between \( H_2 \) and \( H_3 \), and one step horizons were used in between \( H_1 \) and \( H_2 \), \( H_1 \) and \( H_3 \). Figure 8.3 and Figure 8.5 shows an example when PRBS and MRES were used. Detection occurred in both cases, however output regulation performances were degraded. These figures show that MRES enables faster detection than the PRBS. In Figure 8.5, shows that the probing signal’s frequency changes from high frequency to low frequency after 200sec. Figure 8.4 shows that the log posterior ratio is wandering in the 45 degree direction when the PRBS was used. However, Figure 8.6 shows that the evolution of the log posterior ratio is quickly entered into the decision boundary \( H_2 \), without wandering in the 45 degree direction in the neutral zone. A further simulation
was performed using the same noise data, and the following statistics were obtained.

\[ H_1 \rightarrow H_2 \]

<table>
<thead>
<tr>
<th>Class</th>
<th>SNR</th>
<th>[| u |_{\text{max}}]</th>
<th>CA</th>
<th>FA</th>
<th>WA</th>
<th>MA</th>
<th>m(time)</th>
<th>std(time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No signal</td>
<td>N/A</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>139.8</td>
<td>70.9</td>
</tr>
<tr>
<td>PRBS</td>
<td>6.0</td>
<td>0.02</td>
<td>49</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>70.9</td>
<td>27.8</td>
</tr>
<tr>
<td>MRES</td>
<td>6.0</td>
<td>0.02</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>44.1</td>
<td>7.3</td>
</tr>
<tr>
<td>PRBS</td>
<td>12.0</td>
<td>0.04</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25.7</td>
<td>13.9</td>
</tr>
<tr>
<td>MRES</td>
<td>12.0</td>
<td>0.04</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15.3</td>
<td>3.3</td>
</tr>
<tr>
<td>PRBS</td>
<td>15.6</td>
<td>0.06</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12.1</td>
<td>6.1</td>
</tr>
<tr>
<td>MRES</td>
<td>15.6</td>
<td>0.06</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.8</td>
<td>1.5</td>
</tr>
<tr>
<td>PRBS</td>
<td>18.1</td>
<td>0.08</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.6</td>
<td>3.8</td>
</tr>
<tr>
<td>MRES</td>
<td>18.1</td>
<td>0.08</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.2</td>
<td>1.0</td>
</tr>
</tbody>
</table>

50 repetition. \( SNR(dB) = 10 \log(\| u \|_{\text{max}}^2/\sigma^2) \)

Here CA, FA, WA and MA respectively means correct alarm, false alarm, wrong alarm and missed alarm. An incorrect fault alarm before the model switch was classified as a false alarm. and an incorrect alarm after the model switch was classified as a wrong alarm. When a fault alarm did not occur, this was classified as a missed alarm.

Based on these statistics, the MRES performed better than the PRBS since the average detection time, \( m(\text{time}) \), was smaller, and its standard deviation, \( \text{std}(\text{time}) \), was smaller too. Note when \( \text{std}(\text{time}) \) becomes small, the detection becomes reliable. We can see that the detection performance increases as the power of the probing signal increases.
8.2 Simulation 2 (MIMO Linear Gaussian System)

A simulation for testing the anomaly detection among the following four hypothesized linear MIMO dynamic models was performed.

\[
P_2 \begin{cases}
    H_1 : y_{k+1} = \begin{pmatrix} 0.5 & 1.0 \\ 0 & 0.7 \end{pmatrix} y_k + \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix} u_k + e_{k+1}^{(1)} \\
    H_2 : y_{k+1} = \begin{pmatrix} 0.8 & 1.0 \\ 0 & 0.5 \end{pmatrix} y_k + \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix} u_k + e_{k+1}^{(2)} \\
    H_3 : y_{k+1} = \begin{pmatrix} 0.7 & 1.0 \\ 0 & 0.6 \end{pmatrix} y_k + \begin{pmatrix} 1.0 & 0.5 \\ 0 & 1.0 \end{pmatrix} u_k + e_{k+1}^{(3)} \\
    H_4 : y_{k+1} = \begin{pmatrix} 0.5 & 1.0 \\ 0 & 0.7 \end{pmatrix} y_k + \begin{pmatrix} 0.6 & 0.5 \\ 0 & 0.6 \end{pmatrix} u_k + e_{k+1}^{(4)}
\end{cases}
\]

Here \( e_{k}^{(i)} \) is i.i.d. sequence with \( e_{k}^{(i)} \sim N(0, \text{diag}\{0.01^2, 0.01^2\}) \) for each \( i = 1, 2, 3, 4 \). We assumed that \( \bar{u}_k = 0 \). The test condition used is as follows.

(a) Time when the test starts ( \( t_s \)) = 10
(b) Time when the system switches ( \( t_w \)) = 200
(c) Time when the test ends = 400
(d) \[ \text{Probability of detecting } H_i \text{ when } H_j \text{ is true, } ( [\beta]_{ij} ) \] = \[
\begin{pmatrix}
0.985 & 0.005 & 0.005 & 0.005 \\
0.005 & 0.985 & 0.005 & 0.005 \\
0.005 & 0.005 & 0.985 & 0.005 \\
0.005 & 0.005 & 0.005 & 0.985 \\
\end{pmatrix}
\]
(e) Priors \( P(H_1) = P(H_2) = P(H_3) = P(H_4) = 1/4 \)
(f) Thresholds ( \( T_{ij} \)) = \[ \log \frac{p(H_i)}{p(H_j)} + \log \frac{\beta_{ii}}{\beta_{ij}} = \log \frac{\beta_{ii}}{\beta_{ij}} \]
In this simulation, two different random signals were considered as follows.

\[
\begin{align*}
\text{PRBS1: } u_k &= \begin{pmatrix} c_1 \cdot \text{sign}(v_{1k}) \\ c_1 \cdot \text{sign}(v_{2k}) \end{pmatrix} \\
\text{PRBS2: } u_k &= \begin{pmatrix} c_2 \cdot \cos(w_k) \\ c_2 \cdot \sin(w_k) \end{pmatrix}
\end{align*}
\]

Here \(v_{1k}\) and \(v_{2k}\) are two i.i.d. RVs with \(N(0, 1)\), and \(w_k\) is a i.i.d. RV uniformly distributed in the \([0, 2\pi]\). \(c_1\) and \(c_2\) were set such that \(\| u_k \| = 1\). In obtaining the MRES, two step horizon was used in between \(H_1\) and \(H_2\), and a one step horizon was used in the other cases. The statistic of the test result follows.

\(H_1 \rightarrow H_2\)

<table>
<thead>
<tr>
<th>Class</th>
<th>(| u_k |)</th>
<th>CA</th>
<th>FA</th>
<th>WA</th>
<th>MA</th>
<th>(m) (time)</th>
<th>(\sigma) (time)</th>
<th>(\sigma(y_{1,k}))</th>
<th>(\sigma(y_{2,k}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>No signal</td>
<td>0.01</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>89.7</td>
<td>49.1</td>
<td>0.024</td>
<td>0.01</td>
</tr>
<tr>
<td>PRBS1</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>37.5</td>
<td>21.8</td>
<td>0.033</td>
<td>0.013</td>
</tr>
<tr>
<td>MRES</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21.2</td>
<td>16.8</td>
<td>0.036</td>
<td>0.013</td>
</tr>
<tr>
<td>PRBS2</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17.3</td>
<td>11.5</td>
<td>0.064</td>
<td>0.018</td>
</tr>
</tbody>
</table>

50 repetition. \(SNR=-3.0dB\) for each case

\(H_1 \rightarrow H_3\)

<table>
<thead>
<tr>
<th>Class</th>
<th>(| u |)</th>
<th>CA</th>
<th>FA</th>
<th>WA</th>
<th>MA</th>
<th>(m) (time)</th>
<th>(\sigma) (time)</th>
<th>(\sigma(y_{1,k}))</th>
<th>(\sigma(y_{2,k}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>No signal</td>
<td>0.01</td>
<td>42</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>118.0</td>
<td>38.6</td>
<td>0.023</td>
<td>0.011</td>
</tr>
<tr>
<td>PRBS1</td>
<td>0.01</td>
<td>49</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>55.1</td>
<td>23.9</td>
<td>0.029</td>
<td>0.013</td>
</tr>
<tr>
<td>MRES</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>35.1</td>
<td>16.4</td>
<td>0.031</td>
<td>0.013</td>
</tr>
<tr>
<td>PRBS2</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24.1</td>
<td>11.5</td>
<td>0.056</td>
<td>0.019</td>
</tr>
</tbody>
</table>

50 repetition. \(SNR=-3.0dB\) for each case
Based on these statistics, the PRBS2 can disturb the system more than the PRBS1. In case of \( H_1 \rightarrow H_2 \) and \( H_1 \rightarrow H_3 \), the PRBS2 performed better detection than the MRES, but the regulation capability decreased seriously. Based on these results, the overall performance of the MRES is better than that of the others.

### 8.3 Simulation 3 (Robust Detection in the Linear Gaussian System)

A simulation for testing the dynamic anomaly detection in the presence of modeling uncertainties are considered. The test conditions and hypotheses applied are the same as those of the last simulation (Simulation 2), except that the true \( H_2 \) has modeling uncertainties. The modeling uncertainty considered is

\[
H_2 : \quad y_{k+1} = \begin{pmatrix} 0.8 + \Delta_1 & 1.0 \\ 0 & 0.5 + \Delta_2 \end{pmatrix} y_k + \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix} u_k + \epsilon_{k+1}^{(2)}
\]
where $\Delta_1 \sim N(0, 0.1^2)$ and $\Delta_2 \sim N(0, 0.1^2)$. The following three cases were compared.

1. The true model does not have modeling uncertainty
   The true model has modeling uncertainty
   but is not considered in detection nor in the probing signal generation
   The true model has modeling uncertainty
   and this is considered in the detection
   as well as in the probing signal generation

The test result was

$$H_1 \rightarrow H_2$$

<table>
<thead>
<tr>
<th>Class</th>
<th>$|u|$</th>
<th>CA</th>
<th>FA</th>
<th>WA</th>
<th>MA</th>
<th>$\mu$(time)</th>
<th>$\sigma$(time)</th>
<th>$\sigma(y_{1,k})$</th>
<th>$\sigma(y_{2,k})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) w/ MRES</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>21.2</td>
<td>16.8</td>
<td>0.036</td>
<td>0.013</td>
</tr>
<tr>
<td>(2) w/ MRES</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23.1</td>
<td>20.8</td>
<td>0.037</td>
<td>0.013</td>
</tr>
<tr>
<td>(3) w/ MRES</td>
<td>0.01</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12.6</td>
<td>11.4</td>
<td>0.039</td>
<td>0.013</td>
</tr>
</tbody>
</table>

50 repetitions, $SNR=−3.0dB$ for each case

It is clear that the consideration of the uncertainties for probing signal generation improves the detection performances.

### 8.4 Simulation 4 (Variable Size Probing Signal)

The probing signal can be efficiently applied to the system by varying the signal energy according to the amount of the uncertainties of the posterior distribution. A simulation for the probing signal generation, in which magnitude is self-adjusted, was considered. The test conditions are the same as those in the Simulation 2. The following Entropy function was used for the measure of uncertainties in the posterior distribution at each time step.

$$H(p) = −\sum_{i=1}^{4} p(H_i \mid y^k, u^{k−1}) \cdot \log_e p(H_i \mid y^k, u^{k−1})$$
The power of the probing signal was controlled by the following simple law.

\[
\begin{align*}
\text{if } H(p) &\geq 0.1 \text{ then } \| u_k \| = 0.03 \\
\text{else } \| u_k \| & = 0.0075
\end{align*}
\]

With this condition, when the system's posterior distribution was certain, the probing signal's energy was about half of the energy which was applied in the Simulation 2. However, the energy was raised up to nine times more than that of the energy applied in the Simulation 2 when the above condition was satisfied. The selection of the threshold (here 0.1) was not sensitive in detection, because when the model switched, the entropy changed quickly. Figure 8.7 shows the probing signal and the entropy's evolution. The instantaneous output regulation performance was reduced, but the overall regulation performance was not reduced. With this Variable Size Maximum Relative Entropy Signal (VSMRES), the following results were obtained.

<table>
<thead>
<tr>
<th>Class</th>
<th>Switching</th>
<th>CA</th>
<th>FA</th>
<th>WA</th>
<th>MA</th>
<th>m(time)</th>
<th>(\sigma)(time)</th>
<th>(\sigma(y_{1,k}))</th>
<th>(\sigma(y_{2,k}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSMRES</td>
<td>(H_1 \rightarrow H_2)</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10.9</td>
<td>5.98</td>
<td>0.034</td>
<td>0.013</td>
</tr>
<tr>
<td>VSMRES</td>
<td>(H_2 \rightarrow H_3)</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20.1</td>
<td>13.81</td>
<td>0.012</td>
<td>0.012</td>
</tr>
<tr>
<td>VSMRES</td>
<td>(H_2 \rightarrow H_4)</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47.5</td>
<td>19.02</td>
<td>0.013</td>
<td>0.013</td>
</tr>
</tbody>
</table>

50 repetition

This result shows that a better performance (than that of the Simulation 2) was obtained for the detection purpose as well as for the overall regulation purpose. In practical situations, it is necessary to decide how much the system can tolerate instantaneous performance degradation of the output regulation. This simulation shows an example that the probing signal can be more effective if the signal energy is adjusted in accordance with the amount of uncertainties of the posterior distribution. However, how to adjust the energy is an open problem.
8.5 Simulation 5 (Detection Network)

A simulation for the test of the dynamic anomaly detection among the following four hypothesized model was performed.

\[ H_1 : \quad y_{k+1} = 0.3y_k + 1.0u_k + e^{(1)}_k, \quad e^{(1)}_k \sim N(0, 0.05^2) \]
\[ H_2 : \quad y_{k+1} = 0.9y_k + 0.4u_k + e^{(2)}_k, \quad e^{(2)}_k \sim N(0, 0.05^2) \]
\[ H_3 : \quad y_{k+1} = 0.7y_k + 0.4u_k + e^{(3)}_k, \quad e^{(3)}_k \sim N(0, 0.05^2) \]

\[ H_4 : \begin{cases} 
  y_{k+1} = f(y_k) + 0.5u_k + e^{(4)}_k, \\
  where \quad f(y_k) = \frac{0.5y_k}{0.45 + y_k^2} \\
  e^{(4)}_k \sim \frac{1}{3}N(0, 0.12^2) + \frac{1}{3}N(0, 0.03^2) + \frac{1}{3}N(0, 0.01^2) 
\end{cases} \]

Here \( e^{(i)}_k \) is zero mean i.i.d. noise process for each \( i = 1, 2, 3, 4 \). Note that \( H_4 \) has non-linear dynamics, non-Gaussian additive noise, and three fixed points \{-\sqrt{0.05}(stable), 0(unstable), \sqrt{0.05}(stable)\}. The following predictive modules and p.d.f. modules were assumed to be known and were used for each hypothesized model.

**Prediction modules**

\[ H_1 : \quad \hat{y}_{k+1|k} = 0.3y_k + 1.0u_k \]
\[ H_2 : \quad \hat{y}_{k+1|k} = 0.9y_k + 0.4u_k \]
\[ H_3 : \quad \hat{y}_{k+1|k} = 0.7y_k + 0.4u_k \]
\[ H_4 : \begin{cases} 
  \hat{y}_{k+1|k} = \mathcal{P}(y_k) + 0.5u_k \\
  where \quad \mathcal{P} \text{ is a GBRF with unknown number of kernels} 
\end{cases} \]

**p.d.f. modules**

\[ H_1 : \quad N(0, 0.05^2) \]
\[ H_2 : \quad N(0, 0.05^2) \]
\[ H_3 : \quad N(0, 0.05^2) \]
\[ H_4 : \begin{cases} 
  Q(e^{(4)}_k) \\
  which \text{ is a GBFN with unknown number of kernels} 
\end{cases} \]
Identification: In the identification stage of $H_4$, a uniform random input was used to excite the system. The input magnitude was determined such that the output range covers the operating range, by trial and error method. Actually the training range was set such that it is 20~30% larger than the operating range, because Neural Network’s fitting performance tend to be degraded in the region where input population is small. This usually can occur in the edge of the domain. As a model selection method among different representations, the Validation Test was used in the fitting problem and the MDL method was used in the p.d.f. estimation problem. A total of 1000 samples was assumed to be available. In the fitting problem 500 samples were used for training data and the other 500 samples were used for model selection. In the p.d.f. estimation problem, 1000 samples were used for the training process since the MDL method does not require the evaluation of the representation performed by different samples. The parameters of the GBFN was tuned by minimizing the following LMS criteria.

$$\frac{1}{2} \sum_{k=1}^{500} \| y_k - 0.5u_{k-1} - P(y_{k-1}; \theta) \|^2$$

More details about the training procedure is demonstrated in the Appendix 2 and Appendix 3 for a predictive module and a p.d.f. estimation module, respectively.

The p.d.f. estimation was performed by using a GBFN. The residual process

$$\{ y_k - 0.5u_{k-1} - P(y_{k-1}; \tilde{\theta}) \}_{k=1}^{1000}$$

was used as training samples, and the Maximum Likelihood method incorporated with the Expectation Maximization Algorithm was used as a training method. Figure 8.9 shows the true p.d.f. of the noise process and estimated p.d.f. of the residual process by using the tuned GBFN. Figure 8.10 shows the result of the p.d.f. estimation performed by using a single Gaussian distribution. It is a fact that if the p.d.f. estimation is performed by using one Gaussian kernel, then the result can be sensitive
to the less probable samples. In this case study, the result of the GBRF estimation method outperformed the results of the single Gaussian estimation method.

**Control:** The nominal input was set in accordance with the MVC law. To be more specific, if the dynamics is

\[
y_{k+1} = g^{(i)}(y_k) + b^{(i)}u_k + e^{(i)}_{k+1}
\]

then the MVC input is

\[
u_k = \frac{g^{(i)}(y_k)}{b^{(i)}}
\]

such that \(y_{k+1}\) has a minimum variance by this control input. Here \(i\) was determined by the detected hypothesis, thus it may be correct or it may be incorrect.

**Detection:** The test condition used is as follows.

\[
\begin{array}{c|c|c|c}
0 & 10 & 500 & 1000 \\
\hline
\hline
\end{array}
\]

- (a) Time when the test starts \((t_s) = 10\)
- (b) Time when the system switches \((t_w) = 500\)
- (c) Time when the test ends = 1000
- (d) \[
\left\{ \begin{array}{c}
\text{Probability of detecting } H_i \\
\text{when } H_j \text{ is true, } [\beta]_{ij}
\end{array} \right\} = \begin{pmatrix}
0.985 & 0.005 & 0.005 & 0.005 \\
0.005 & 0.985 & 0.005 & 0.005 \\
0.005 & 0.005 & 0.985 & 0.005 \\
0.005 & 0.005 & 0.005 & 0.985 \\
\end{pmatrix}
\]
- (e) Priors \(P(H_1) = P(H_2) = P(H_3) = P(H_4) = 1/4\)
- (f) Threshholds \([T_{ij}] = \log \left( \frac{p(H_i)}{p(H_j)} \right) + \log \frac{\beta_{ii}}{\beta_{ij}} = \log \frac{\beta_{ii}}{\beta_{ij}}\)

Figure 8.11 shows the result of the control and the detection when the model switch occurred from \(H_1\) to \(H_2\) at \(t = 500\). The MVC signal was applied but no
probing signal was used. Detection occurred 248 sec after the occurrence of the model switch. Thus for the time steps between 500-748 sec, the MVC was performed by an incorrectly recognized model. Accordingly, the output was not white signal for this duration since there was error dynamics. The output variances were \( L_1 = 5.13 \cdot 10^{-2} \) in the first stage (i.e., 1-500 sec) and \( L_2 = 6.27 \cdot 10^{-2} \) in the second stage (i.e., 501-1000 sec).

Figure 8.12 shows the result of the control and the detection when the same conditions were applied except the probing signal. The size of the probing signal was adjusted as follows.

\[
\begin{align*}
\text{if entropy} > E, & \quad \| \tilde{u}_k \| = 0.1/b^{(*)} \\
\text{otherwise,} & \quad \| \tilde{u}_k \| = 0.02/b^{(*)}
\end{align*}
\]

where \( E \) was set as 0.2 by trial and error, and \( b^{(*)} \) denotes the parameter of the detected model as defined in equation 8.1. The division by \( b^{(*)} \) was performed to uniformize the sensitivity of the probing signal to each model. Detection occurred 27 sec after the occurrence of the model switch. Thus, the portion of time to which the incorrect MVC was applied was much smaller than that of the time in Figure 8.11. The output variances were \( L_1 = 5.75 \cdot 10^{-2} \) in the first stage and \( L_2 = 5.54 \cdot 10^{-2} \) in the second stage.

These results show that the probing signal degraded the regulation performance in the first stage, but it improved the regulation performance in the second stage by correctly recognizing the present dynamic mode. The overall regulation performance became uniform with the aid of the probing signal. Thus the probing signal was helpful in this case.

Figure 8.13 shows the result of the control and detection when the model switch occurred from \( H_1 \) to \( H_4 \) at \( t = 500 \). Neither the MVC signal nor the probing signal was inputted. Detection occurred at 25 sec after the occurrence of the model switch. Concerning the second stage, although detection occurred, since no control signal was inputted, the output is fluctuating between two non-zero equilibrium points. Thus the output regulation performance was inferior.

Figure 8.14 shows the result of the control and the detection when the same conditions were applied except for the MVC. Detection occurred at 22 sec after the
occurrence of model switch. Thus, the portion of time to which the incorrect MVC was applied was only 22 seconds, and for the time $522 < t < 1000$ sec the correct MVC was being applied. This shows how well the proposed Detection Network performs its task and shows an important illustration that the MVC can be well applied by using the GBFN. The regulation performance in the second stage was better than that of the case when the probing signal was applied. The main reason is that the detection occurred quickly even without the use of the probing signal in this case.

However, the probing signal does not guarantee a better regulation performance than that of the incorrect MVC. Although the MVC aims the zero dynamics, it is only possible when the true dynamics are accurately identified. Hence there will be a nonzero error dynamics when the true model is not consistent with the detected model. In some cases, this error dynamics can be smaller than the effect of the probing signal. Hence there will be a trade-off between the size of the error dynamics and the size of the probing signal. (To be more specific, if the size of the probing signal is too small, then the detection performance will be limited. If the size of the probing signal is large, then the regulation performance can be even worse than the performance degradation caused by the error dynamics. This phenomenon was observed in other simulations.) Thus a necessary condition for the size of probing signal is required and is derived as follows.

Suppose the detection is performed between the following two first-order time-invariant linear dynamic systems.

\begin{align*}
    H_i : \quad y_{k+1} &= a^{(i)}y_k + b^{(i)}u_k + e_{k+1}^{(i)} \\
    H_j : \quad y_{k+1} &= a^{(j)}y_k + b^{(j)}u_k + e_{k+1}^{(j)}
\end{align*}

where $e_{k+1}^{(i)}$ and $e_{k+1}^{(j)}$ are white i.i.d. noise processes with variance $R^{(i)}$ and $R^{(j)}$ respectively. The MVC for the $i$-th model is

$$u_k^{MV} = -\frac{a^{(i)}}{b^{(i)}}y_k$$
Suppose the probing signal is not used and correct detection does not occurred. Then the dynamics caused by the incorrect MVC is

\[ y_{k+1} = a^{(j)} y_k + b^{(j)} \left( -\frac{a^{(i)}}{b^{(i)}} u_k \right) + e^{(j)}_{k+1} \]

\[ = Ey_k + e^{(j)}_{k+1} \]

where \( E = a^{(j)} - \frac{b^{(i)}}{b^{(j)}} a^{(i)} \). Hence \( y_{k+1} \) can be represented as follows.

\[ y_{k+1} = E^k y_0 + \left( E^{k-1} e^{(j)}_1 + \cdots + E e^{(j)}_k + e^{(j)}_{k+1} \right) \]

Thus its variance can be obtained as follows.

\[ \text{var}(y_{k+1}) = E^{2k} \text{var}(y_0) + \left( E^{2(k-1)} + \cdots + E^2 + 1 \right) R^{(j)} \]

\[ = E^{2k} \text{var}(y_0) + \frac{1 - E^{2k}}{1 - E^2} R^{(j)} \]

Assuming \(^1\) \(|E| < 1\), the output variance becomes asymptotically

\[ \text{var}(y_{k+1}) \xrightarrow{k \to \infty} \frac{R^{(j)}}{1 - E^2} \]

Hence there exists the following increase in the variance caused by the incorrect MVC.

\[ \Delta \text{var}(y_{k+1}) \xrightarrow{k \to \infty} R^{(j)} - \frac{R^{(j)}}{1 - E^2} = \frac{E^2 R^{(j)}}{1 - E^2} \]

On the other hand, suppose the probing signal enabled the correct detection. Then the system equation will be as follows.

\[ y_{k+1} = a^{(j)} y_k + b^{(j)} (\bar{u}_k + \bar{\bar{u}}_k) + e^{(j)}_{k+1} \]

Suppose the probing signal is \( \bar{u}_k = U_k / b^{(j)} \) where \( U_k \) is MRES with fixed magnitude. Hence

\[ y_{k+1} = a^{(j)} y_k + b^{(j)} \left( -\frac{a^{(i)}}{b^{(i)}} u_k + \frac{U_k}{b^{(j)}} \right) + e^{(j)}_{k+1} \]

\[ = U_k + e^{(j)}_{k+1} \]

\(^1\)If the error dynamics is unstable then it will increases the output variance and detection will occur.
The probing signal can be modeled as the following two point valued RV.

\[ U_k = \begin{pmatrix} U_{\text{max}} \\ -U_{\text{max}} \end{pmatrix} \quad EU_k = 0 \]

Thus the increase in the variance caused by the probing signal is

\[ \Delta \text{var}(y_{k+1}) = U_{\text{max}}^2. \]

The variance increase caused by the probing signal should be smaller than that caused by the incorrect MVC. Thus the following necessary condition can be obtained.

\[ U_k^2 < \sqrt{\frac{E^2 R^{(j)}}{1 - E^2}} \]

where \( E = a^{(j)} - \frac{y^{(j)}}{y^{(j)}} a^{(i)}. \)
Figure 8.1: An example of the detection result when there is no probing signal.
Figure 8.2: Evolution of the log posterior ratio when there is no probing signal.
Figure 8.3: An example of the detection when PRBS is applied.
Figure 8.4: Evolution of the log posterior ratios when PRBS is applied.
Figure 8.5: An example of the detection result when MRES is applied.
Figure 8.6: Evolution of the log posterior ratios when MRES is applied.
Figure 8.7: An example of the detection result when VSMRES is applied.
Figure 8.8: Result of the Curve Fitting by using a GBN.
Figure 8.9: Result of the p.d.f. estimation by using a GBN.
Figure 8.10: Result of the p.d.f. estimation when one Gaussian kernel is used.
Figure 8.11: Result of the MVC and the detection without the probing signal when the model switched from H1 to H2.
Figure 8.12: Result of the MVC and the detection with probing signal when the model switched from H1 to H2.
Figure 8.13: Result of the output and the detection when the model switched from H1 to H4. No control and no probing signal was used.
Figure 8.14: Result of the MVC and detection without the probing signal when model switched from H1 to H4.
Chapter 9

Conclusions

9.1 Thesis Sketch

This thesis aimed at the class of detection problems which was able to work in hard conditions but required fast detection. To be more specific, the classes of dynamics where some anomalies was highly non-linear, noises were seriously considered, and output dimension was small were concerned.

The first step to FDI synthesis toward our intention was the use of observation space. Feature vectors were output estimation errors rather than state estimation errors or parameters, and detection was performed over the observation space rather than over state space or over parameter space.

In Chapter 3 a new detection mechanism was developed where decision conditions could be adjustable by the prescribed probabilities of incorrect detection. This mechanism had the Bayesian property, was applicable to the M-ary detection problem, and needed only \( M - 1 \) comparisons to make a decision. It required the computation of each posterior probability and a decision was made based on the predetermined probabilities of incorrect detection. To be more specific, the detector alerts at the time the strongest hypothesis suppresses all the other hypotheses with a certain amount of information. The overall performance of the proposed detection system depends on the speed at which information that is capable of distinguishing the failure modes is growing in the statistical sense. The minimum time detection
property, in the sense of Bayesian risk, could not be shown yet and is left as an open problem. This was explained in Chapter 3.

In most cases, rather than terminate the detection task after a detection occurred, the test should be continued so that abnormal behaviors can be identified if they occur. To accomplish this, the Bayesian detection can be continued. In that case, lower bounds should be set to the posterior probabilities to avoid the excessive time delay in the change detection stage. This insures that when a model switches the posterior probabilities can start their changes from reasonable initial values. However, this leads to a problem in designing the detection procedure: there is a trade-off between the probability of an incorrect detection at the beginning stage and that of a change detection in the later stages of the detection process. This undesirable behavior can be more or less overcome by estimating the source hypothesis and the destination hypothesis (i.e. by monitoring from where to where the change occurs) and estimating the time. Rather than give a mathematical justification, a heuristic method was studied in Chapter 5. One disadvantage of the proposed method is that it requires a large amount of computation. More studies should be followed to change detection problem in the $M$-ary detection problems.

Often linear mathematical models are preferred (possibly with some terms representing uncertainties) if the uncertainties are not large. The reason for this is that the linear equations are beneficial in processing the data and hypotheses because of their simplicity. This uncertainties are modeled by additive i.i.d. noise terms. However there are often state dependent uncertainties that can not be effectively modeled by such an additive i.i.d. noise process. A parametrically structured uncertainty was considered for this class of uncertainty, and this was modeled by Gaussian random variables. The advantage of such an approach is that the noise process depends explicitly on the variance parameters, and this automatically gives different contributions to the net information at each time step. If the corresponding parameters of the state dependent uncertainties can be identified, then the proposed approach gives even better detection performance. More detailed notions are explained in Chapter 6.
The overall performance of the detection system depends on the speed at which information that is capable of distinguishing the failure modes is growing. The rate at which information is growing can be increased if an input signal for the system can be effectively selected. This signal was called as 'probing signal'. To insure that the probing does not overwhelm the regulatory input, there should be a (time varying) bound for the magnitude of the probing input. Even if the power of such probing signal is small, the amount of the information that is delivered can be great if the signal’s direction is selected appropriately. Generally, as the number of hypothesized models increases, the probing signal’s responsibility increases. For example, if there is one hypothesized normal model and five different anomaly models then there will be \( \binom{1+5}{2} = 15 \) different conditional relative entropies. The underlying design policy was to avoid the probing signal's responsibility in distinguishing between two or more impossible models. The critical information that is necessary for efficient detection depends on the operating situation, and only the necessary information is required for fast detection. The proposed idea for selecting the probing signal in an \( M \)-ary detection problem was to find the signal that maximizes the conditional relative entropy between future output distributions of the two most plausible models. By this way, a time domain synthesis for the probing signal into the feedback form was established in Chapter 4. A probing signal with variable power can be effective by monitoring the entropy of the posterior distribution.

In order to obtain the posterior probabilities and compute the likelihood functions, predictive models are required and linear Gaussian models are commonly used. However in real systems, such linear models might not accurately represent the dynamic behavior associated with each failure mode. There is a lack of sufficient mathematical results to deal with non-linear non-Gaussian type models, thus empirical approaches are required. Mixed structures of models, where some models are of the linear Gaussian type and some are of the nonlinear non-Gaussian type, may be important in practical situations. This can be accomplished by using an information based detection approach, where detections are made probabilistically in the obser-
vation space. The detection procedure could be synthesized without changing the basic working mechanism by replacing each linear Gaussian model with an improved model, if necessary. For these reasons, we developed a Detection Network which can accommodate various generalized models including non-linear, non-Gaussian models as well as linear Gaussian models.

For empirical non-linear, non-Gaussian models in the Detection Network, we considered Neural Networks (NN). Highly complex nonlinear functions can be closely approximated by such networks using compositions of smooth functions. NNs provide a means for general functional approximation, by using a distributed information structure with appropriate memories. By using such NNs, we can build prediction models (using supervised learning methods) and p.d.f. models (using unsupervised learning methods). Parametric method was considered to reduce the computational burden. This was due to the reason that we needed real time evaluation of both the predictive model and the p.d.f. model for each hypothesis, and we needed computation of the Jacobian matrix. Various Gaussian Basis Function Networks (GBFNs) were established by constraining the parameters in the representation and corresponding training laws were derived. For example, in multi-variate case, GBRF can be a linear superposition of the independent ellipses or can be a linear superposition of the dependent circles. More details are given in Chapter 7.

Estimating p.d.f.s are very important because it determines the net information that can be obtained from samples. A simple Gaussian assumption may not be adequate in some cases. In obtaining the p.d.f. estimate by using the GBFN, the Maximum Likelihood (ML) estimation method with Expectation Maximization (EM) algorithms was applied. The ML estimation method minimizes the Kullback distance (or relative entropy) from the true distribution. However, the ML method, if used alone, does not provide a sufficient answer since it gives simultaneous nonlinear equations to be solved. An effective method was developed to address this problem: the EM algorithm. This method was developed for the incomplete data processing problem like ‘credit-assignment’ problem, and was applied to a wide range of super-
vised and unsupervised learning problems. Samples are realizations of the underlying p.d.f., and this can be viewed as a two step procedure. First one component of the mixture is selected and a sample is drawn from the distribution of that component. By using this algorithm, training laws for various kinds of GBFNs were derived in Chapter 7.

Even in the GBFN, there are additional degrees of freedom: the number of kernels and the type of the base function. These are important because under-parameterization and over-parameterization can occur when the samples are insufficient. Because every sample is contaminated with noise, this often results in over-fitting or under-fitting the data in the training procedure, so the generalization performance of the network is degraded. This lack of ability to generalize is certainly detrimental to the detection performance. One way to save was the use of the Minimum Description Length (MDL) approach.

In information theory, the MDL principle has been studied as an effective way of measuring the descriptive complexity. The descriptive complexity means the length of the minimum required codes such that the observed data sequence is reconstructable. The MDL criteria is a method of model selection criteria which leads to a good trade-off between the likelihood and the model complexity. The MDL method had a Bayesian interpretation. Maximizing the log-likelihood from the estimation's point of view was equivalent to minimizing the data code length from the coding's point of view. This MDL approach was applied to the p.d.f. model selection problems in various GBFNs, in Chapter 7.

### 9.2 Contributions

A new detection mechanism was formulated for $M$-ary detection problems. This formulation needs only $M-1$ comparisons and allows the detection conditions to be adjustable in accordance with the prescribed probabilities of incorrect detection. Also a heuristic model switching detection algorithm and a time estimation algorithm were developed.
A new time domain synthesis for the selection of the probing signal in $M$-ary detection problem was performed and demonstrated. A good compromise between the simplicity and the performance is to find the signal that maximizes the conditional relative entropy between future output distributions of the two most plausible model.

A detection method for the linear model in the presence of state dependent uncertainties was studied and was demonstrated by modeling these uncertainties as Gaussian random variables.

A new Detection Network was developed and demonstrated for improved detection in which non-linear non-Gaussian models can be involved as well as linear Gaussian models. Two requirements for each model to join the proposed Detection Network are: each model should be a predictive model and detection should be performed over the observation space. Non-linear, non-Gaussian models can join this Detection Network by modeling as a parametric Gaussian-Basis Function Network (GBFN).

P.d.f. estimation methods for various classes of GBFN were derived by using the Maximum Likelihood estimation method with the Expectation Maximization algorithm. A model selection method was studied and demonstrated by applying the Minimum Description Length principle in various GBFNs.

### 9.3 Unsolved Problems and Future Work

In this thesis, many important algorithms were heuristically presented but the optimality and the convergence properties could not be proved mathematically. To prove strong theoretic claims about the proposed ideas, these problems should be clearly identified. These unsolved problems not only consist of the main body of this thesis, but offer a challenge as well. The main difficulty in proving results is that the situation that the number of samples to reach a decision boundary can not be analytically calculated. The open problems that need to be solved are as follows.
(1) Consider the following class of predictive dynamics.

\[ H_1: \quad y_{k+1}^{(1)} = f_{k+1}^{(1)}(y_k, u_k) + n_{k+1}^{(1)} \]

\[ \vdots \]

\[ H_M: \quad y_{k+1}^{(M)} = f_{k+1}^{(M)}(y_k, u_k) + n_{k+1}^{(M)} \]

where \( f_{k+1}^{(1)}, \ldots, f_{k+1}^{(M)} \) are known deterministic maps, and \( n_{k+1}^{(1)}, \ldots, n_{k+1}^{(M)} \) are i.i.d. white noises with known distributions. The optimal \( M \)-ary sequential decision rule needs to be mathematically solved. This is the most important part in this thesis. The formulation can be accomplished by minimizing the following Bayesian risk as explained in Chapter 3.

\[
\min_{\phi, \delta} \ p(H_1)R_1(\phi, \delta) + \cdots + p(H_M)R_M(\phi, \delta)
\]

s.t. \[ \sum_{i=1}^{M} p(H_i) = 1 \]

\[ \phi_0 = 0 \]

where \( \phi \) is the stopping rule, \( \delta \) is the decision rule, \( p(H_1), \ldots, p(H_M) \) denote the prior probabilities, and \( R_1, \ldots, R_M \) denote the penalty functions, defined by

\[ R_1(\phi, \delta) = C_1 \mathbb{E}\{ \| \delta_N(y^N) - E_1 \|_{L_1} \mid H_1 \} + C \mathbb{E}\{ N \mid H_1 \} \]

\[ R_2(\phi, \delta) = C_2 \mathbb{E}\{ \| \delta_N(y^N) - E_2 \|_{L_1} \mid H_2 \} + C \mathbb{E}\{ N \mid H_2 \} \]

\[ \vdots \]

\[ R_M(\phi, \delta) = C_M \mathbb{E}\{ \| \delta_N(y^N) - E_M \|_{L_1} \mid H_M \} + C \mathbb{E}\{ N \mid H_M \} \].

Here, \( C > 0 \) is the cost per sample, \( C_1, \ldots, C_M > 0 \) are the costs for incorrect decisions. \( N \) is the stopping time, \( \| \cdot \|_{L_1} \) denotes the \( L_1 \) norm, and \( E_1, E_2, \ldots, E_M \) denote the unit vectors in the \( M \) dimensional Euclidean space. In Sequential-Bayesian problems, the only case that was mathematically solved is that in which samples \( \{y_k\}_{k=0}^{\infty} \) are independent and identically distributed. In this case, the Average Number of Samples (ANS) could be calculated. However in the detection problem, samples are not independent (they are correlated each other since they are collected from the memory system) and calculation of the ANS is not certain at this time. This makes the proof more difficult and thus the solution has not been discovered for more than a quarter of a century since Wald's first discovery of the SPRT and Fugerson's work about the
Bayesian interpretation (1967).

(2) If the above Bayesian problem can be mathematically solved then the optimal Detection Network can be developed under the following conditions.

\[
\begin{pmatrix}
  f_{k+1}^{(1)}(y^k, u^k) = P_1(y^k, u^k) \\
  \vdots \\
  f_{k+1}^{(M)}(y^k, u^k) = P_M(y^k, u^k)
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
  p(n_{k+1}^{(1)}) = Q_1(n_{k+1}^{(1)}) \\
  \vdots \\
  p(n_{k+1}^{(M)}) = Q_M(n_{k+1}^{(M)})
\end{pmatrix}
\]

where \( P_1, \ldots, P_M \) denote a predictive module for each hypothesis and \( Q_1, \ldots, Q_M \) denote a conditional p.d.f. module for each hypothesis.

(3) A minimum detection time property and a convergence of the proposed algorithm in Chapter 5 need to be proved as well. The algorithm shown in Chapter 5 was established for change detections, hence the source hypothesis is assumed to be known. This seemed to make the proof easier than that of the Bayesian problem, but it still could not be handled in this \( M \)-ary detection system.

(4) A study of the detection mechanism whose decision boundaries are smooth needs to be continued. The detection formulation which is based on the 'probabilities of the incorrect decisions' gives sharp decision boundaries in the space of the log-posterior-ratio, as in Figure 3.1. However, for example, if the problem is formulated based on the 'Entropy' then the decision boundary becomes smooth as in Figure 9.1. Generally the relation between the probability of the incorrect decision and the entropy is not one-to-one. More studies should be continued concerning the smooth decision boundaries and the difference between the two.

(5) In the simulation chapter we illustrated the possible usage incorporating the Minimum Variance Control (MVC). How to synthesis the probing signal in relation to this MVC is an open problem and should be continued. One example is that if the formulation is performed such that the detection time is uniform with respect to the chance of dynamic anomalies, then the regulation performance can not
be uniform. On the other hand, if the probing signal is synthesized such that the regulation performance is uniform, then the detection time cannot be uniform with respect to the chance of the dynamic anomalies.

(6) Given data, the study to find the best representation of the GBNF should be continued because this is an open problem. This should include the study about the number of kernels and the class of kernels required. This is applicable to supervised learning problems and unsupervised learning problems. These problems are now extensively being studied by many researchers in various fields.

(7) The proposed detection approach used the assumption of the independence between the noise and the state. However, in some cases, noise processes can be strongly state-dependent. New detection mechanisms should be developed to handle this detection problem and it should be compatible to the state-independent detection problem, because different anomalies can have different types of noise processes.
Figure 9.1: Entropy levels and decision boundaries
Appendix A

Mathematical Preliminaries

A.1 A notion for Probability

Suppose we are collecting samples or observations from a sample set $\Omega$. The specification of probability distribution on the sample set $\Omega$ requires the assignment of probabilities to subsets of $\Omega$. For some sample space of interest it is not possible to assign consistent probabilities to all subsets of $\Omega$; thus we associate with $\Omega$ a collection of events $F$ to which we wish to assign a probabilities. Therefore a probability space is defined as $(\Omega, F, P)$ where $\Omega$ is the sample space, $F$ is the collection of events and $P$ is a probability measure on $F$. The random variables are then functions defined on $\Omega$. For analytical reasons, we assume that the collection $F$ is a $\sigma$- algebra of subsets of $\Omega$; that is we assume that $F$ contains all complements and countable unions of its members (i.e, $F$ has a property that $A \in F$ implies $A^c \in F$ and $A_i \in F$, for $\forall i$ implies $\bigcup_{i=1}^{\infty} A_i \in F$.)

Suppose $X$ is a RV\(^1\) defined on $\Omega$, then $X(w)$ denotes the particular value taken by the random variable when the sample point $w \in \Omega$ is realized. The probability distribution (or probability measure) of this RV is induced by $P$.

$$\Pr\{X \in X\} \triangleq P\{X \mid X(w) \in X\} \triangleq P_X(X)$$

\(^1\)We will use 'RV' to abbreviate 'random variable' in the one dimensional case or 'random vector' in the multi dimensional case.
If the probability measure is continuous, the RV is said to be continuous and if the probability measure is discrete, the RV is said to be discrete.

For a discrete observation space \((\mathcal{A}, 2^{\mathcal{A}}, P)\), probabilities can be assigned to subsets of alphabet \(\mathcal{A}\) in terms of probability mass function\(^2\), \(p : \Omega \rightarrow [0, 1]\), according to

\[
p(x) = \Pr\{X = x\} \quad x \in \mathcal{A}
\]

and has a probability distribution function\(^3\)

\[
\mathbb{P}_X(X) = \Pr\{X \in X\} = \sum_{x_i \in X} p(x_i) \quad X \in 2^{\mathcal{A}}
\]

where \(\mathbb{P}(X)\) denote the probability that the observation \(x\) lie in the set \(X\).

For continuous sample space \((\mathbb{R}^n, \mathcal{B}^n, P)\), continuous\(^4\) RVs are assigned in terms of probability density function \(p : \mathbb{R}^n \rightarrow [0, \infty)\) by way of

\[
\mathbb{P}_X(X) = \Pr\{X \in X\} = \int_X p_X(x) \, dx \quad X \in \mathcal{B}^n
\]

We let \(p\) denote the p.d.f. of a continuous RV case and p.m.f. of a discrete RV.

Let \(U\) and \(V\) be a \(n_1\) and \(n_2\) dimensional RVs respectively. The joint probability distribution of \((U, V)\) is denoted by \(P_{U,V}(\cdot, \cdot)\). This is a probability measure on \(\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}\) such that for subsets \(U \subset \mathbb{R}^{n_1}\) and \(V \subset \mathbb{R}^{n_2}\)

\[
P_{U,V}(U \times V) = \Pr\{U \in U \text{ and } V \in V\}
\]

The mean, or expected value of any function \(g\) of \((U, V)\) is

\[
\mathbb{E}g(U, V) = \int_{\mathbb{R}^{n_1}} \int_{\mathbb{R}^{n_2}} g(u, v) \cdot P_{U,V}(du \times dv)
\]

In particular, the mean of \(U\) and \(V\) denoted by \(m(U)\) and \(m(V)\) respectively are given by \(m(U) \doteq \mathbb{E}U\) and \(m(V) \doteq \mathbb{E}V\), respectively with variances \(\sigma^2(U) = \mathbb{E} |U - m(U)|^2\)

\(^2\)We will use ‘p.m.f.’ to abbreviate ‘probability mass function’

\(^3\)We will use ‘p.d.f.’ to abbreviate ‘probability density function’

\(^4\)absolutely continuous probability measure with respect to Lebesgue measure
and $\sigma^2(V) = E[|V - m(V)|^2]$, respectively. Here $|\cdot|$ is a Euclidean norm. The covariances of $U$ and $V$ are the matrix

\[
\text{cov}(U) = E(U - m(U))(U - m(U))^T \\
\text{cov}(V) = E(V - m(V))(V - m(V))^T
\]

and the covariance of $(U, V)$ is the $n_1 \times n_2$ matrix

\[
\text{cov}(U, V) = E(U - m(U))(V - m(V))^T
\]

If $(U, V)$ has a probability density, denoted by $p_{U,V}(\cdot, \cdot)$, then

\[
P_{U,V}(U \times V) = \int_U \int_V p_{U,V}(u, v) du \cdot dv
\]

and

\[
Eg(U, V) = \int_{\mathbb{R}^{n_1}} \int_{\mathbb{R}^{n_2}} g(u, v) \cdot p_{U,V}(u, v) \, du \, dv
\]

Similarly, the marginal probability densities of $U$ and $V$ respectively,

\[
p_U(u) = \int_{\mathbb{R}^{n_2}} p_{U,V}(u, v) dv, \quad p_V(V) = \int_{\mathbb{R}^{n_1}} p_{U,V}(u, v) du
\]

One can obtain the marginal distributions $P_U$ and $P_V$ from $P_{U,V}$ by

\[
P_U(U) = P_{U,V}(U \times \mathbb{R}^{n_2}), \quad P_V(V) = P_{U,V}(\mathbb{V} \times \mathbb{R}^{n_1})
\]

The conditional distribution of $U$ given $V$, denoted by $P_{U|V}(\cdot|\cdot)$, is a probability measure on $\mathbb{R}^{n_1}$ such that for subsets $U$ and $V$

\[
P_{U|V}(U \times V) = \int_V P_{U|V}(U|v) \cdot P_V(dv)
\]

If the relevant densities exist, the conditional density is

\[
p_{U|V}(u|v) = \frac{p_{U,V}(u, v)}{p_V(v)}
\]

This representation of the conditional density is a version of Bayes' rule. The conditional expectation of $g(U, V)$ given $V = v$ is denoted by $E\{g(U, V) | v\}$ and is defined by

\[
E\{g(U, V)|v\} = \int g(u, v) \cdot P_{U|V}(du | v) = \int g(u, v) \cdot P_{U|V}(u | v) \cdot du
\]
Note that $E\{g \mid v\}$ is a measurable function of $v$. (i.e a random variable.)

We can define the same properties for a discrete RV where the $\sum$ replaces the $\int$ formulas including expectations. Hereafter for convenience, we will denote the p.d.f. for a continuous RV by $p(x)$ rather than $p_X(x)$, and the p.m.f. for a discrete RV $p(x)$ rather than $p_X(x)$.

### A.2 The notion of Information

Suppose $X$ is a discrete RV over the alphabet $\{1, \ldots, M\}$, with p.m.f. $\{p(1), p(2), \ldots, p(M)\}$. The entropy $H(X)$ of a discrete RV $X$ is often used as a measure of uncertainty. $H(X)$ is defined by

$$H(X) = -\sum_{x=1}^{M} p(x) \log p(x) = E_p \log \frac{1}{p(X)}$$

When the log is the base 2, the entropy is measured in bits. Entropy is a function of the distribution of the RV and is a measure of the amount of information required on the average to describe the RV. We will use the notation $H(p)$ to denote the entropy when the distribution is to be emphasized.

If $(X, Y)$ is a discrete RV with $(X, Y) \sim p(x, y)$, $X \in A_X$, $Y \in A_Y$ then the joint entropy $H(X, Y)$ of the discrete RV $(X, Y)$ is defined as

$$H(X, Y) = -\sum_{x \in A_X} \sum_{y \in A_Y} p(x, y) \log p(x, y) = E_{p(x,y)} \log \frac{1}{p(X,Y)}$$

Note $H(X, Y) = H(Y, X)$.

The uncertainty of discrete RV $X$ given another RV $Y = y$ is usually measured by the conditional entropy, or the equivocation. The random variable $Y$ may be discrete or continuous. If $(X, Y)$ is a discrete RV with $(X, Y) \sim p(x, y)$. $X \in A_X$, $Y \in A_Y$, then the conditional entropy $H(X \mid Y)$ is defined as

$$H(X \mid Y) = \sum_{x \in A_X} p(y) H(X \mid Y = y) = E_{p(x,y)} \log \frac{1}{p(X \mid Y)}$$

Note that $H(Y \mid X) \neq H(X \mid Y)$. The relationship between the joint entropy and the conditional entropy is given by the so called Chain rule.

$$H(X, Y) = H(Y) + H(X \mid Y)$$
This Chain rule can also be applied to the conditional entropy, that is

\[ H(X, Y \mid Z) = H(X \mid Z) + H(Y \mid X, Z) \]

If \( Y \) is a continuous RV over an arbitrary sample space with a probability distribution \( P(y) \), then we define the conditional entropy as

\[ H(X \mid Y) = - \int \sum_x p(x \mid y) \cdot \log p(x \mid y) \cdot dP(y) \]

\[ = E_{p(x,y)} \frac{1}{\log p(X \mid Y)} \]

Suppose \( X \) is either a continuous or discrete RV. The relative entropy (or Kullback Leibler distance) between two p.d.f. (or p.m.f. in the discrete case) \( p_1(x) \) and \( p_2(x) \) is defined as

\[ D(p_1 \parallel p_2) = E_{p_1} \log \frac{p_1(X)}{p_2(X)} \]

The relative entropy \( D(p_1 \parallel p_2) \) is often used as a measure of the distance between two distribution \(^5\). Relative entropy is always non-negative and is zero iff \( p_1 = p_2 \). This inequality also holds even if the \( p_1 \) and \( p_2 \) are conditional p.d.f. However, it is not a true distance measure (metric) between distributions because it is not symmetric and does not satisfy the triangle inequality. Nonetheless, it is often useful to think of relative entropy as a distance metric.

We define the conditional relative entropy \( D(p_1(y \mid x) \parallel p_2(y \mid x)) \) as the average of the relative entropies between the two conditional p.d.f.s in the continuous RV case. (or between two conditional p.m.f.s in the discrete RV case).

\[ D(p_1(y \mid x) \parallel p_2(y \mid x)) = E_{p_{1(y \mid x)}} \log \frac{p_1(Y \mid X)}{p_2(Y \mid X)} \]

\(^5\)There is an important meaning in relation to the coding theory. Let X is a discrete RV. The entropy \( H(p_1) \) (or \( H(X) \)) is used as a measure of the amount of information required on average to describe the RV. On the other hand, relative entropy \( D(p_1\parallel p_2) \) is used as a measure of inefficiency of assuming that the true distribution is \( q \) when the true distribution is \( p \). For example, if we knew the true distribution of the RV, then we could construct a code with average description length \( H(p_1) \). If, instead, we used the code for a distribution \( q \), we would need \( H(p_1)+D(p_1\parallel p_2) \) bits on the average to describe the RV. For more details, refer to [Cover and Thomas 91].
The mean conditional relative entropy $\overline{D}(p_1(y \mid x) \parallel p_2(y \mid x))$ is the average of the relative entropies between the conditional p.d.f.s $p_1(y \mid x)$ and $p_2(y \mid x)$, averaged over the p.d.f. $p(x)$. (or between the conditional p.m.f.s $p_1(y \mid x)$ and $p_2(y \mid x)$ averaged over the p.m.f. $p(x)$ in the discrete RV case).

$$\overline{D}(p_1(y \mid x) \parallel p_2(y \mid x)) = E_{p(x,y)} \log \frac{p_1(Y \mid X)}{p_2(Y \mid X)}$$

Baram’s Information measure is a true distance metric between two probability distributions. The Baram’s distance between two probability density functions $p_1(x)$ and $p_2(x)$ for a continuous RV (or between two mass functions $p_1(x)$ and $p_2(x)$ for a discrete RV) is defined as

$$B(p_1(x) \parallel p_2(x)) = E_{p_{\text{true}}(x)} \log \frac{p_1(X)}{p_2(X)}$$

where $p_{\text{true}}$ denotes the true probability density for the RV $X$. If $p_{\text{true}} = p_1$ then Baram’s distance is same as Kullback’s distance, where $p_{\text{true}}$ denote the true $p$.

The conditional Baram’s information measure as a true distance between two conditional probability density functions. Baram’s distance between two p.d.f. $p_1(y \mid x)$ and $p_2(y \mid x)$ (or between two p.m.f. $p_1(y \mid x)$ and $p_2(y \mid x)$ in for a discrete RV) is defined as

$$B(p_1(y \mid x) \parallel p_2(y \mid x)) = E_{p_{\text{true}}(y\mid x)} \log \frac{p_1(Y \mid x)}{p_2(Y \mid x)}$$

The mean conditional Baram’s information measure a true distance between two conditional probability density functions averaged over the true probability density function $p_{\text{true}}(x,y)$.

$$B(p_1(y \mid x) \parallel p_2(y \mid x)) = E_{p_{\text{true}}(x,y)} \log \frac{p_1(Y \mid X)}{p_2(Y \mid X)}$$

If $f$ is convex function and $X$ is a RV, then Jensen’s inequality is

$$Ef(X) \geq f(EX)$$

If $f$ is strictly convex, then equality implies that $X = EX$ with probability one. and so, $X$ is a constant almost surely.
A.3 Model Conversion

In this section, we show how the ARMAX model can be converted to a prediction model. Consider the ARMAX representation of a linear stochastic system given by:

\[ A(z^{-1})y_{k+1} = B(z^{-1})u_k + C(z^{-1})e_{k+1}, k \geq 0 \]

With initial conditions \( y_k = 0, u_k = 0, e_k = 0, k < 0 \). If it is assumed that the \( e_k \) are independent and have zero mean, and \( A(0) = C(0) = I \). Then the model can be expressed as a prediction model with the following steps.

\[ C^{-1}(z^{-1})A(z^{-1})y_{k+1} = C^{-1}(z^{-1})B(z^{-1})u_k + e_{k+1} \]

\[ \Rightarrow y_{k+1} = (I - C^{-1}(z^{-1})A(z^{-1})) y_{k+1} + C^{-1}(z^{-1})B(z^{-1})u_k + e_{k+1} \]

\[ \Rightarrow \hat{y}_{k+1/k} = E(y_{k+1} | y^{k}, u^{k}) \]

\[ = (I - C^{-1}(z^{-1})A(z^{-1})) y_{k+1} + C^{-1}(z^{-1})B(z^{-1})u_k \]

and we can compute \( \hat{y}_{k+1/k} \) from the recursive equation

\[ C(z^{-1})\hat{y}_{k+1/k} = (C(z^{-1}) - A(z^{-1}))y_{k+1} + B(z^{-1})u_k \]

with initial conditions \( \hat{y}_k = 0, y_k = 0, u_k = 0 \) for \( k < 0 \). The modification for the non-zero horizon and non-zero initial condition can be done without much modification.

A.4 Matrix Identities

**Lemma 6** Let \( G(t) \) be a continuously differentiable \( n \times n \) matrix-valued function of the scalar parameter. Denote

\[ \left[ \frac{\partial}{\partial t} G(t) \right]_{ij} = \frac{\partial}{\partial t} [G(t)]_{ij} \]

Suppose that \( G(t) \) is non-singular at \( t = \bar{t} \). Then

\[ \frac{\partial}{\partial t} G^{-1}(t) = -G^{-1}(t) \left( \frac{\partial}{\partial t} G(t) \right) G^{-1}(t) \]

at \( t = \bar{t} \).
Proof. Since \( G(t) \) is non-singular and \( G \) is continuous, \( G(t) \) is non-singular on some neighbourhood \( \mathcal{N} \) of \( \bar{t} \) and thus,

\[
G(t)G^{-1}(t) = I \tag{A.1}
\]

on \( \mathcal{N} \). By the implicit function theorem, the neighbourhood \( \mathcal{N} \) can be chosen so that the function \( G^{-1}(t) \) is continuously differentiable on \( \mathcal{N} \). By differentiating both sides of Eq. A.1, with respect to \( t \), we can obtain that

\[
\left( \frac{\partial}{\partial t} G^{-1}(t) \right) G(t) + G^{-1}(t) \left( \frac{\partial}{\partial t} G(t) \right) = 0
\]

on \( \mathcal{N} \). Thus

\[
\frac{\partial}{\partial t} G^{-1}(t) = -G^{-1}(t) \left( \frac{\partial}{\partial t} G(t) \right) G^{-1}(t) \tag{A.2}
\]

at \( t = \bar{t} \). \( \Box \)

Suppose \( m(S) \) is a scalar-valued function on a space of matrices \( S \). In what follows we shall interpret that

\[
\left( \frac{\partial}{\partial S} m(S) \right)_{ij} = \frac{\partial}{\partial s_{ij}} m(S) .
\]

Lemma 7 Let \( \bar{S} \) be a non-singular \( n \times n \) matrix. Then

\[
\frac{\partial}{\partial S} \log \det S = (S^{-1})^T
\]

on a neighbourhood of \( \bar{S} \) in the space of \( n \times n \) matrices.

Proof. Let \( \mathcal{N} \) be a neighbourhood of \( \bar{S} \) on which \( \det \bar{S} \neq 0 \). Fix a pair of indices \((i, j)\). By Cramer's rule,

\[
(det S) I = S \text{ Adj } S
\]

where Adj denotes the adjoint matrix of \( S \). Equating the \((i, i)\)th components of the matrices in this equation, we obtain

\[
\det S = \sum_k s_{ik} [\text{Adj } S]_{ki} .
\]
It follows that on $\mathcal{N}$,
\[
\left[ \frac{\partial}{\partial S} \log \det S \right]_{ij} = \frac{\partial}{\partial s_{ij}} \log \det S = (\det S)^{-1} \frac{\partial}{\partial s_{ij}} \det S = (\det S)^{-1} [\text{Adj } S]_{ji}.
\]
Thus
\[
\frac{\partial}{\partial S} \log \det S = \left( S^{-1} \right)^T
\]
\[
\Box
\]

**Lemma 8** Let $\mathbf{S}$ be a non-singular $n \times n$ matrix and let $a$ be an $n$-vector. Then
\[
\frac{\partial}{\partial S} a^T S^{-1} a = \left[ S^{-1} a a^T S^{-1} \right]^T
\]
on a neighbourhood of $\mathbf{S}$ in the space of $n \times n$ matrices.

**Proof.** Let $\mathcal{N}$ be a neighbourhood of $\mathbf{S}$ on which $\det \mathbf{S} \neq 0$. By Eq. A.2. the followings are obtained on $\mathcal{N}$.
\[
\left[ \frac{\partial}{\partial S} a^T S^{-1} a \right]_{ij} = \frac{\partial}{\partial s_{ij}} a^T S^{-1} a = -a^T S^{-1} (O(i,j)S^{-1}a)
\]
where $O(i,j)$ denotes the matrix with 1 in the $(i,j)$th entry and zeros elsewhere. Thus.
\[
\begin{align*}
&= -\text{trace}\left\{ O(i,j)S^{-1}a a^T S^{-1} \right\} \\
&= -\sum_{k,l} O(i,j)_{k,l} \left[ S^{-1} a a^T S^{-1} \right]_{l,k} \\
&= -\left[ O(i,j) \right]_{i,j} \left[ S^{-1} a a^T S^{-1} \right]_{j,i} \\
&= -\left[ S^{-1} a a^T S^{-1} \right]_{j,i}
\end{align*}
\]
Therefore
\[
\frac{\partial}{\partial S} a^T S^{-1} a = -\left[ S^{-1} a a^T S^{-1} \right]^T.
\]
Appendix B

Simulation for the Gaussian Basis Function Network

In this appendix, identifications for the prediction module and the p.d.f. module is demonstrated by using the GBNF. In simulation B1, a surface fitting capability is demonstrated using noisefree data. Surface fitting (or curve fitting) capability is the key property for the prediction module in the proposed Detection Network. In Simulation B2, a similar test is performed, but the noisy data is considered. In Simulation B3, a p.d.f. estimation capability is demonstrated by using the GBNF, and the MDL method is used for a model selection.

B.1 Simulation B1 (Surface fitting with noisefree data)

A simulation for the prediction module using the GBNF's curve fitting capability is considered. Although the prediction module requires dynamic maps, a static map was considered here to emphasize the fitting capability more clearly. Extension to dynamic systems can be performed by using delayed coordinates. The selected target function for the demonstration is $f : \mathbb{R}^2 \rightarrow \mathbb{R}^1$. More specifically,

$$z = \frac{\cos(1.2x^2 + 1.2y^2)}{1 + 2x^2 + 2y^2}$$
This function is called *Mexican Hat* and is shown in Figure B.1. Tests were performed for two cases: one case is that the training data is noisefree (this will be shown in this simulation) and the other one is that the training data is noise contaminated (this will be shown in the next simulation). Although our aim is to train the noisy data, these two tests are performed to compare how difficult the training is when the data is contaminated.

A total of 1000 samples was uniformly collected over the domain \([-2, 2]^2\) and is shown in Figure B.1. However, the fitting was performed over the shrinked domain \([-1, 1]^2\) preceded by an appropriate linear transformation. The following GBRF with 49 kernels was used. Each kernel is set with same covariance structure, and kernel centers were fixed uniformly over the shrinked domain.

\[
\sum_{l=1}^{81} a_l \exp \left\{ -\frac{1}{2} \| x - m_l \|^2_{\Sigma^{-1}} \right\}
\]

where \(\Sigma^{-1} = \text{diag}(v,v)\). The kernel bandwidth was chosen to be equal to the distance between each adjacent kernel center so that each kernel overlaps. Generally, large bandwidths make final results smooth, however if they are too large then bias error increases (Geman and Doursat 92). In other words, the kernel bandwidth governs the trade-off between the bias error and the smoothness. Since the kernel center’s positions and bandwidths were fixed, parameters to be estimated are \(\{a_1, \ldots, a_{81}\}\). Thus the problem becomes a linear coefficient tuning problem, and the Singular Value Decomposition (SVD) method was used to find the 81 parameters.

Figure B.2 shows the test result. More tests were performed by using different numbers of kernels and the results are shown in the Figure B.3. These results show that as the number of kernels increases the quality of the fitting result increases. This tendency will continue until the number of kernels is equal to the number of samples.
B.2 Simulation B2 (Surface fitting with Noise contaminated data)

In this simulation, the curve fitting capability from the noised data is illustrated. Generally, the quality of the result we can expect depends on three things: how big the noise is, how many samples are available, and how complex the target function is. The extent to which the fitting quality can be expected from a given circumstance is certainly a difficult open question. In this simulation, the following class of noise was injected to the samples.

\[ i.i.d. \ N(0, 0.1^2) \]

Figure B.4 shows the contaminated training data. Tests were performed with different numbers of kernels (i.e., 25, 36, 49, 64, 81), but the same class of GBFN was used as in the previous simulation. The Validation Test method was used for a model selection method. For this Validation Test, the last 500 samples were reserved, and the first 500 samples were used for the tuning procedure.

Figure B.5 shows the test results when 49 kernels were used and Figure B.6 shows the test results when different numbers of kernels were used. Most of results seem to be worse than that of the noisefree case. The Validation Test gave the following result.

<table>
<thead>
<tr>
<th>Number of Kernels</th>
<th>LMS error of the test data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>$5.643 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>36</td>
<td>$5.760 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>49</td>
<td>$5.621 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>64</td>
<td>$5.826 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>81</td>
<td>$5.836 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>

\[
\text{LMS error} = \frac{\sum_{\text{test data}}(\text{output of GBFN})^2}{2 \cdot \text{number of test data set}}
\]

This result shows that the result obtained by using 49 kernels is the best and the one obtained by using 25 kernels is the second best without much difference between the
two. On the other hand, the two figures B.5 and B.6 show that the result obtained by using 25 kernels is the best and the one obtained by using 49 kernels is the second best without much difference between the two. This test shows that the Validation Test method is acceptable in this class of simulation. In practical situations, since we do not know the error function, we may have to use some model selection methods to infer a good representation among several different representations. like this Validation Test method. Further classes of GBFNs (e.g., superpositions of independent circles or independent ellipses) were tested under a time limitation in the tuning process. Note that these parameterizations increase the number of parameters. if the same number of kernels is used. Parameters were tuned by minimizing the LMS criterion, and this was performed by the search methods like Steepest Descent method or the Adaptive method. The initial parameters were chosen by using the result of the linear approach (i.e., with fixed kernel center positions and fixed size of circles). This method saves time to reach the local minimum. Training was continued for 100 epochs. It took a longer time than that of the linear GBFN during the training. When the training data was noisefree, this process often ended up with better results. However when the training data was contaminated, the result was not acceptable.

This search method was applied to other classes of simulations where the target function was less complex. This enabled us to use a fewer number of kernels (at most 25). The test results were acceptable in the noise cases too. Based on our observation, when the number of kernels (so is the number of parameters) is too large, search methods may not guarantee acceptable results, since the dimension of the parameter space is too large. For example, a GBFN with 49 kernels with independent circles needs 196 parameters and that with independent ellipses needs 294 parameters to be specified. Gradient based searches over the large dimensional space also increase the expectation for the result to fall into the local minimum.

B.3 Simulation B3 (p.d.f. estimation)

A simational study for the p.d.f. estimation module by using a GBFN was performed. Random samples were collected from the following two-dimensional
target function.

\[ 0.2 N(m_1, \Sigma_1) + 0.2 N(m_2, \Sigma_2) + 0.6 N(m_3, \Sigma_3) \]

where

\[
\begin{align*}
m_1 &= \begin{pmatrix} -0.75 \\ -0.75 \end{pmatrix}, \\
\Sigma_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
m_2 &= \begin{pmatrix} -0.25 \\ -0.25 \end{pmatrix}, \\
\Sigma_2 &= \begin{pmatrix} (2/3)^2 & 0 \\ 0 & (2/3)^2 \end{pmatrix} \\
m_3 &= \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix}, \\
\Sigma_3 &= \begin{pmatrix} (5/9)^2 & 0 \\ 0 & (5/9)^2 \end{pmatrix}
\end{align*}
\]

This function is an example that represents a skewed p.d.f. and is shown in Figure B.7. A total of 1000 samples was collected and used for the training.

The following generalized GBNF (i.e. non-linear superpositions of ellipses) was used for the p.d.f. estimate.

\[
\sum_{j=1}^{M} a_j |\Sigma_j|^{-1/2} (2\pi)^{-1} \exp \left\{ -\frac{1}{2} \| x - m_j \|_{\Sigma_j^{-1}}^2 \right\} \tag{B.1}
\]

Hence the parameters to be estimated are \( \{a_1, m_1, \Sigma_1^{-1}, \ldots, a_M, m_M, \Sigma_M^{-1}\} \). Five different representations were made by changing the number of kernels (i.e. \( M = 1, 2, 3, 4, 5 \)). In the parameter tuning procedure, it is highly recommended to start with several different initial conditions. However if the estimation task with fewer trials (say one) is required, then the initialization by using clustering methods can be considered. To demonstrate this, the k-mean clustering algorithm was applied for the selection of the initial parameters. During this initialization stage, covariance matrices were constrained such that each covariance matrix was diagonal (hence covariance matrices represent independent circles), but this constraint was removed during the tuning process such that the covariance matrices represent independent ellipses. For the parameter tuning, the Maximum Likelihood estimation method incorporated with the Expectation Maximization algorithm was used. The first simulation was performed by using three kernels. Figure B.8 shows the initial parameters that resulted from the k-mean clustering, and Figure B.9 shows the final parameters
that resulted from the EM algorithm. In Figure B.8, solid lines show how each kernel center has evolved during the clustering process. The diameter of the each painted circle represents the weight parameter (i.e. \( a_j \) in the equation B.1) of the each kernel, and each ellipsis represents the covariance structure of the each kernel. Figure B.10 shows the result of the estimation. Error functions of the p.d.f. estimation when different numbers of kernels were used are shown in Figure B.11. The following Minimum Description Length Criteria show that the best estimate in this class of simulation is the one obtained by using only two kernels. This model selection result is consistent with the graphical result. Note that the result of the two kernels is far more superior to that of the single Gaussian kernel.

<table>
<thead>
<tr>
<th>Number of kernels</th>
<th>Number of parameters</th>
<th>Log likelihood sum</th>
<th>MDL penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>-2406</td>
<td>2416.0</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>-2306</td>
<td>2335.9</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>-2301</td>
<td>2360.8</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>-2300</td>
<td>2399.0</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>-2296</td>
<td>2443.6</td>
</tr>
</tbody>
</table>
Figure B.1: Target function and noise-free data.
Figure B.2: Result of the surface fitting when noise-free data was used. 81 kernels were used.
Figure B.3: Result of the surface fitting when different number of kernels were used.
Figure B.4: Noise contaminated training data.
Figure B.5: Results of the surface fitting when noise in contained in the dataset. 49 kernels were used.
Figure B.6: Result of the surface fitting when noise is contained in the dataset.
True p.d.f.

Figure B.7: Target p.d.f.
Figure B.8: Samples and the result of parameter initialization obtained by applying the k-mean clustering algorithm.
Figure B.9: Log likelihood evolution during the estimation procedure, and estimated parameters using the GBFN.
Figure B.10: Result of the p.d.f. estimation.
Figure B.11: Error functions in the p.d.f. when different number of kernels were used. Each estimation was performed using k-mean clustering for initial parameters and further tuning was done using ML method with the EM algorithm.
Bibliography


184


[58] Poggio, T. and Girosi, F. (1990) Networks for approximation and learning, Proce-
dings of the IEEE. Vol.78, pp.1481-1497.

[59] Poor, H. V. (1986) ‘Robustness in signal Detection’, in Communication and Net-
works, Ed. by Blake I. F. and Poor H. V., pp. 131-156, Springer-Verlag

York: Springer-Verlag.


Singapore.

[64] Rumelhart, D.E. and McClelland, J.L., eds. (1986) Parallel Distributed Process-
MIT Press.

New York, Chapman and Hall

vances in Neural Information Processing Systems 6 (Cowan, J. D., Tesauro, G.,


[68] Tantaratanas, Sawasd (1986) Sequential Detection of a positive signal. in Com-
munications and Networks ed. by Ian F. Blake and H. Vincent Poor

ror Bars and an Application to Spectroscopy. In Advances in Neural Information
Processing Systems 6 (Cowan, J. D., Tesauro, G., and Alspector. J. eds.). pp.208-
215. San Francisco., CA: Morgan Kaufmann.

[70] Traven H. (1991) ‘A Neural Network Approach to Statistical Pattern Classifi-
cation by Semiparametric Estimation of probability Density Functions’. IEEE Tr.
on Neural Networks, Vol 2, No. 3, pp.366-377


[72] Wald, A. and Wolfowitz, J. (1948) Optimum character of the sequential proba-

by weight-elimination with application to forecasting. In R.P. Lippmann. J.E.
Moody, and D.S. Touretzky, editors, Advances in Neural Information Processing


