PATTERN RECOGNITION APPROACH AND ARRAY PROCESSING FOR DISTRIBUTED SOURCE IDENTIFICATION IN WATER POLLUTION SYSTEMS

Yoshitaka Shibata

December 1984
CSD-840062

The dissertation of Yoshitaka Shibata is approved.

Jack W. Carlyle

Jack W. Carlyle

L. P. McNamee

Lawrence P. McNamee

Reiner Stenzel

Edward C. Deland

Walter J. Karplus, Committee Chair

University of California, Los Angeles

1985

To my mother Ineyo

and in memory of my brother Rokuro (1938-1980)

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$a(x,t,u,c)$	field parameter function
$a_{i,j}(x,t,u,c)$	field parameter function
$a_j(x,t,u,c)$	field parameter function
$A(x,t)$	transition matrix
$A_u(x_i)$	auto correlation
$(b_{i,j})$	matrix element
$b(x)$	velocity of flow
$b_i(x,t,u,c)$	field parameter function
$b(x,t,u,c)$	field parameter function
$(b_{i,j})$	matrix element
B	coefficient matrix
$B_u(x_i,f)$	power spectra function
$c(x,t)$	pollutant index
$(c_{i,j})$	matrix element
$C(x_{m_i})$	correlation function
$C_{uz}(x_i,f)$	cross spectra
d_i	discriminant function
D	coefficient matrix or space region
$D(x,y,t)$	despersion coefficient
$e_i(t_k)$	measurement noise
e	threshold value
$E(\alpha,i)$	cost function
$E[.]$	mean operation

$f(t)$	time function
$F(x,t)$	pollution sourec function
F_i	input function vector
$g(x)$	space function
$G(n)$	vector for nodal space function
G	matrix
H	transfer coefficient matrix
i	index
I	integer or identity matrix
j	index
$J(\alpha)$	cost function
k	index
$K(x,y,t)$	conductibity function
K_d	re-aeration coefficient
K_r	deoxygenation coefficient
$K(n)$	gain matrix
l	index
L	integer
m	index
M	integer
n	index or porocity
N	integer
O	zero matrix
p_i	transition probability
P_i	subclass

$P_{(x)}$	matrix element
$P(n+1)$	covariance matrix
q_i	matrix element
Q	volume flow rate
$Q(\alpha)$	cost function
$Q(n+1)$	covariance matrix
Q	matrix
$r_i(x_i,f)$	coherence function
$R(u,x,t)$	input function
S	specific storage
S_i	subclass
$S(n+1)$	transition matrix
t	time index
T	time region, or sampling period
$u(x,t)$	BOD index
$U(x_{m_i},f)$	output from FFT
$v(x,t)$	DO index
$v_s(x,t)$	DO saturation
$V(x_{m_i},f)$	output from FFT
w_i^j	weighting vector
W_i	measurement matrix
x	space index
X	space index
$y(t)$	time functon

$Y(n+1)$	output matrix
$z(x_x,t)$	measurement data
$Z(n+1)$	measurement vector
α	weighting parameter
α_i	feature vector
β_i	feature vector
γ_i	feature element
$\delta(x-x_i)$	delta function
Δt	time interval
Δx	space interval
ω	frequency band
ϕ_i	phase angle
ψ_i	randum noise componet
ρ_i	normalized correlation function
σ	standard deviation
τ	time index
$\xi(k)$	noise component
$\frac{\partial .}{\partial t}$	partial derivative
$\lvert . \rvert$	absolute value
A^T	transposition of matrix
A'	transition matrix of DO
$\bar{A}$	mean value of matrix A
$\hat{G}$	optimal estimate value of G
$\Phi(x)$	transition matrix

∂D	boundary region
$\sum_{i=1}^{n}$	summation
$\int_{0}^{\infty}$	integration
$\bigcup$	union
$[\,.\,]^{-1}$	inverse matrix
U_*	conjugate complex
∇J	gradiant vector
$\approx$	approximation

Acknowledgment

I would like to express my deep appreciation to Prof. W. J. Karplus, my advisor, for his guidance, discussions, generousity and patience. He offered me an assistant research position. Without his guidance and encouragement, this work could not have been completed.

My gratitude is extended to the doctoral committee members, Dr. Jack Carlyle, Dr. Lawrence McNamee, Dr. Reiner Stenzel and Dr. Edward Deland.

I am especially grateful to Dr. William Yeh of Systems Engineering in UCLA for his comments and encouragement. I would also like to thank Mr. Borgioli of CSPI Inc. for his suggestions and for the opportunity to use Mini-Map. I also appreciate the assistant administrator in Computer Science Department, Ms. June Myers who helped things go smoothly for me, and the Ph.D candidacy Shun Cheung who helped me for operations of the computer systems, LOCUS in UCLA.

This work was supported in part by the National Science Foundation under grant ECS 79-26058 and Teledyne Controls Inc. in conjuction with the state of California MICRO program since 1982.

Last but not least, I would like to thank my mother, Ineyo and my wife, Reiko for their constant encouragement and patience.

VITA

November 13, 1950 - Born, Akita, Japan

1974 BSEE Soshia University, Tokyo, Japan

1976 M.S. Sophia University, Tokyo, Japan

1976-1979 System Engineer, Ishikawajima-Harima Heavy Industry Co., Japan

1979-1980 Research Assosiate, Sophia University, Tokyo, Japan

1981-Present Post Graduate Research Engineer, University of California, Los Angeles.

PUBLICATIONS

[1] Shibata, Y. & M. Kato, " Performance Analysis of Digital Processing Techniques by Means of Interpolated Z-Transform," the Soc. of Ele. Com., Japan, CAS, Vol.79, No.277,1979, P57-P60.

[2] Shibata, Y. & M. Kato, " Performance Analisis of Interpolated Z-Transform," Information Processing Societty, Japan, Proc. 1979.

[3] Shibata, Y. & Robert, M., Deiters, " Program Package for Design and Analysis of Linear Dynamic Systems," Japan Simulation Society, Proc., 1975.

[4] Shibata, Y., " Non-Linear Systems Identification: Parameters and State Vectors," Master's Thesis in Sophia University, Japan, 1976.

ABSTRACT OF THE DISSERTATION

Pattern Recognition approach and Array Processing
for Distributed Source Identification
in Water Pollution Systems
by
Yoshitaka Shibata
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 1985
Professor Walter J. Karplus, Chair

The research described in this dissertation is directed to the development of a methodology for the identification of input functions in distributed parameter systems and more specifically of pollution sources in water pollution systems. Two major challenging problem areas, i.e., river pollution systems and aquifer pollution systems, are discussed as examples. Conventional identification methods such as the regularization method among others have several crucial drawbacks. In particular they need restricted assumptions on pollution sources and involve a large mount of computation time. In order to overcome these difficulties, a pattern recognition approach including feature extraction and signal processing is introduced. Coherence functions and the normalized correlation function are employed as feature vectors to extract the original pollution pattern from the measurement data with the presence of high-level noise. Conventional identification methods are then employed to specify the extracted pollution sources more precisely. The entire identification procedure is executed by a host/peripheral array processor to improve computational speed. In particular, performance evaluation of the partial differential equations, the calculation of the feature vectors, the calculation of the conventional identification method of the identification process, are performed using DEC VAX-11/750 and CSPI Mini-Map array processor. The Monte Carlo method is introduced and executed to solve the partial differential equations. In order to demonstrate the verification of the entire identification procedure, several numerical examples are analyzed with the aid of simulations. Evaluations of the identification procedure are made by varying the noise level of the measurement data.

CHAPTER 1

Introduction

1.1 Background

Environmental pollution has recently become one of the important problems menacing the survival of human beings, animals and plants in many parts of the world. Water pollution problems in rivers, lakes and aquifers are no exception as the dependency on water for municipal, industrial and agricultural needs increases rapidly in urban areas. A vast of body of research in the modeling and analysis of water pollution systems in many countries has been made possible by advancements in science and technology. In particular, mathematical modeling and computer simulation techniques have been improved significantly by the development of large scale and high-speed computers. As a result, parameter identification and state estimation problems for the development of the mathematical models of water pollution systems have been discussed in the literature more and more frequently.

However, there have been relatively few studies of the modeling of the locations and intensities of pollution sources in water resources systems. This problem is significant especially for environmental control agencies attempting to supervise and predict environmental conditions.

In general, there are major difficulties involved in the identification of distributed pollution sources in environmental systems.

(1) Distributed pollution sources to be identified generally vary in space and time.

(2) General nature of pollution sources are at best only approximately known, therefore, they must be inferred using physical insight.

(3) Since environmental pollution systems are generally modelled by partial differential equations (PDE's), a large amount of computations are involved. Therefore, the selection of computing hardware facilities as well as algorithms and software must be carefully considered.

(4) From the mathematical point of view, pollution source identification problems in environmental systems are equivalent to the determination of input functions in distributed parameter systems, and are generally non-well posed inverse problems.

(5) The system observations involve measurements of certain system variables at a relatively small number of measuring stations and relatively infrequently in time, i.e. the observational data are generally very sparse.

(6) The observations are subject to substantial measurement error.

S.Ikeda and et al. [IkMS74] [IkMS77] [IkMS78] proposed a possible approach which employs the conventional identification method, (i.e., the regularization method introduced by A.N. Tihonov [Tiho63]) in order to solve identification problems of pollution sources in river pollution systems. Although this approach gave satisfactory numerical results, several restrictive assumptions were involved as discussed in Chapter 3.

In order to overcome those limitations and difficulties, pattern recognition methods [Karp72] are introduced in this dissertation. There are already several studies [Simu75] [Simu76] [Apen79] in which pattern recognition techniques, including clustering and region-growing, have been found to be very useful in providing an adequate mathematical model of the aquifer quantity problems for which sufficient information is not available to implement conventional identification methods. In the classification of another application of pattern recognition methods, the identification of distributed pollution sources is discussed. Basically, the pattern recognition approach involves the following major steps:

1) Pattern generation

2) Feature selection and extraction

3) Classification

4) Performance evaluation

On the first stage of the pattern recognition, several classes of patterns as possible model to be identified are designed and generated. It is the function of the pattern recognition system to determine from the observations to which of these classes the pollution sources under study actually belongs. For example several pollution sources locations might be nominated or their time patterns are briefly predicted. These preliminary knowledges could direct to the pattern generation: On the second step, the features are selected for the ability to enhance the similarities the output generated by the input pollution sources of the same pattern class, while enhancing the differences between the output generated by the input pollution sources of the different pattern classes. These features must be inherent in the measurement noise. On the third step, Classification involves the making of the decision as to which of the subclassified pattern is the closest to the original pollution sources. Discriminant function is introduced to execute the decision making. Finally the performance evaluation of the selected pattern including parameter adjustment of it is executed using computer simulation and conventional identification methods.

The entire identification procedure is computer intensive and involves a large mount of computation. Conventional sequential computers are inefficient and expensive for the entire identification procedure where relatively simple operations, such as additions and multiplications, need to be performed very large step of data elements. Array processor systems, which are combined with host computer are very suitable for this king of problem because they can enhance the performance of the host computer through the extensive parallelism and/or pipelining. Therefore the entire identification process can be carried out with very high speed at a modest cost.

1.2. Objectives of the dissertation

The following aspects of pattern recognition approaches to water pollution source identification are treated :

(1) The study of the mathematical formulation models of surface water ground-water pollution systems, in which measurement stations provide observational data.

(2) The augmentation of the regularization method and several alternative conventional methods.

(3) The establishment of the limitations and difficulties of the conventional methods.

(4) The introduction of the pattern recognition approaches.

(5) The introduction of time and space separation approaches for pollution sources.

(6) The formulation of cross-correlation functions and/or their spectra (Fourier transforms) in pattern selection and feature extraction.

(7) The exploitation of the computational advantages of the Monte Carlo method.

(8) The implementation and evaluation of array processors in order to improve computational efficiency.

1.3. Contents of the Dissertation

Chapter.2 opens with the formulation of mathematical models of both river pollution systems and ground-water pollution systems. Next, their observation systems over actual environments are described. Then, the identification problems of pollution sources of each system from measurement data are formulated. Finally, the general structure of the pollution sources to be identified are discussed by introducing separable functions of time and space.

Chapter.3 describes several approaches to the identification of pollution sources by conventional identification methods including the regularization method, the cross-validation method, the least- square method, the Kalman filter and the dynamic programming. Identification procedures for each method are derived and their advantages and disadvantages are discussed. Finally, the substantial difficulties and limitations common to all the conventional identification methods are summarized.

Chapter.4 introduces pattern recognition as a methodology for system identification in order to complement the conventional identification methods rather than replace them. General description of pattern recognition approaches and applications for pollution sources identification are formulated at each step. Finally, general of criteria to evaluate the performance of pattern recognition approaches are discussed.

Chapter.5 constitutes an introduction to the numerical implementations of the Monte Carlo methods for the solution of partial differential equations at the feature generation stage in pattern recognition methods. Finite difference equations to formulate the Monte Carlo method are derived by discretizing the original partial differential equations. Then the computer algorithms are formulated to execute the entire process of the Monte Carlo method. Finally the computational errors contained in the algorithm are characterized.

Chapter.6 organizes the entire identification procedures and combines the pattern recognition approaches with possible conventional identification methods to determine the unknown pollution sources. First the pattern recognition approaches discussed in Chapter 4 are employed to characterize an intermediate structure of pollution sources. Then various function identification or parameter identification methods discussed in Chapter 3 are applied.

Chapter.7 is devoted to a discussion of the numerical implementation for the entire proposed identification process including the pattern recognition approach as well as the conventional identification methods using high-speed array processors. Particular attention is focussed upon implementations by peripheral array processors, which are connected to a general purpose computer and can perform high-speed computations at a modest cost. Several identification tasks are executed both using a conventional computer, the VAX-11/750 and a combination of the VAX-11/750 and a peripheral array processor, the Mini-Map. Performance evaluations of the host-plus-peripheral array processor used for source identification are presented.

In Chapter.8, several numerical examples are presented to evaluate the entire identification process formulated in the preceding chapters. The first several numerical examples involve distributed pollution sources in river pollution in which the pollution sources are identified from BOD and DO. The second set of examples deals with point pollution sources in both river pollution and aquifer pollution systems. The computational results are evaluated by studying the manner in which the proposed identification process is capable of identifying the original pollution sources in the presence of increasing levels of measurement noise.

Chapter 9 includes a discussion and an evaluation of the proposed identification approach, summarizes accomplishments from the research and suggests future studies.

Finally, the appendices contain several derivations of theoretical formulations and array processor programs.

CHAPTER 2.

Mathematical Models of Water Pollution Systems

2.1 Introduction

This chapter opens with the formulation of mathematical models of both river pollution systems and ground-water pollution systems. Next, their observation systems over actual environments are described. Then, the identification problems of pollution sources of each system from measurement data are formulated. Finally, the general structure of the pollution sources to be identified are discussed by introducing separable functions of time and space.

2.2 Mathematical Modeling and Problem Statement

2.2.1 Mathematical Modeling

Generally, environmental systems such as water pollution systems, air pollution systems and thermal pollution systems are distributed parameter systems governed by natural physical laws. Their dynamics are characterized by parabolic partial differential equations based on the balance equation as follows [ApKa82] [Karp80] [Simu75] ;

$$a(x,t,u,c)\frac{\partial u}{\partial t}$$

$$= \sum_{j=1}^{N}\frac{\partial}{\partial x_j}[\sum_{i=1}^{N}a_{ij}(x,t,u,c)\frac{\partial u}{\partial x_i} + a_j(x,t,u,c)u]$$

$$+ \sum_{j=1}^{N}b_j(x,t,u,c)\frac{\partial u}{\partial x_j} + b(x,t,u,c)u + F(x,t) \qquad (2.2\text{-}1)$$

with initial conditions

$$u(x,t_0) = u_0(x) \qquad x \epsilon D \qquad (2.2\text{-}2)$$

and boundary conditions

$$f_1(x,u,t,c)\frac{\partial u}{\partial n} + f_2(x,u,t,c)u = u_s(x,t) \quad x \epsilon \partial D \qquad (2.2\text{-}3)$$

where

N : dimensionality in Euclidian space

u : pollution concentration indices

x,t : belongs to the product domain D x $[t_0,T]$

F(x,t) : internal pollution sources inputs

u_s : boundary inputs

f_1,f_2 : functions

a,a_{ij},a_j,b_j,b,c : distributed field parameters

The left-hand side of equation (2.2-1) represents the changes in the amount of the pollutant index in time. The first term of the right-hand side indicates the diffusion part, the second expresses the convection or transport part, the third represents the chemical or biochemical reaction, and the last part characterizes internal pollution sources. The field parameters and pollution sources must be provided as explicit functions or as numerical values in order to solve the mathematical model (2.2-1) analytically or numerically.

2.2.2 Observation systems

The pollutants are observed by measuring facilities distributed over fields. Because of physical and economic reasons, actual observations are executed in a discrete time-sampling manner at finite number of measuring locations. The observed data are subjected to measurement noise and errors so that the observation data are formulated as follows :

$$u_m(x_i,t_j) = Hu(x_i,t_j) + n_i(t_j) \quad \text{for } i=1,2,\ldots,M \text{ and } j=1,2,\ldots,L. \tag{2.2-4}$$

Where

u_m : measurement data,

H : transfer coefficients,

n_i : measurement noise and error,

M : the number of measurement stations, and

L : the number of measurement data.

2.2.3 Problem statement

Since pollution originates from pollutant sources, it is important to determine pollution source locations and the time variant intensities in order to predict and control current and future pollution conditions. Unfortunately, the direct observation and determination of pollution sources very difficult. Therefore, pollution sources must be indirectly identified from measurement data and the mathematical model. Two steps of identification process are considered ;

(1) first the identification of field parameters from measurement data so as to include them in the mathematical model.

(2) the identification of input pollution sources from measurement data and the mathematical model developed in step 1.

So far, most of the identification researches have focused on the first problem. In particular, pattern recognition approaches have succeeded in attaining the satisfactory results concerned in the modeling of of aquifer dynamics [Simu75] [SiKa75] and in ground-water pollution systems [Apen79] [ApKa82]. In our research, identification problems of the field parameters, which are assumed to be known, are not discussed. Rather the identification of distributed pollution sources F(x,t) as input functions will be investigated. The identification problems of pollution sources can be formulated as follows.

Problem

Determine the unknown pollution sources F(x,t) from the observation data $u_m(x_i,t_j)$ in the (2.2-4) and the mathematical model (2.2-1), the initial condition (2.2-2) and the boundary condition (2.2-3).

2.3 River Pollution Systems

Mathematical models of water quality systems are usually expressed by parabolic partial differential equations (PDE's) based on the mass balance equation. In the surface water quality systems comprised of rivers flowing through industrial areas where each factory is a potential pollution source, the biochemical oxygen demand (BOD) and the dissolved oxygen (DO) concentration are employed as pollutant indices [Rina79] [Sted81] [Ikeda76] ;

$$\frac{\partial u}{\partial t} = \sum_{j=1}^{N} \frac{\partial}{\partial x_j}[\sum_{i=1}^{N} a_{ij} \frac{\partial u}{\partial x_i}] - \sum_{j=1}^{N} b_j \frac{\partial u}{\partial x_j} - ku + F(x,t) \tag{2.3-1}$$

$$\frac{\partial v}{\partial t} = \sum_{j=1}^{N} \frac{\partial}{\partial x_j}[\sum_{i=1}^{N} a_{ij} \frac{\partial v}{\partial x_i}] - \sum_{j=1}^{N} b_j \frac{\partial v}{\partial x_j} - k_r(v-v_s) - k_d u \tag{2.3-2}$$

where

N : dimensionality

u(x,t) : biochemical oxygen demand (BOD) (ML^{-3})

v(x,t) : dissolved oxygen (DO) (ML^{-3})

v_s : dissolved oxygen saturation (ML^{-3})

k : BOD decay coefficient (T^{-1})

k_r : deoxygenation coefficient (T^{-1})

k_d : re-aeration coefficient (T^{-1})

a_{ij} : dispersion coefficient (L^2T^{-1})

b_j : velocity of the flow (LT^{-1})

F(x,t) : pollution sources to be identified $(ML^{-3}T^{-1})$

with initial conditions

$$u(x,0) = u_0(x) \quad x\epsilon D \tag{2.3-3}$$

$$v(x,0) = v_0(x) \tag{2.3-4}$$

and boundary conditions

$$u(x_b,t) = u_b(x) \quad x_b\epsilon\partial D \tag{2.3-5}$$

$$v(x_b,t) = v_b(x) \tag{2.3-6}$$

In practical applications, it is possible to make reasonable simplifications and approximations of the general mathematical models (2.3-1) and (2.3-2) without losing any substantial characteristics of the actual river being observed. In most cases, for example, the width and/or the depth of the river are relatively small compared with the length along the river, consequently the dimensionality can be reduced from three to one or two. In other cases, based on physical and geometrical situations, some part of the field parameters can be regarded as time-invariant, homogeneous or constant so that the complexities of the mathematical model can be reduced. The most frequently employed models to date are summarized in Table.2.1. In our research, the diffusion model is examined.

Table.2.1 TYPICAL RIVER POLLUTION MODELS

Model Type	Model Name	Characteristics	—
Tank Type	Young & Beck	Continuously stirred	[YoBe74]
	$\frac{du_i}{dt}=-B_{i-1}u_{i-1}-B_iu_i+R(u_i,t)$		
Transport Type	Streeter & Phelps	Steady Flow	[StPh25]
	$\frac{\partial u}{\partial t}=-b\frac{\partial u}{\partial x}+R(u,x)$		
	O'Connor	Flow changes with space	[O'con67]
	$\frac{\partial u}{\partial t}=-b(x)\frac{\partial u}{\partial x}+R(u,x)$		
	O'Connor & Ditoro	Flow changes with time	[O'cDi70]
	$\frac{\partial u}{\partial t}=-b(x,t)\frac{\partial u}{\partial x}+R(u,x,t)$		
Diffusion Type	Dobbins	Diffusion	[Dobb64]
	$\frac{\partial u}{\partial t}=a\frac{\partial^2 u}{\partial x^2}-b\frac{\partial u}{\partial x}+R(u,x,t)$		

Monitoring stations are assumed to be located at a finite number of locations x_i , for i=1,...,M. At these pollution index BOD̄ and/or DO in the presence of noise and measurement error can be directly observed at selected times t_j. Therefore, the measured data can be expressed as :

$$u_m(x_i,t_j) = u(x_i,t_j) + e_i(t_j) \tag{2.3-7}$$

$$v_m(x_i,t_j) = v(x_i,t_j) + w_i(t_j) \tag{2.3-8}$$

for i=1,2, ... ,M and j=1,2, ...,L where $e_i(t_j)$ and $w_i(t_j)$ are measurement noise and errors. Thus, the identification problems of pollution sources can be formulated as follows:

Problem

Determine the input function F(x,t) in (2.3-1) from the measurement data $u_m(x_i,t_j)$ in (2.3-7) or $v_m(x_i,t_j)$ in (2.3-8) and the initial conditions (2.3-3) and (2.3-4) as well as the boundary conditions (2.3-5) and (2.3-6).

2.4 Ground-Water Pollution Systems

The general mathematical model of ground-water pollution systems would involve a complete physical-chemical description including the movement of the ground-water as well as chemical reactions among the materials [NeFa80] [Ande79]. From the hydrological point of view, however, many cases of ground-water pollution problems usually deal with a simplified subclass of the general problem by assuming that no chemical reactions occur [Khan79] [Khan80]

[Pere72]. In order to characterize the dynamic phenomena of ground-water pollution systems, the ground-water flow equation which describes water movement in an aquifer and the dispersion equation which describes the movement of dissolved chemical materials in ground-water are solved simultaneously [NeFa80] [Ande79] [MeFa80].

The mathematical model for simulating ground-water flow is based on a water balance equation as

$$\sum_{j=1}^{N} \frac{\partial}{\partial x_j} [\sum_{i=1}^{N} K_{ij} \frac{\partial h}{\partial x_i}] = S\frac{\partial h}{\partial t} + Q \qquad (2.4\text{-}1)$$

with initial conditions

$$h(x, t_{0)} = h_0(x) \qquad x \epsilon D \qquad (2.4\text{-}2)$$

and boundary conditions

$$h(x_b, t) = h_b(t) \qquad x \epsilon \delta D \qquad (2.4\text{-}3)$$

where

h : hydraulic head (L)

Q : volume flow rate (T^{-1})

K : hydraulic conductivity tensor (LT^{-1})

S : specific storage (L^{-1})

The dispersion equation is based on a mass balance equation as

$$\frac{\partial c}{\partial t} = \sum_{j=1}^{N} \frac{\partial}{\partial x_j} [\sum_{i=1}^{N} D_{ij} \frac{\partial c}{\partial x_i}] - \sum_{i=1}^{N} \frac{\partial}{\partial x_i}(v_i c) - \frac{cQ}{n} + F \qquad (2.4\text{-}4)$$

with initial conditions

$$c(x,0) = c_0(x) \quad x \epsilon D \tag{2.4-5}$$

and boundary conditions

$$c(x_b,t) = c_b(t) \quad x_b \epsilon \delta D \tag{2.4-6}$$

where the velocities are calculated using the Darcy equation as

$$v_i = \sum_{j=1}^{N} \frac{K_{ij}}{n} \frac{\partial h}{\partial x_j} \tag{2.4-7}$$

where

C : material concentration (ML^{-3})

D : coefficient of dispersion (L^2T^{-1})

F : concentration of the source or sink fluid $(ML^{-3}T^{-1})$

n : effective porosity (dimensionless)

v_i: seepage velocity or pore velocity (LT^{-1})

In many cases, two dimensional mathematical models which deal with dynamic phenomena in the horizontal plane are widely employed because the movement in the vertical direction can generally be neglected [KoBr74] [HeLa77] [BGST82]. Two dimensional models, furthermore, are classified into two types, areal model and profile model. Areal model deals with horizontal slice of the surface. This model is useful when it is necessary to consider the effects of distributed sources and are well suited to problems which use of a regional rather than a local scale. Many applications for actual environments have employed areal model [BrPi73] [Koni77] [KoBr74]. Profile model, on the other hand, simulates flow through a vertically oriented slice of the surface. In sluding the movement of pollutants from a point source where it is necessary to determine the vertical extent of the pollutants plume, it is often advisable to use a profile model [Robi78] [Pick77] [PiCh78].

Observations of material concentrations are made directly from observation wells at monitoring stations at a finite number of locations where the pollution concentrations can be observed at selected time in the presence of noise and error. Therefore, measured data can be expressed as follow ;

$$c_m(x_i,y_i,t_j) = c(x_i,y_i,t_j) + e_i(t_j) \tag{2.4-8}$$

for i=1,...,M and j=1,2...,L where $e_i(t_j)$ are measurement noise and error.

Problem

Determine the pollution sources F(x,y,t) from measured data $c_m(x_i,y_i,t_j)$, mathematical models (2.4-1), (2.4-4) and (2.4-7) and initial conditions (2.4-10) and (2.4-5) as well as boundary conditions (2.4-3) and (2.4-6).

2.5 Mathematical Expressions of Pollution Sources

Generally, pollution sources in actual environmental systems vary in time and in location. Therefore, they must be expressed mathematically as a function of time and space. The most common and suitable expression is to employ a separable form of the input source as follows :

for a single source,

$$F(x,t) = y(t)g(x) \tag{2.5-1}$$

for multiple sources,

$$F(x,t) = \sum_{i=1}^{L} y_i(t) g_i(x) \tag{2.5-2}$$

This assumption of separability has a physical justification because $g_i(x)$ can be regarded as the spatial profile reflecting the location and the nominal intensity of a pollution source, while $y_i(t)$ exhibits the time variation of pollutant discharge intensity from sources. On occasion pollution patterns on different fields can be classified based on a priori knowledge of their time-pattern or their locations. In a typical situation of river pollution systems, for example, factories, located at x_i along the river through the industrial area, are regarded as point pollution source $f_i(t)$ with cycles of daily, weekly, monthly, seasonally or yearly periods. Entire pollution sources, therefore, can be expressed as

$$F(x,t) = \sum_{i=1}^{K} \{a_i \delta(x - x_i)\} y_i(t) \tag{2.5-3}$$

or

$$= \sum_{i=1}^{K} \delta(x - x_i) f_i(t) \tag{2.5-4}$$

In an agricultural situation [Shen79], for example, spatial pollution caused by chemicals distributed over agricultural fields penetrate into the river. The entire pollution field can be expressed as

$$F(x,t) = g(x) y(t) \tag{2.5-5}$$

Some pollution sources are regarded as being constant in time, others might be considered as a time variant distributed function. In these cases, the mathematical expession of the pollution sources can be simplified. In our research, the identification problems of point and distributed pollution source are separetely formulated in the following chapter.

CHAPTER 3

Conventional Identification Method

3.1 Introduction

This chapter describes several approaches to the identification of pollution sources by conventional identification methods including the regularization method, the cross-validation method, the least- square method, the Kalman filter and the dynamic programming. Identification procedures for each method are derived and their advantages and disadvantages are discussed. Finally, the substantial difficulties and limitations common to all the conventional identification methods are summarized.

3.2 General Parameter Optimization Methods

The approaches most often used in systems identification, involve parameter optimization methods [Eykh63], [Eykh73] and [Beke70], and require an explicit understanding of the mathematical structures of distributed pollution sources as input function F(x,t) of space and time. The parameters of the assumed pollution source function F(x,t) are then modified so as to minimize the difference between the response of the mathematical model based using on assumed F(x,t) and the response observed from the actual physical system. These methods work only if the starting model structure is reasonably acculate

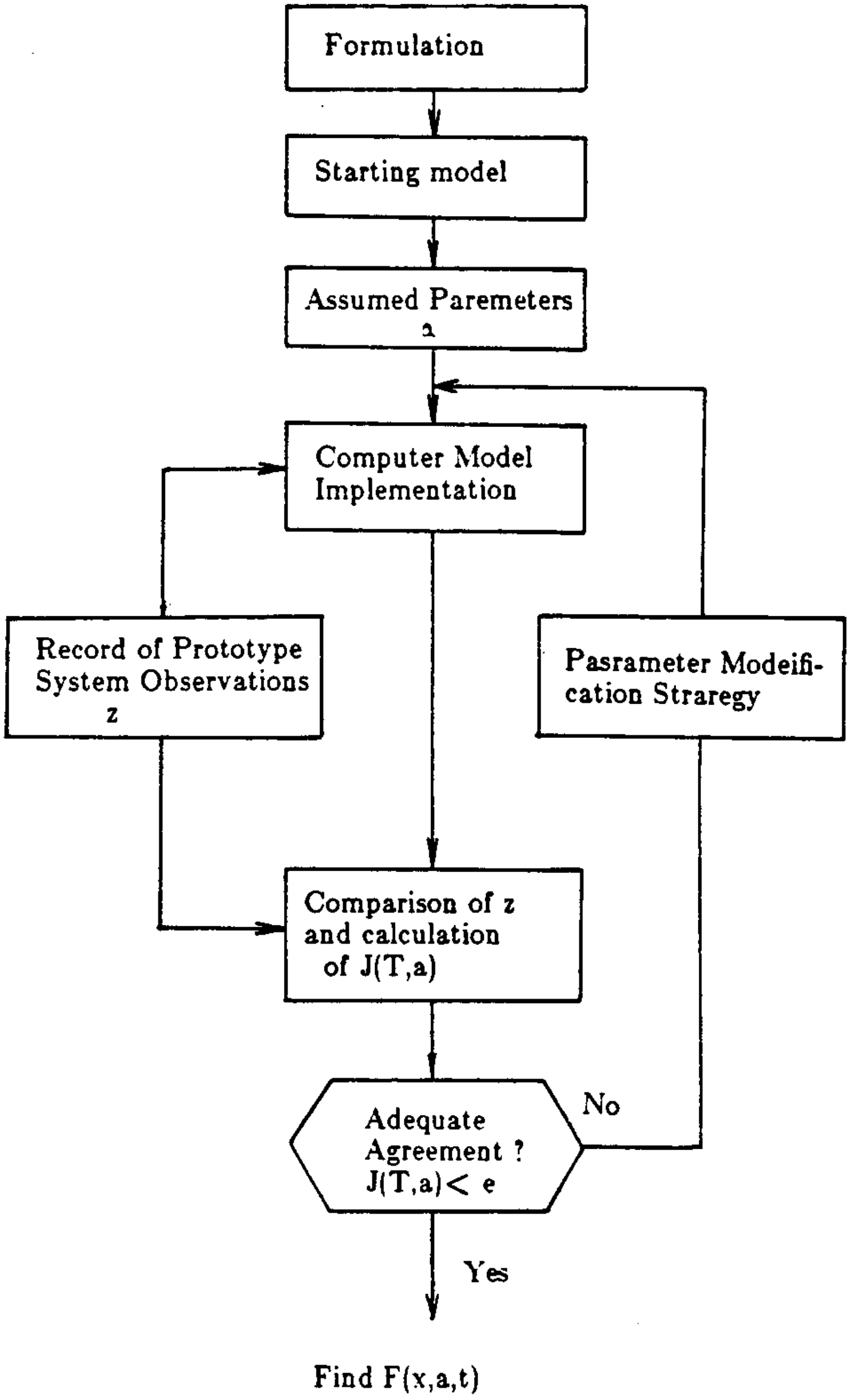


Fig 3.1 Conventional Identification Method

description of the actual pollution sources as presented in Fig.3.1. In attempting to apply these methods, the following difficulties are encountered.

(1) There is no evaluation of validity of the structure of pollution sources weather or not it is sufficiently-close to the original physical pollution sources.

(2) Observations from physical system are corrupted by noise.

For these reason, the attaining of a minimum in a criterion function cannot be taken with confidence as an indication of the validity of pollution sources.

3.3. Regularization Method

S.Ikeda at al. [IkMS74], [IkMS77], [IkMS78] attempted the identification of pollution sources in river pollution systems using the method of regularization which was originally introduced by A.N. Tihonov [Tiho63], [Tiho65]. This method basically reduced the identification problem to the solution of the Fredholm equation as follows: Suppose that there is a river pollution system whose pollution index BOD is expressed by the one-dimensional parabolic partial differential equation as :

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b\frac{\partial u}{\partial x} - ku + F(x,t) \tag{3.3-1}$$

with initial condition

$$u(0,t) = 0 \tag{3.3-2}$$

and boundary conditions

$$u(\pm\infty,t) = 0 \tag{3.3-3}$$

and observation is taken at **x**=c

$$z(t) = u(c,t) \qquad t \in (0,T)\ (c \in R : \mathit{fixed}) \tag{3.3-4}$$

where all field parameters are constant. The pollution source as an input function F(**x**,t) is assumed to be expressed as separable function in time and space as ;

$$F(x,t) = g(x)f(t). \tag{3.3-5}$$

Furthermore, the time function f(t) is assumed to be known and g(**x**) is unknown. Therefore, the problem to identify the F(**x**,t) can be reduced to the identification of g(**x**) from the observation z(t) by solving the equation of the first kind :

$$z(t) = \int_{0}^{t}\int_{-\infty}^{+\infty} U(t-\tau,c,x)f(\tau)g(x)dxd\tau \tag{3.3-6}$$

where U(t,x,y) is the fundamental solution of the (3-1) given by

$$U(t,x,y) = \frac{1}{2\sqrt{a\pi t}}\exp\left\{-\frac{(x-y)^2}{4at} + \frac{b}{2a}x - \left(\frac{b^2}{4a} + k\right)t\right\} \tag{3.3-7}$$

Eqn. (3.3-6) is equivalent to the Fredholm equation

$$z(t) = \int_{-\infty}^{+\infty} K(x,t)g(x)dx \tag{3.3-8}$$

where the kernel equation is expressed by

$$K(x,t) = \int_{0}^{t} U(t-\tau,c,x)f(\tau)d\tau. \tag{3.3-9}$$

The optimization problem is to find that g(**x**) which minimize the functional J_1

$$J_1 = \int_0^T \{ z(t) - \int_{-\infty}^{+\infty} U(t-\tau,c,x)f(\tau)g(x)d\tau \}^2 dt. \tag{3.3-10}$$

Since this problem is not well-posed, the optimization process might well diverge computationally. Thus the regularization method was introduced to stabilize the problem by introducing an extra term so that

$$J_\alpha = \int_0^T \{ z(t) - \int_{-\infty}^{+\infty} K(x,t)g(x)dx \}^2 dt + \alpha \int_{-\infty}^{+\infty} g^2(x)dx \tag{3.3-11}$$

By solving the Euler equation associated with (3.3-11), the optimal g(x) for J_α can easily obtained. That is, for the optimal g_α, the Euler equation can be satisfied ;

$$\int_{-\infty}^{+\infty}\int_0^T K(x,t)K(t,s)dt g(s)ds - \int_0^T K(x,t)z(t)dt + \alpha g(x)^2 = 0 \tag{3.3-12}$$

By finite difference approximation of (3.3-12), we can obtain

$$(K^T K + \alpha I)g_\alpha = K^T z \tag{3.3-13}$$

Where

$$K = (K_{ij}) = (K(i\Delta t, j\Delta x)) \tag{3.3-14}$$

$$g = (g_j) = (g(i\Delta x)) \tag{3.3-15}$$

$$z = (z_i) = (z(i\Delta x)) \tag{3.3-16}$$

Eqn. (3.3-13) can be solved easily by any of several numerical techniques such as the Gauss elimination method [Wait79], [Rice73], Gauss-Seidel iteration method or Jacobi iteration method [Rice73]. Although this method gave satisfactory numerical results, the following restrictive assumptions were involved :

(1) It was assumed that the input function F(x,t) to be identified can be separated into g(x) and f(t). Furthermore either g(x) or f(t) was assumed to be known in advance. Therefore, the identification problem was to determine either the function g(x) or f(t). However, generally both g(x) and f(t) must be identified.

(2) A least square errors term was employed as an evaluation function,

$$J_\alpha = \int_0^T \{ z(t) - \int_{-\infty}^{+\infty} K(x,t)g(x)dx \}^2 dt + \alpha \int_{-\infty}^{+\infty} g^2(x)dx$$

and the optimal g(x) that minimizes the function J_α was determinedd. However the optimal value g(x) depends strongly on the parameter α, and unfortunately there is no general procedure to determine the most suitable value of α.

(3) The numerical applications were only implemented for a simple model, namely, a one-dimensional linear, homogeneous field and a single pollution source.

(4) It proved quite difficult to find the fundamental solution U(x,y,t) for complicated mathematical models, higher dimension and complicated boundary.

(5) The computation time for the solution of the algebraic equation (3.3-13) became very large with increasing time and space discriterization.

(6) Measurement error and noise were not considered.

3.4 Formulation of Functional Optimization Problems

In order to overcome the difficulties and the limitations inherent in the regularization method, the possibilities of several alternative conventional approaches, discussed in the following sections, are considered below. Consider the reformulation of the optimization problem by using the finite difference approximation of the function of space :

$$u_{k+1}(x) \approx A(x)u_k(x+\Delta x) + B(x)u_k(x) + C(x)u_k(x-\Delta x) + g(x)f_k(x)\Delta t \tag{3.4-1}$$

For discrete space points $x_k = k\Delta x$, Eqn (3.4-1) can be written as a vector expression

$$\begin{bmatrix} u_{k+1}(0) \\ u_{k+1}(\Delta x) \\ u_{k+1}(2\Delta x) \\ . \\ . \\ u_{k+1}(N\Delta x) \end{bmatrix}$$

$$= \begin{bmatrix} B_0 A_0 & & & & \\ C_1 B_1 A_1 & & & & \\ & C_2 B_2 A_2 & & & \\ & & . \; . \; . & & \\ & & & . \; . \; . & \\ & & & & C_N B_N \end{bmatrix} \begin{bmatrix} u_k(0) \\ u_k(\Delta x) \\ u_k(2\Delta x) \\ . \\ . \\ u_k(N\Delta x) \end{bmatrix} + \begin{bmatrix} g(0) \\ g(\Delta x) \\ g(2\Delta x) \\ . \\ . \\ g(N\Delta x) \end{bmatrix} f_k\Delta t \tag{3.4-2}$$

or simply,

$$U_{k+1} = AU_k + Gf_k \tag{3.4-3}$$

where the state vector U_k is defined :

$$U_k = [u_k(0), u_k(\Delta x), .., u_k(N\Delta x)]^T \tag{3.4-4}$$

and the gain vector G to be identified is defined as :

$$G = [\ g(0),\ g(\Delta x),\ g(2\Delta x),\ \cdots\ ,\ g(N\Delta x)\ ]^T \tag{3.4-5}$$

and the known scalar time function f_k as

$$f_k = f_k \Delta t \tag{3.4-6}$$

and the known coefficient matrix A as :

$$A = \begin{bmatrix} B_0 & A_0 & & & & \\ C_1 & B_1 & A_1 & & & \\ & C_2 & B_2 & A_2 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & C_N & B_N \end{bmatrix} \tag{3.4-7}$$

By applying the same discreterization procedure, the measurement equation can be also expressed by a vector expression

$$Z_k = CU_k + n_k \tag{3.4-8}$$

where the measurement vector, Z_k is :

$$Z_k = [z_k(x_{m1}), z_k(x_{m2}), .., z_k(x_{ml})]^T \tag{3.4-9}$$

The terms $x_{m_1}, x_{m_2}, .., x_{m_l}$ indicate the measurement locations, C is the coefficient matrix and n_k is the observation noise and error vector. Next, we define the discrete type of the error functional J instead of (3.3-11) such that

$$J = \sum_{k=1}^{n} (Z_k - CU_k)^T (Z_k - CU_k) + \alpha G^T G \tag{3.4-10}$$

Therefore, the identification problem can be reformulated as follow :

Problem Statement

Given the system equation with known initial condition U_0

$$U_{k+1} = AU_k + Gf_k \tag{3.4-11}$$

the observation equation

$$Y_k = CU_k \tag{3.4-12}$$

and measurement data from the actual environment

$$Z_k = Y_k + n_k, \tag{3.4-13}$$

Find the optimal estimate value $\hat{G}$ to minimize the error functional J

$$J = \sum_{k=1}^{n} (Z_k - CU_k)^T (Z_k - CU_k) + \alpha G^T G \tag{3.4-14}$$

3.5 Cross-Validation Methods

In order to determine the best α without any a priori knowledge of the generating mechanism for the measurement data, the cross-validation method [Ston74] [Geis75] [Wahb75] is introduced in this section. At first, any non-negative value of α is selected. Next the $U(i,\alpha)$ and the $G(i,\alpha)$ which satisfies the system Eqn. (3.4-11) and which minimizes the error function $E(i,\alpha)$

$$E(i,\alpha) = \frac{1}{n-1} \sum_{k=1,k\neq i}^{n} [(Z_k - CU_k)]^T (Z_k - CU_k)] \tag{3.5-1}$$

is calculated for i=1,...n. That is, $U(i,\alpha)$ is the solution to the system Eqn. (3.4-11) with point U_i and Z_i removed from the data set. Then let $Q(\alpha)$ be defined as

$$Q(\alpha) = \frac{1}{n}\sum_{i=1}^{n}[(Z_i - U(i,\alpha)^T(Z_i - U(i,\alpha)] \tag{3.5-2}$$

Finally, the optimal α , namely α_{opt}, that minimize the $Q(\alpha)$ is be determined. The curve $U(k,\alpha_{opt})$ and $G(i,\alpha_{opt})$ are the optimal solutions for U and G. In order to clearly understand this method, an identification algorithm is described in more detail.

Step 1 Read in parametric data, such as A, C, U_0 , N, n, and measurement data Z(i) for i=1,...,n.

Step 2 Specify initial positive starting value α.

Step 3 Set index i to 1 where i is the number of computations when the α is fixed at some value.

Step 4 Find the $G(i)$ which minimizes Eqn.(3.5-1)

Step 5 Calculate and accumulate $Q(\alpha)$.

Step 6 Check whether the counter is i= n. If $i \neq n$, add 1 to i and go back to Step 3. If i=n, then go to the next step.

Stop 7 Check whether $Q(\alpha)$ is at the minimum point with α .

Stop 8 By using the α_{opt}, find the optimal $G(\alpha_{opt})$ and print all calculated values.

END

This cross-validation method possesses the following advantages over the regularization method :

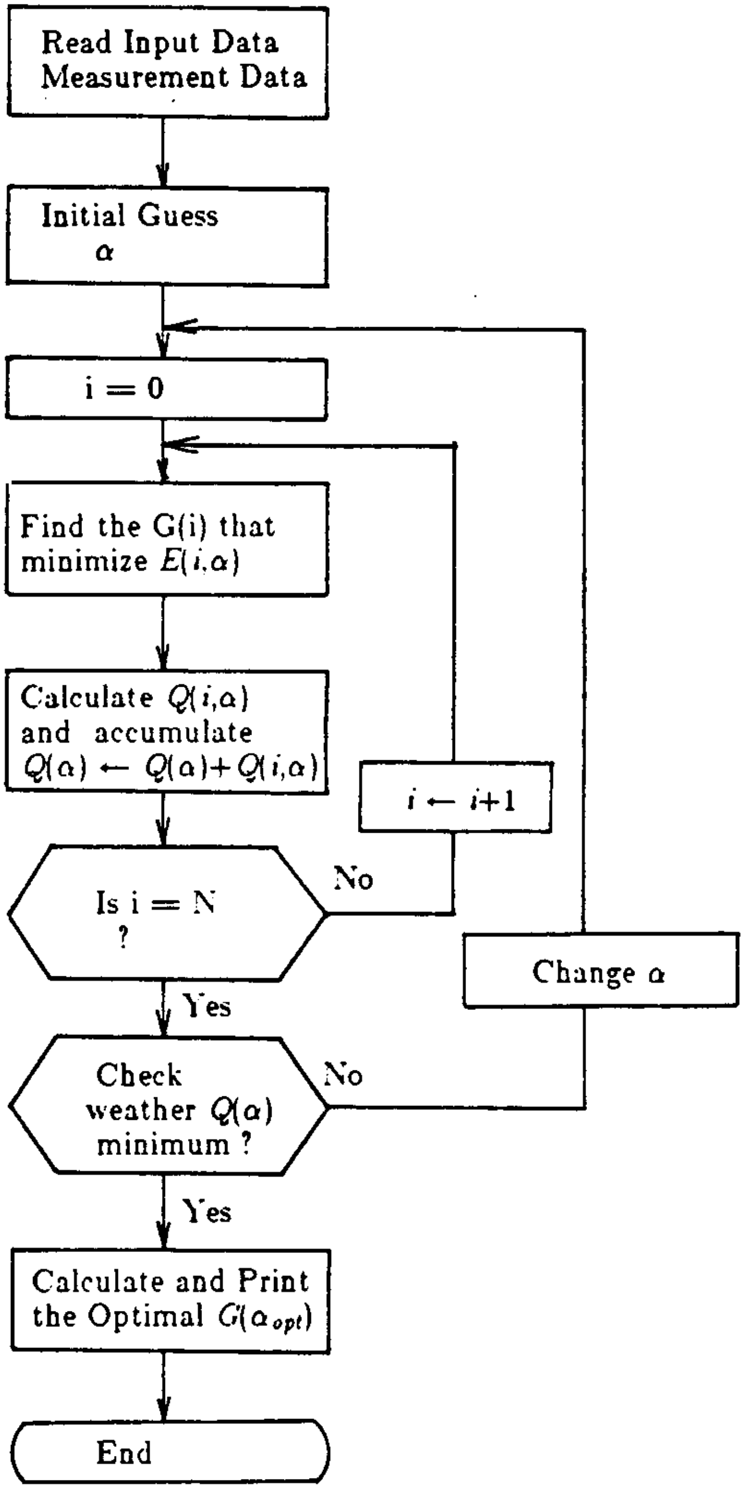


Fig.3. 2 Identification by The Cross-Validation Method

(1) The difficulties in the regularization method to obtain the optimal α can be overcome by using the cross-validation.

(2) The noise and error terms of the observed data can be taken into consideration by modefying α and by evaluating the sensitivity error function $E(i,\alpha)$.

(3) The general case of identification problem where the field parameters are a function of time and space can be formulated by finite difference discreterization.

The cross-validation method has the following disadvantages :

(1) The time function $f(t)$ must be known.

(2) Since $G(i,\alpha)$ must be recalculated for different α, the computation time increases as the number of observed data and the number of space points increases even though the entire algorithm includes vector and matrix calculations which are quite simple in structure.

3.6 Least Square Method

It was pointed out in the previous section that identification problems (3.3-7) through (3.3-10) are usually inverse problem and not well-posed. Therefore sequential calculation methods which do not involve the matrix inversion are desired. To this end, a sequential type of the least square method is introduced in this section.

Initially the system equation (3.4-1) is transformed to an equation which does not include the state vector U_k as follows :

For k=1,

$$Y_1 = CU_1 = C(AU_0+Gf_0). \tag{3.6-1}$$

For k=2,

$$Y_2 = CU_2 = C(AU_1+Gf_1) = CA^2U_0+CAGf_0+CGf_1. \tag{3.6-2}$$

$$\vdots$$

For k=k

$$Y_k = CU_k = C(AU_{k-1}+Gf_{k-1}) \tag{3.6-3}$$

$$=CA^kU_0+CA^{k-1}Gf_0+CA^{k-2}Gf_1+\cdots+CGf_{k-1}. \tag{3.6-4}$$

For convenience, the initial condition is contained in Y_k so that

$$Y_k \equiv Y_k - CA^kU_0. \tag{3.6-5}$$

Therefore,

$$Y_k = C(A^{k-1}f_0 + A^{k-2}f_1 + A^{k-3}f_2 + \cdots + If_{k-1})G \tag{3.6-6}$$

$$\equiv X_kG$$

where

$$X_k = C(A^{k-1}f_0 + A^{k-2}f_1 + \cdots + If_{k-1}) \tag{3.6-7}$$

Using Eqn.(3.6-7), the error function J can be written as

$$J = \sum_{k=1}^{n} (Z_k - X_k G)^T (\dot{Z}_k - X_k G). \tag{3.6-8}$$

In order to find the optimal value $\hat{G}(n)$, that minimizes the error function J, the following condition must be satisfied :

$$\frac{\partial J}{\partial G} = 0 \quad at\ G = \hat{G}(n). \tag{3.6-9}$$

From this condition, we obtain

$$\frac{\partial J}{\partial G} = \sum_{k=1}^{n} \{-2X_k^T Z_k + 2X_k^T X_k \hat{G}(n)\} = 0. \tag{3.6-10}$$

That is,

$$(\sum_{k=1}^{n} X_k^T X_k)\hat{G}(n) = \sum_{k=1}^{n} (X_k^T Z_k). \tag{3.6-11}$$

Moreover, we define

$$P(n) = [\sum_{k=1}^{n} X_k^T X_k]^{-1}. \tag{3.6-12}$$

By using these equations, the following sequential algorithm of the least square method to find the optimal estimate value of $\hat{G}(n)$ can be finally derived (see Appendix A).

$$\hat{G}(n+1) = \hat{G}(n) + K(n)[Z(n+1) - X(n+1)\hat{G}(n)] \tag{3.6-13}$$

$$P(n+1) = P(n) - K(n)X(n+1)P(n) \tag{3.6-14}$$

$$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1}. \tag{3.6-15}$$

The many advantages of the least square method include :

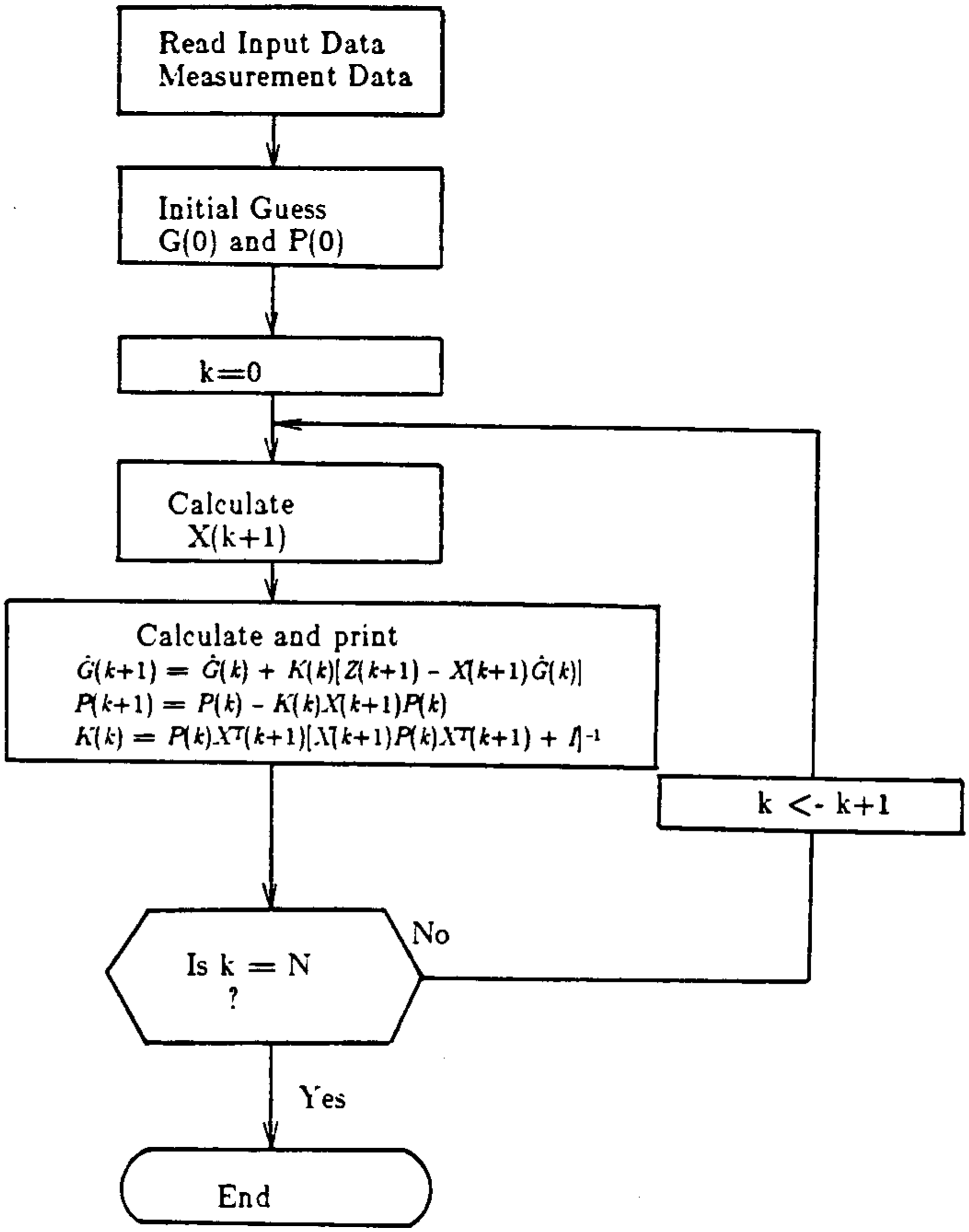

Read Input Data
Measurement Data
Initial Guess
G(0) and P(0)
k=0
Calculate
X(k+1)
Calculate and print
$\hat{G}(k+1) = \hat{G}(k) + K(k)[Z(k+1) - X(k+1)\hat{G}(k)]$
$P(k+1) = P(k) - K(k)X(k+1)P(k)$
$K(k) = P(k)X^T(k+1)[X(k+1)P(k)X^T(k+1) + I]^{-1}$
k <- k+1
Is k = N
?
No
Yes
End

Fig.3.3 Identification by The Least Square Method

(1) This method has very simple sequential calculation involving the simultaneous solutions of Eqn.(3.6-13) through (3.6-15). Only the initial values of G(0) and P(0), which are selected for the identity matrix as starting values must be specified.

(2) There is no need to determine α. Therefore the computation time is reduced.

(3) The computer algorithm is vector and matrix oriented so modern computer architecture such as array processors can be used to advantage.

However, as in the method described previousely,

(1) This method also requires a priori knowledge of time function f(t).

3.7 Kalman Filtering Method

In the previous section, the sequential least square method was derived for the optimal $\hat{G}$. In this section, not only the optimal estimate value of $\hat{G}$ but also $\hat{U}_k$ are derived by Kalman Filtering [KoPh76] [BoGr78]. By this approach, modeling errors by finite difference approximation as well as the random the behavior of pollution sources are taken into consideration. For convenience, we regard the parameter value g_i as a new vector $u_k(i)$ for i=N+1,N+2, .. ,2N defined as :

$$\begin{bmatrix} u_k(N+1) \\ u_k(N+2) \\ . \\ . \\ u_k(\ 2N) \end{bmatrix} = \begin{bmatrix} g(\ 0\ ) \\ g(\ \Delta x) \\ g(2\Delta x) \\ . \\ . \\ g(N\Delta x) \end{bmatrix} \tag{3.7-1}$$

Therefore, Eqn.(3.4-1) can be written as a vector expression for each discrete point as follows :

$$\begin{bmatrix} u_{k+1}(0) \\ u_{k+1}(1) \\ . \\ . \\ u_{k+1}(N) \\ --- \\ u_{k+1}(N+1) \\ u_{k+1}(N+2) \\ . \\ . \\ u_{k+1}(2N) \end{bmatrix} = \begin{bmatrix} A(k) & | & F(k) \\ --- & + & --- \\ O & | & I \end{bmatrix} \begin{bmatrix} u_k(0) \\ u_k(1) \\ . \\ . \\ u_k(N) \\ --- \\ u_k(N+1) \\ u_k(N+2) \\ . \\ . \\ u_k(2N) \end{bmatrix} \tag{3.7-2}$$

where O is zero matrix, and I is unity matrix. The vector Eqn. (3.7-2) can be expressed more simply as :

$$U(k+1) = \Phi(k)U(k) \tag{3.7-3}$$

where

$$U(k) = [u_k(0),u_k(1),\ldots,u_k(N),u_k(N+1),u_k(N+2),..,u_k(2N)]^T, \tag{3.7-4}$$

$$A(k) = \begin{bmatrix} B_0 & A_0 & & & & \\ C_1 & B_1 & A_1 & & & \\ & C_2 & B_2 & A_2 & & \\ & & . & . & . & \\ & & & . & . & . \\ & & & & C_N & B_N \end{bmatrix}$$

(3.7-5)

$$F(k) = \begin{bmatrix} f_k & & & & \\ & f_k & & & \\ & & f_k & & \\ & & & \ddots & \\ & & & & f_k \end{bmatrix} \tag{3.7-6}$$

We include in the model a term $D\xi(k)$ which represents unknown random inputs and/or modeling errors by finite difference approximation so that

$$U(k+1) = \Phi(k)U(k) + D\xi(k) \tag{3.7-7}$$

The measurement data are $Z_1,Z_2,..,Z_i$. Therefore, the system measurement equation takes the form

$$Z(k) = HU(k) + \eta(k) \tag{3.7-8}$$

where $\eta(k)$ is measurement errors and H is the coefficient matrix. Assume that $\xi(k)$ and $\eta(k)$ are random Gaussian noise with the mean and covariance such that

$$E[\xi(k)] = E[\eta(k)] = 0, \tag{3.7-9}$$

$$E[\xi(k)\xi'(j)] = R\delta_{k,j}, \tag{3.7-10}$$

$$E[\eta(k)\eta'(j)] = Q\delta_{k,j}, \tag{3.7-11}$$

and

$$E[\xi(k)\eta(j)] = 0 \qquad \text{for } all\ k,j \tag{3.7-12}$$

The Kalman filter equations are, therefore, expressed as follows : Let $\hat{U}(k|k)$ be the optimal estimate of U(k), given the measurements data

$$Y(k) = [Z(0),Z(1), \cdots ,Z(k)] \tag{3.7-13}$$

Let the covariance of the estimation error be defined as

$$E\{[U(k) - \hat{U}(k|k)][U(k) - \hat{U}(k|k)]^T\} = P(k|k) \tag{3.7-14}$$

Where E{ } represents the conditional expectation. Then the optimal filter is given by the Kalman formulation :

$$\hat{U}(k+i|k+1) = \Phi(k)\hat{U}(k|k) + K(k+1)[Z(k+1) - C\Phi(k)\hat{U}(k|k)] \tag{3.7-15}$$

$$K(k+1) = P(k+1|k)C^T[CP(k+1|k)H^T + R]^{-1} \tag{3.7-16}$$

$$P(k+1|k) = \Phi(k)P(k|k)A(k)^T + DQD^T \tag{3.7-17}$$

$$P(k+1|k+1)=P(k+1|k)-P(k+1|k)C^T[CP(k+1|k)C^T+R]^{-1}CP(k+1|k) \tag{3.7-18}$$

with initial conditions

$$P(1|0) = A(0)P(0|0)A(0) + DQD^T \tag{3.7-19}$$

This algorithm has several impotant advantages :

(1) Not only the optimal estimate value of the $G(k)$ but also the state value $U(k)$ can be calculated sequentially on the basis of initial values.

(2) Not only the randomness of measurement data but also the randomness of $G(k)$ as well as finite difference approximation errors can be taken into consideration.

(3) There is no need to calculate α, as in the regularization method and the

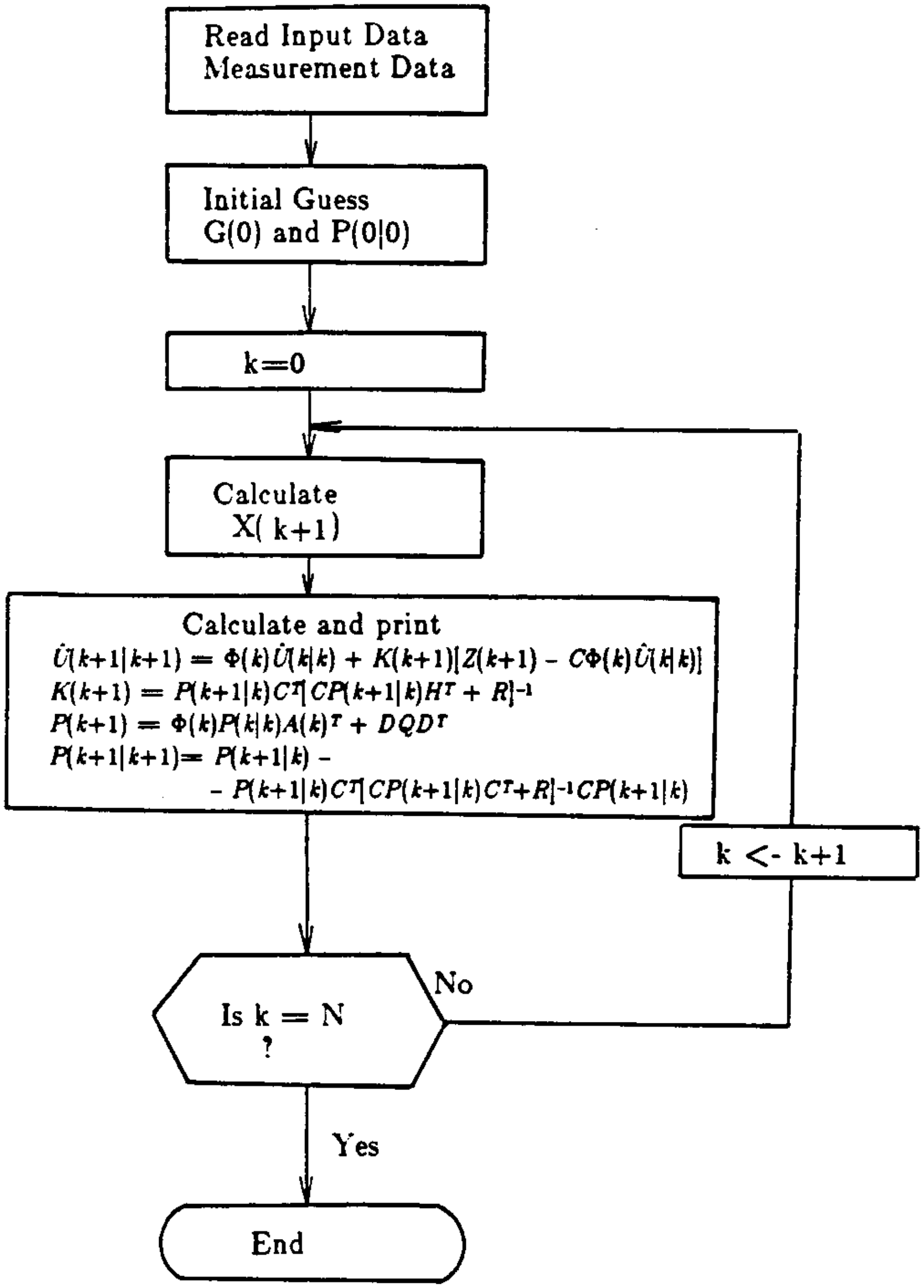

Read Input Data
Measurement Data
Initial Guess
G(0) and P(0|0)
k=0
Calculate
X(k+1)
Calculate and print
$\hat{U}(k+1|k+1) = \Phi(k)\hat{U}(k|k) + K(k+1)[Z(k+1) - C\Phi(k)\hat{U}(k|k)]$
$K(k+1) = P(k+1|k)C^T[CP(k+1|k)H^T + R]^{-1}$
$P(k+1) = \Phi(k)P(k|k)A(k)^T + DQD^T$
$P(k+1|k+1) = P(k+1|k) -$
$- P(k+1|k)C^T[CP(k+1|k)C^T + R]^{-1}CP(k+1|k)$
k <- k+1
Is k = N
?
No
Yes
End

Fig.3.4 Identification by Kalman Filtering

cross-validation method.

(4) Furthermore, time variant coefficients of system equations can be permitted to be included in the algorithm.

On the other hand, from the computational point of view,

(1) The vector and matrix size of the algorithm are twice and four times that of the least-square method, respectively. Therefore, the computation time will become more and more significant as the vector size increases even though the algorithm is a relating simple method.

(2) Again, the time function f(t) must be known in advance.

3.8 Dynamic Programming Method

So far the identification problems of the pollution location function g(x) were described while assuming that the time function $f(t)$ was known in advance. In this section, an identification problem of the time functions $f_i(t)$ is discussed while assuming the space function $g_i(\mathbf{x})$'s are known. This problem can be solved by employing the dynamic programming method which is based on the principle of the optimality of control theory. In order to comform to the dynamic programming method, the problem statement formulated in the section 3.3 must be modified.

Suppose that the function $g_i(x)'s$ which represent the pollution point source locations are known and their time function $f(t)'s$ are unknown so that the pollution sources F(x,t) can be expressed as ;

$$F(x,t) = \sum_{i=1}^{L} g_i(x) f_i(t) \tag{3.8-1}$$

$$= \sum_{i=1}^{L} \delta(x - x_{l_i}) f_i(t). \tag{3.8-2}$$

where x_{l_i}'s, the known pollution source locations, are assumed to be expressed as the multiple number of the discrete space interval Δx so that

$$x_{l_i} = i\Delta x \quad \text{for} \quad 1 \leq i \leq L < N \tag{3.8-3}$$

where L : the number of the pollution source locations

N : the number of the discrete space points

N must be larger than L because the number of the pollution point sources are restrected on space. The vector equation (3.4-3) can be modified using the equation (3.8-2) and (3.8-3) as follow :

$$U_{k+1} = AU_k + GF_k \tag{3.8-4}$$

where

$$F_k = \begin{bmatrix} f_k(l_1\Delta x) \\ f_k(l_2\Delta x) \\ \cdot \\ \cdot \\ f_k(l_L\Delta x) \end{bmatrix} \tag{3.8-5}$$

where the matrix G is determined by the pollution locations. Furthermore the controllability and observability with the equation (3.8-4) is assumed to be satisfied in order to be able to determine the F_k by the dynamic programming.

Therefore, the problems tatement can be changed as follows :

Problem Statement

Given the state equation

$$U_{k+1} = AU_k + GF_k, \qquad (3.8\text{-}6)$$

the output equation

$$Y_k = CU_k, \qquad (3.8\text{-}7)$$

and the observation system

$$Z_k = CU_k + n_k, \qquad (3.8\text{-}8)$$

find the optimal time function $\hat{F}_k$ that minimize the cost function

$$J = (Z_N - CU_N)^T(Z_N - CU_N)$$

$$+ \sum_{i=1}^{N-1}\{(Z_k - CU_k)^T(Z_k - CU_k) + (F_k^T RF_k)\} \qquad (3.8\text{-}9)$$

This problem can be regarded as a tracking problem to find the optimal control F_k that follows the desired function Z_k while minimizing the control cost. By the principle of the optimality, the optimal control F_k can be derived from the following iterative equation. Let us assume J_{N-k} to be the cost function needed to go from the $N-k^{th}$ stage to the N^{th} stage as

$$J_{N-k,N} = (Z_{N-k} - CU_{N-k})^T(Z_{N-k} - CU_{N-k})$$

$$+ F_{N-k}^T RF_{N-k} + J_{N-k+1,N} \qquad (3.8\text{-}10)$$

with

$$J_{N,N} = (Z_N - CU_N)^T (Z_N - CU_N) \tag{3.8-11}$$

The optimal control $\hat{F}_{N-k}$ must satisfy the condition.

$$\frac{\partial J_{N-k,N}}{\partial \hat{F}_{N-k}} = 0 \tag{3.8-12}$$

From this condition and the state vector equation (3.8-6), the following iterative equations for the optimal control values $\hat{F}_k$ can be derived (see Appendix B).

$$\hat{F}(N-k) = K(N-k)[Z(N+k-1) - CAU(N-k)] \tag{3.8-13}$$

$$K(N-k) = [R + B^T C^T P(k-1) CB]^{-1} C^T B^T P(k-1) \tag{3.8-14}$$

$$P(k) = P(k-1) - P(k-1)CB[R + B^T C^T P(k-1) CB]^{-1} B^T C^T P(k-1) \tag{3.8-15}$$

and

$$P(0) = I. \tag{3.8-16}$$

This equations can be calculated iteratively as indicated in Fig.3.4. First, the P(1) can be calculated using (3.8-15) with the initial condition (3.8-16). Then K(N-1) can calculated using (3.8-14) and P(1). The calculated K(N-1) is stored. For k=2, P(2) is calculated, and K(N-2) etc. This calculation is continued until k=N. Next the optimal time function F_k can be calculated using the stored K(N-k) and the state vector equation (3.8-6). This method has the following advantages :

(1) Multiple time functions which do not have periodical time pattern can be identified.

(2) The computer algorithm is simple.

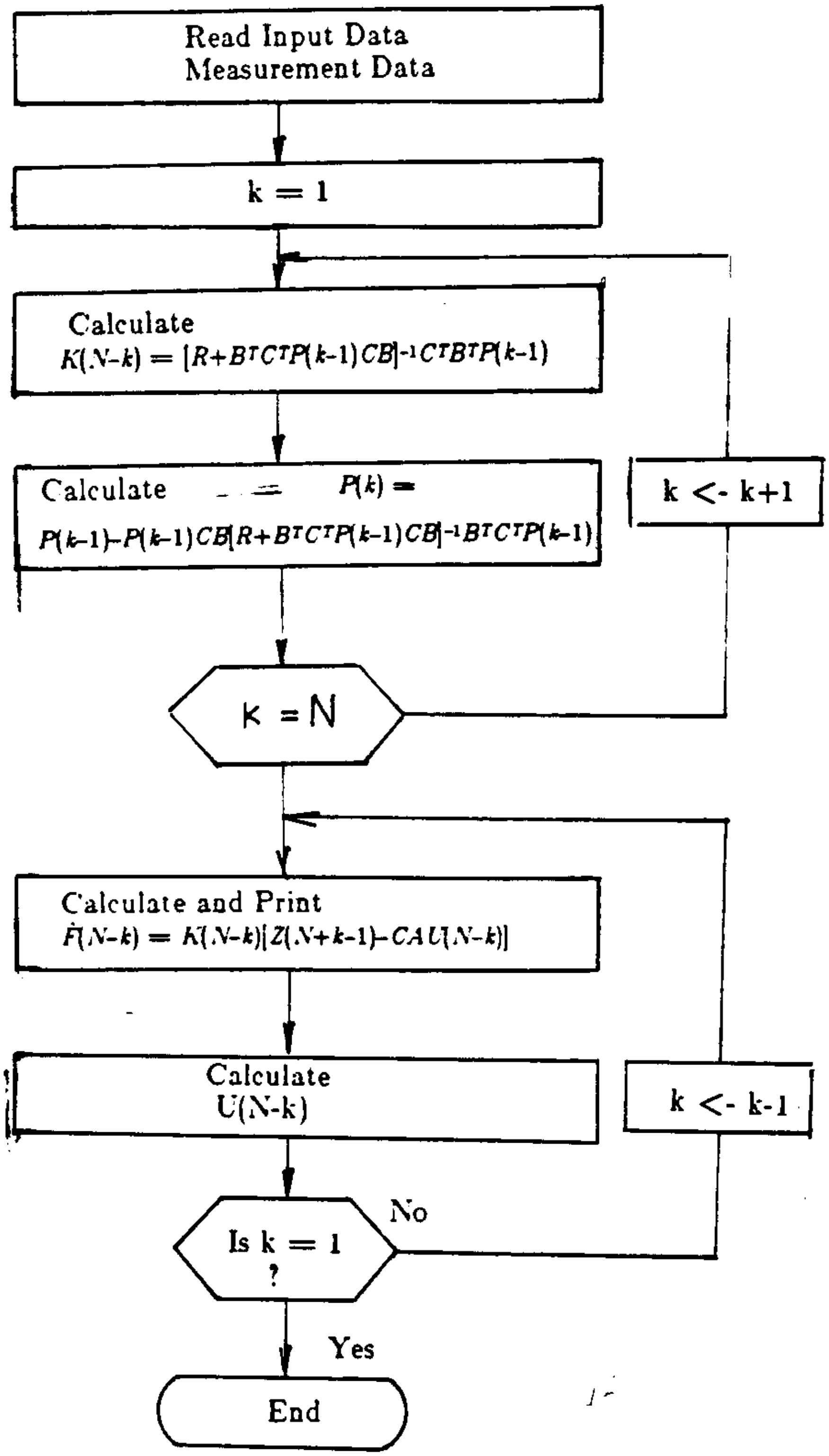

Read Input Data
Measurement Data
k = 1
Calculate
$K(N\text{-}k) = [R+B^TC^TP(k\text{-}1)CB]^{-1}C^TB^TP(k\text{-}1)$
Calculate $P(k) =$
$P(k\text{-}1)\text{-}P(k\text{-}1)CB[R+B^TC^TP(k\text{-}1)CB]^{-1}B^TC^TP(k\text{-}1)$
k <- k+1
k = N
Calculate and Print
$\hat{F}(N\text{-}k) = K(N\text{-}k)[Z(N+k\text{-}1)\text{-}CAU(N\text{-}k)]$
Calculate
U(N-k)
k <- k-1
No
Is k = 1
?
Yes
End

Fig.3.5 Identification by Dynamic Programming

However,

(1) Since the matrix K(N-k) must be stored. A lot of data memory are required. In particular, memory requirement becomes critical as the number of pollution locations L and the number of the time step N , N, increases.

(2) There is no general method to find the optimal weighting matrix. Therefore, the cross-validation method must be combined with this method.

(3) The pollution source locations must be known in advance.

3.9 Limitation of Conventional Methods

To model river pollution systems, several conventional methods are possible for the identification of pollution souces. The advantages and disadvantages of each method are discussed above. Improvements over the regularization method can also be effected. However, two major problems which are common to all of the conventional methods remain.

First, either the time function f(t) or the space function g(t), must be known in advance. However, this assumption is usually unrealistic because in most cases the time function is not known, and the space function g(t) also cannot be determined in advance particularly when there are multiple pollution sources. In another words, both g(x) and f(t) which are both component of F(x,t) must be identified simulatenously from measurement data. From this point, conventional identification methods are severely limited. It should be noted, however, that if one or both of f(t) and g(x) are precisely or even briefly determined in some way, conventional methods can be applied.

The second difficulty is that in all conventional methods computational burden increses sharply when there is an increase in the the number of space discreterization points, the number of measurement stations and the number of time steps.

CHAPTER 4

Pattern Recognition Approach

4.1 Introduction

This chapter introduces pattern recognition as a methodology for system identification in order to complement the conventional identification methods rather than replace them. General description of pattern recognition approaches and applications for pollution sources identification are formulated at each step. Finally, general of criteria to evaluate the performence of pattern recognition appproaches are discussed.

4.2 Definitions

Generally speaking, a pattern recognition system is a mechanism designed to use input data $\{z_1,z_2, \cdots ,z_n\}$ to classify an object into subclasses $\{S_1,S_2, \cdots ,S_m\}$ by means of extracting features $\{\gamma_1,\gamma_2, \cdots ,\gamma_l\}$ which characterize the subclass S_i. Its mechanism consists of four major steps or functional operations as shown in Fig.4.1. The sensor first observes a set of measurement data characterizing systems or objects to be classified. Next the observed data are conditioned in the preprocessor so as to enhance the observed signals and so as to reduce the effect of observation noise. Then the feature

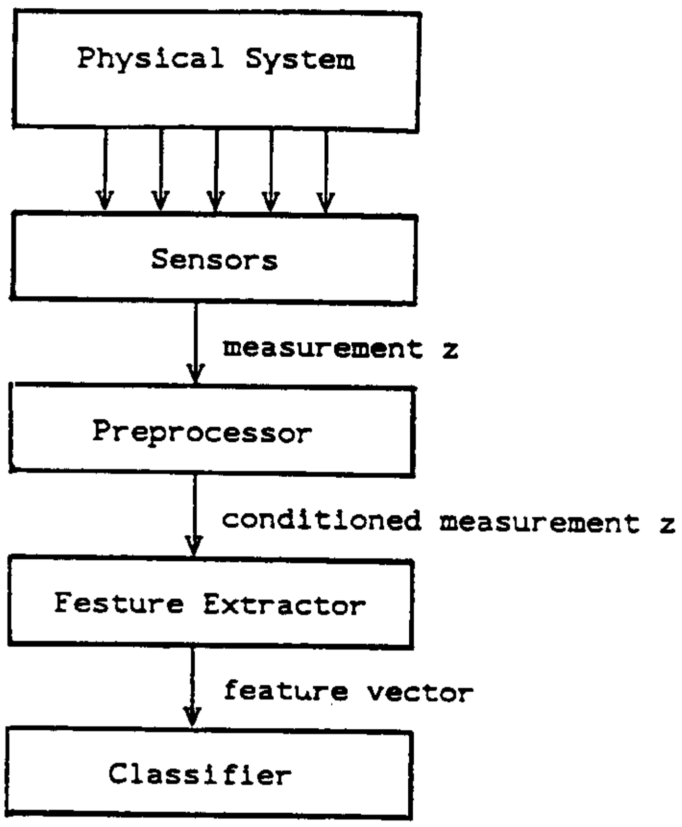


Fig.4.1

General Pattern Recognition System

extractor calculates the feature vector γ by performing suitable transformations of the observed data $\{z_i\}$. The feature transformation is derived so as to enhance the similarities of objects belonging to the same class while enhancing the difference of objects belonging to different classes. Finally, the classifier places the objects into a pattern class S_i. The classifier usually makes decision by using a discriminant function $d(\gamma)$ such that S_i is selected when

$$d_i(\gamma) > d_j(\gamma) \quad \text{for } i \neq j \tag{4.2-1}$$

where d_i is the i-th discriminant function.

4.2.1 Sensors

In order to enhance the validity of the pattern recognition method, it is desirable to obtain the number of measurement data of high quality. Since in water pollution studies, observations are usually restricted to a limited number of measurement poits and to discrete time intervals (or discrete sampling) for practical reasons, the next two major problems must be carefully considered.

(1) The locations of measurement stations,

(2) Sampling interval or sampling frequency.

The first problem arises from the fact that since continuous measuring sensors in space are physically impossble, a finite number observations must be made at optimal measuring locations so as to obtain as much imformation about systems as possible. This kind of problem is termed as an optimal sensor location problem. It can be proved that in linear distributed systems the optimal sensor location can be determined for a finite number of sensor [CuIc76] [AmBm77] [NaMi80]. However, it is quite difficult to determine the optimal locations for more general situations. In our research, the locations of the measurements are assumed to be specified and their optimal locations are not discussed.

The second problem, sampling frequency (or interval) is also important in the design and analysis of water pollution systems. Until recently because the resources available for observation were limited. However, the resources available for the observation of the environment have increased in recent years and the determination of the optimal utilization of these resources becomes of practical significance. The cost of operating monitoring network is likewise related to the sampling frequency. In another words, if sampling is done too often, data obtained are redundant and expensive, and on the other hand, if sampling is done too infrequently, some variable informations is lost. Therefore,the sampling frequency must be optimized in order to achieve realistic simulations. Some criteria to evaluate the optimal frequency are based on standard deviation of the measurement data. If standard deviation is larger, then more frequent samples obtained. Our research concerned with identification problems after collecting the observed data. the sampling problem therefore falls outside the scope of our research.

4.3 Pattern Recognition Systems for Pollution Sources Identification

The pattern recognition systems discussed in the previous section can be applied for identification of input function as pollution sources shown in Fig.4.2.

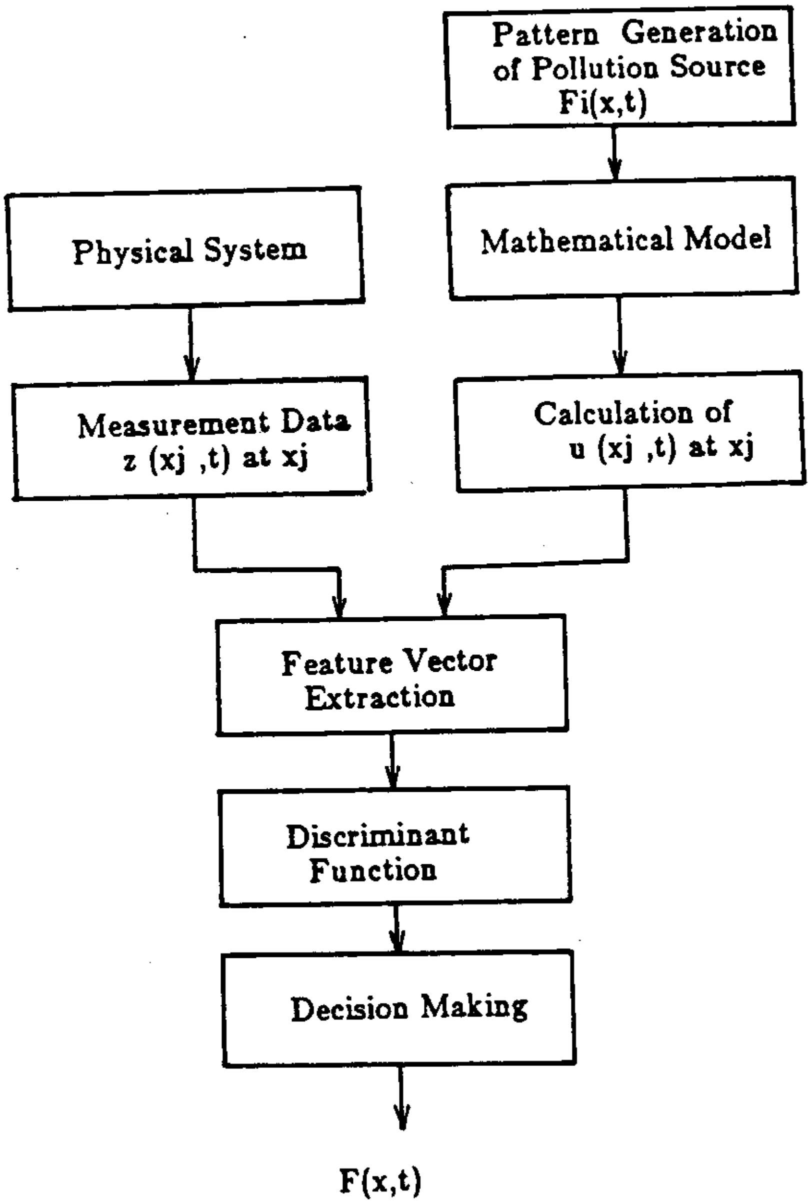

Pattern Generation of Pollution Source Fi(x,t)
Physical System
Mathematical Model
Measurement Data z (xj ,t) at xj
Calculation of u (xj ,t) at xj
Feature Vector Extraction
Discriminant Function
Decision Making
F(x,t)

Fig.4.2

Pattern Recognition for Pollution Sources Identification

First subclasses S_i of the input patterns $F_i(x,t)$ equivalent to the pollution sources are generated by considering a priori knowledge of the physical system under consideration. Using the $F_i(x,t)$, a mathematical model equivalent to the actual pollution system is solved to obtain the output $u_i(x_j,t)$ at x_j which are equivalent the measurement locations. On the other hand, observed data $z(x_j)$ from the physical system which contains unknown pollution sources are introduced to the pattern recognition system. Afterwords, the measurement data $z(x_j,t)$ and the output response $u_i(x_j,t)$ from the the mathematical model are processed in the feature extractor to generate feature vectors which characterize similarities and differences between both outputs. Finally decision making for all input pattern is done using discriminant function at the stage of classification which subclass the actual pollution sources belong to.

4.4 Pattern Generation

The first step in the design of any pattern recognition scheme is the designation of the pattern to be recognized. Each pattern generated is to correspond to a particular solution of the mathematical model for given input function. These patterns represent as subclasses of general pollution sources. Hence in this context pattern generation is equivalent to subclassing the pollution sources. The subclassing of a system S is results in a partitioning such that union of these subclasses contains S. There is usually some a priori information available from physical and engineering considerations to bound the class S. In some cases the a priori information is sufficient to constrain S to pollution sources which have the same mathematical expression as the actual pollution sources. Hence the identification problem of pollution sources is completely solved by conventional methods. However, frequently S contains many different structural subclasses S_i which contain different pollution source pattern and different source locations. Therefore, a particular S_i must be selected from a set $\{S_1,S_2,\ldots,S_n\} \epsilon S$. This selection is made directly or indirectly of the basis of the system observation $\{z(x_{j,t})\}$ and mathematical model.

4.4.1 Subclass Generation

Subclasses S_i of the class of the pollution sources S described by eqn (2.5-1) can be generated by

(1) selecting one or more possible pollution source locations x_{l_i} so that space function can be subclassified as follows ;

$$g_i(x) = \delta(x-x_{l_i}) \qquad \text{for } x_0 \leq x_{l_i} \leq x_f \tag{4.4-1}$$

(2) setting the time pattern function f_i with periodical components f_{Ti} so that time function can be subclassified as follows :

$$f_i(t) = f_{Ti}(t) \tag{4.4-2}$$

Type (1) is preferred when it is possible to classify the source locations roughly into a finite number of selected sources, although they can not be exactly specified. Some of these might not be problem or can be neglected, while one or more locations are suspected to be dominant pollution sources. In this case, subclasses S_i are generated as follows :

Let the nominated pollution source locations be $\{x_{l_1}, x_{l_2}, \ldots, x_{l_L}\}$. A reduced subclasses of pollution sources S_i can be described as

$$S_i : F_i(x,t) = \delta(x - x_{l_i}) f_i(t) \tag{4.4-3}$$

Type (2) is the case when the pollution sources can be characterized by periodic time pattern, such as daily, weekly, monthly, seasonal, or annual patterns. In this case the subclasses can generated as follows :

Let the periodic time functions with period be f_{Ti}. A reduced subclasses of pollution sources S_i can be described as

$$S_i : F_i(x,t) = g_i(x) f_{Ti}(t) \tag{4.4-4}$$

Since the periodic function can be expressed as Fourier series [OtEn78] [BePi71], Eqn. (4.4-4) can be alternatively described as

$$S_i : F_i(x,t) = g_i(x)\{1 + \sum_{k=1}^{K} (a_k cos(k\omega t) + b_k sin(k\omega t))\}$$

or

$$= g_i(x)\{1 + \sum_{k=1}^{K} a_k cos(k\omega t + \phi_k)\} \tag{4.4-5}$$

This kind of subclassing is very useful because the subclasses lead directly to the feature extraction, as will discussed in subsequent section.

4.4.2 Partitioning of S

In has been mentioned that the class of pollution sources S is partitioned into L subclasses such that ;

$$\bigcup_{i=1}^{l} S_i = S \qquad (4.4\text{-}6)$$

The partitioning of S is most effective for classification if subclasses S_i are generated such that the objective S_T belongs to only one subclass S_i. If objective S_T belongs to all the S_i, then the utility of classification is reduced. For this reason, the subclass should be generated so that S_i are a disjoint as possible.

4.5 Feature Selection and Extraction

The second stage of a pattern recognition system design includes the selection of features and the derivation and implementation of extractors for those features. This is commonly referred to as the feature extraction problem. In general, feature extraction is the most difficult problem in the design of a pattern recognition system. A survey of the literature on feature extraction indicates [Levi69] :

(1) No general theory exists to indicate how to select features which are relevant to a particular problem.

(2) Design of feature extractors is often heuristic employing many ad hoc strategies.

However. the following general remarks can be made concerning the feature extraction problem. In reference to feature selection, features are usually selected on the basis of their ability to enhance similarities of elements of the same class while enhancing the differences of elements of classes. No matter how good a features is in reference to its ability to discriminate between classes, it is of little utility if it cannot be extracted from measurement data.

For a general pattern recognition system the type of features extracted depends greatly on the form of the input. The data from the actual environment are transformed and preprocessed. Therefore, all features, in the context of the research described here, are of the mathematical transformations of measurements. Mathematical features are extracted by transformations, G, which map measurement data $\{z_i\}$ contained in a measurement space Z into a feature space Y, i.e,

$$\gamma = G(z) \tag{4.5-1}$$

where γ is a feature vector that is characteristic of some pattern. These transformations are derived utilizing some mathematical property of a subclass which allows it to be discriminated from other subclasses.

4.5.1 Feature Vectors from Correlation

It has been shown that the utilization of the correlation of systems input and output as features in pattern recognition to characterize non-linear discrete systems [SaHo74], and to classify the observed systems according to the order of the ordinary differential equation governing the system [THig74], are very successful. Correlation technique utilized in the literature can be applied in our research. The procedure to calculate correlation as features is as follows :

(1) First the subclass of pollution source $F_i(x,t)$ is generated. Using the $F_i(x,t)$, the mathematical medel is calculated in order to obtain the output $u(x_j,t_k)$ at x_j which are equivalent to the measuring locations.

(2) Then the auto-correlation $A_u(x_j)$ of $u(x_j,t_k)$ is calculated [OtEn78] [BePi71].

$$A_u(x_j) = \frac{1}{N}\sum_{k=1}^{N}[u^2(x_j,t_k)] \tag{4.5-2}$$

The auto-correlation $A_z(x_j)$ of the observed data $z(x_j,t_k)$ is also calculated.

$$A_z(x_j) = \frac{1}{N}\sum_{k=1}^{N}[z^2(x_j,t_k)] \tag{4.5-3}$$

(3) Then cross-correlation function between the calculated response $u(x_j,t_k)$ and the observed response $z(x_j,t_k)$ is calculated as follows :

$$C(x_j,n) = \frac{1}{N}\sum_{k=1}^{N}[z(x_j,t_k)u(x_j,t_{k+n})] \tag{4.5-4}$$

The normalized cross-correlation function is then calculated as follows :

$$\rho(x_j,n) = \frac{C(x_j,n)}{\sqrt{A_z(x_j)A_u(x_j)}}. \tag{4.5-5}$$

Step (1) through (3) are repeated for all the subclasses S_i and for the different measurement locations.

(4) The feature vectors γ_i for each subclass S_i are organized by taking the cross-correlation or normalized cross-correlation such as

$$\gamma_i = [C(x_1,n),C(x_2,n),\ldots,C(x_m,n)] \tag{4.5-6}$$

Or

$$\gamma_i = [\rho(x_1,n),\rho(x_2,n),\ldots,\rho(x_m,n)] \tag{4.5-7}$$

One of major advantage of utilizing correlations as feature is that they are invariant to any uncorrelated noise added to the observation. Another advantage is that the implementation of correlations can be easily done by signal processing with high-speed special purpose device. The disadvantage of the technique is that many correlations for each subclasses are required. Consequently, a great deal of computation is required. However, in a pattern recognition approach for source characterization, it may be possible to discriminate between subclasses on the basis of just a few correlations.

4.5.2 Feature Vector from Power Spectra

Features can be also derived from power spectra function, as they arise in typical signal processing, by transforming the time series data into the frequency domain. In the literature [GoFr74] [ToGo74], power spectra functions are employed as feature vectors in a pattern recognition system to characterize nuclear reactor component sureillance. Power specta function can be applied for the identification of pollution sources. The procedure to generate power spectra is :

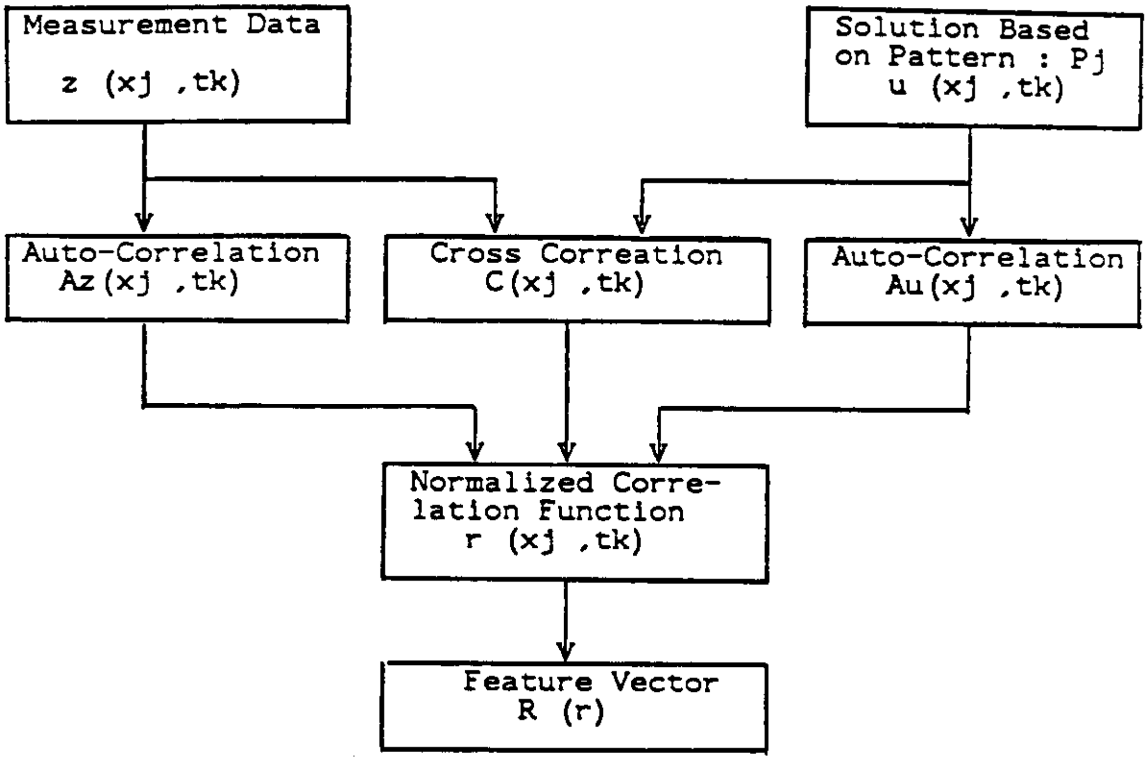


Fig.4.3

Feature Vector from Correlation Functions

(1) First the numerical solution u(x,t) over the region under consideration based on the nominated pollution source $F_i(x,t)$ belonging to subclass S_i, is calculated.

(2) The output response $u(x_j,t_k)$ at (x_j,t_k), equivalent to the measuring location and sample time respectively, is extracted.

(3) Then the output response $u(x_j,t_k)$ is transformed into frequency domain $U(x_j,f)$. This transformation is usually executed by Fast Fourier Transform (FFT) algorithm.

(4) The response $z(x_j,t_k)$ observed from the physical system is also transformed into frequency domain $Z(x_j,f)$ using the FFT.

(5) Next the power spectra functions $B_u(x_j,f)$ and $B_z(x_j,f)$

$$B_u(x_j,f) = \frac{1}{P}\overline{|U(x_j,f)|^2} \tag{4.5-8}$$

$$B_z(x_j,f) = \frac{1}{P}\overline{|Z(x_j,f)|^2} \tag{4.5-9}$$

and the cross power spectra function $B_{uz}(x_j,f)$ of both responses

$$C_{uz}(x_j,f) = \frac{1}{P}\overline{U^*(x_j,f)Z(x_j,f)} \tag{4.5-10}$$

are calculated by ordinary signal processing such as, window-functioning, smoothing and averaging as shown in Fig.4.4, where P indicates the resolution band of sampling, $\overline{[.]}$. means frequency average or unsemble average and $U^*(x_j,f)$ expresses conjugate complex number of $U(x_j,f)$. At the same time, the normalized cross power spectra function, the coherence function $r(x_j,f)$, is also calculated as follows [OtEn78] [BePi71] ;

$$r(x_j,f) = \frac{| C_{uz}(x_j,f) |^2}{B_u(x_i,f)B_z(x_i,f)}. \qquad (4.5\text{-}11)$$

Step (1) through (5) are repeated for all the subclasses S_i and for all different measurement locations.

(6) The feature vectors γ for each subclass S_i are organized by taking the cross spectra function as follows ;

$$\gamma = [|C_{uz}(x_1,f)|,|C_{uz}(x_2,f)|,\ldots,|C_{uz}(x_m,f)|] \qquad (4.5\text{-}13)$$

or

$$\gamma = [r(x_1,f),r(x_2,f),\ldots,r(x_m,f)]. \qquad (4.5\text{-}14)$$

The extraction of features from power spectra function has the advantage that the time function of the original function of the pollution sources can directly lead to the frequencies which are partitioned into subclasses S_i by Fourier expansion. In another words, features expressed as the frequency component, f_T, belonging to the subclass S_T, has also same frequency component as that of the original pollution sources. Therefore, a larger value of the feature element at some frequency, f_T, can be regarded as evidence of original time pattern.

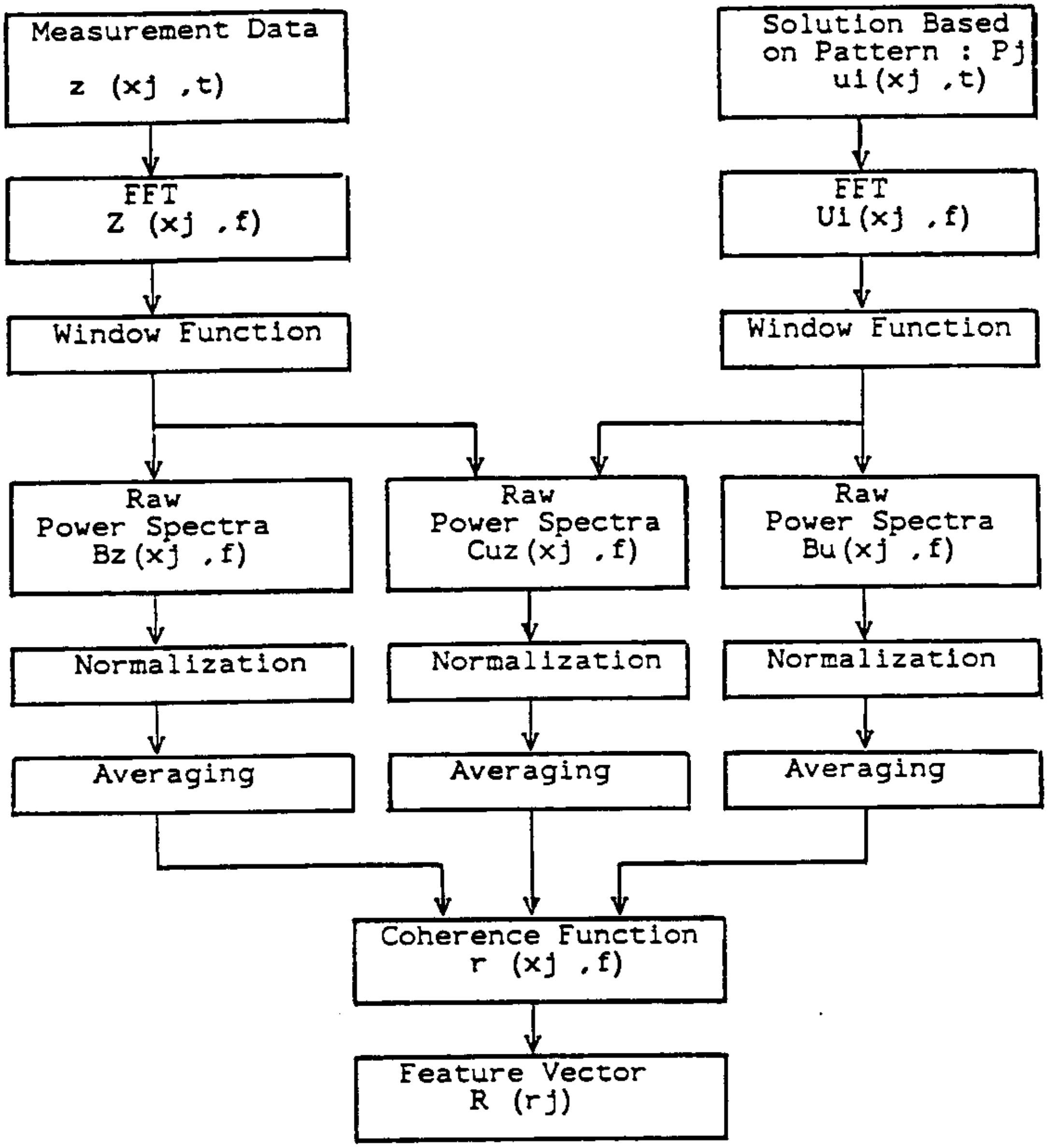


Fig.4.4

Feature Vector by Coherence Function

4.6 Classification

After extracting a set of features, task of classification is executed. Classification involves the making of the decision as to which of the specified pollution source patterns most nearly characterizes the original sources under concideration. This problem can be approached in a variety ways such as

(1) linear and non-linear discriminant functions,

(2) clustering,

(3) nearest-neighbor classification,

(4) stochastic approximation.

In particular, linear discriminant functions are attractive for following reasons;

(1) In some situations the weighting functions can be derived directly from considerations of the results of feature extraction.

(2) The training procedures for these types of classifiers are well documented.

Therefore, in our research, linear discriminant functions are employed. In feature extraction, a feature vector γ for a subclass of S can be obtained as :

$$\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_{n-1}]^T \tag{4.6-1}$$

With the inclusion of the residue subclass the feature vector of (4.6-1) becomes the augmented vector

$$\gamma_A = [\gamma_1, \gamma_2, \ldots, \gamma_{n-1}, 1]^T \tag{4.6-2}$$

The feature vector (4.6-2) indicates a point in n-dimensional space. If this point lies in a region in the n-dimensional space belonging to the i-th subclass, it is classified as S_i. Mathematically, the determination of region in the n-dimensional space in which the feature point lies is done linearly with a projection of the feature vector (4.6-2) onto a set of hyperplanes.

$$d_i(\gamma_A) = w\gamma_A \tag{4.6-3}$$

$$w^i = [w_1^i, w_2^i, \ldots, w_l^i, \ldots, w_n^i] \tag{4.6-4}$$

where w^i is the weight vector or hyperplane corresponding to the i-th subclass. A weight threshold w_r^i is chosen for the residue subclass.

$$w_r^i = w_n^i + w_e^i \tag{4.6-5}$$

where w_n^i is the weight threshold that would be set if the residue subclass did not contain the element S_T,(test fails), and where w_e^i is an added threshold to avoid misclassifications. Utilizing the preceding discriminant functions the decision rule is

$$S_i \quad \text{if} \quad d_i(\gamma_A) > d_j(\gamma_A) \quad i \neq j \tag{4.6-5}$$

and

$$d_i(\gamma_A) > 0 \tag{4.6-6}$$

otherwise the residue class is indicated.

The inequality (4.6-5) decides which of the non-residue subclasses S_i is likely to contain the distributed system description S_T. If the discriminant function evaluation also overcomes a threshold w_r^i, as represented by inequality (4.6-6), then a classification to S_i is made. Otherwise a classification to S residue is made. This latter classification implies that either S_T is not a member of any non residue S_i or that the test fails. The design of a classifier entails a determination of the numerical values of w_r^i and w_r^i. The condition under which the features are extracted (e.g. small errors in the feature extraction) may allow a heuristic setting of the numerical values of w_l^i and w_r^i. However, in most actual situations w_l^i and w_r^i must be determined through training in order to attain an optimal performance of the pattern recognition system.

Training the classifier encompasses the following steps:

(1) Assemble a training set $\{\gamma,S^i\} = \{\gamma^1,S^1;\gamma^2,S^2;\ldots;\gamma^k,S^k\}$ of k sample feature vectors, where γ^i and S^i denote respectively the feature vector and the label or classification information of the i-th sample feature vector.

(2) Select a functional J_e which, when minimized, yields the numerical values of w_l^i and w_r^i.

(3) Select an optimization algorithm to minimize J_e .

The first of the above steps entails :

(1) Select the elements of a subclass w^i and simulate these elements with normalized parameter values.

(2) Record the observations of this simulation (which correspond in time and space to actual measurement data taken from the distributed system being

modelled) and add noise to the observations.

(3) Apply the feature extraction transform to obtain the γ's.

(4) Repeat the above steps for i = 1, ... , n.

Assembling the training set is the biggest computational effort involved in the pattern recognition approach.
The primary reason for this is that during the training set generation, the system equation cannot be dissociated from the boundary conditions and the initial conditions even though the feature extractor may not need this information. For a typical case many simulations may be needed. Duda and Hart [DuHa72] listed several functions, J_e(w), and corresponding minimization algorithms that can be employed in training. Each functional and minimization algorithm has inherent dvantages and disadvantages which appl to any optimization process and do not relate directly to the problems at hand. In the course of this research a functional of the form

$$J_e(w)=\sum_{\gamma_e}(w\ \ \gamma)^2. \tag{4.6-7}$$

where γ_e are the misclassified feature vectors, is employed. The main advantage of this functional form is that an unique minimum exists and that the typical minimization algorithms are gradient type algorithms. That is, they are algorithms of the form

$$[w]^{k+1} = [w]^k - h_k D_k \nabla J_e([w]^k) \tag{4.6-8}$$

where $[w]^k$ is the k-th estimate of the weight vector, h_k is the step size and D_k is a positive definite matrix. The gradient of the functional is

$$\nabla J_e([w])^k = \frac{\partial J_e}{\partial w} \tag{4.6-9}$$

Since a classification to the residue subclass is always considered a correct classification (i.e. if the reside subclass contains the element S_T, the test fails), then in essence J_e(w) can be minimized to zero with classifications to the residue subclass to eliminate errors. Therefore, the object of the training is not only to minimize the misclassifications but also unnecessary classifications to the residue subclass. Thus the optimization process is first minimized over w_1^i , ... , w_n^i.Then the threshold w_e^i is adjusted so as to eliminate any misclassifications contributing to the residue of $J_e(w)$.

4.7 Performance Evaluation

Once the subclass most near to the original pollution source has been found, a computer simulation of the pollution system is implemented for performance evaluation. After characterizing the pollution sources F(x,t), the conventional identification methods are employed to specify the parameter values of g(t) and f(t). The performance of the pattern recognition system is evaluated by comparing the simulated output based on the selected pollution pattern F(x,t) with the measured data from actual system. If the selected pollution source is not satisfied, then the entire pattern recognition procedure is repeated and optimized using conventional identification methods. Performance evaluations including the conventional identification methods will be discussed in the chapter 8 more in detail.

CHAPTER 5

The Monte Carlo Method

5.1 Introduction

This chapter constitutes an introduction to the numerical implementations of the Monte Carlo methods for the solution of partial differential equations at the feature generation stage in pattern recognition methods. Finite difference equations to formulate the Monte Carlo method are derived by discretizing the original partial differential equations. Then the computer algorithms are formulated to execute the entire process of the Monte Carlo method. Finally the computational errors contained in the algorithm are characterized.

5.2 Monte Carlo Method for Partial Differential Equations

Since water quality problems are characterized by partial differential equations, it takes much computation time to calculate the solutions based on the subclass of input functions F(x,t). In the course of the classification procedure, the model PDE's have to be solved repeatedly for the many possible patterns F(x,t) in order to produce the calculated data at the measurement locations. Usually, standard numerical methods such as finite difference methods [VeKa81], finite element methods [Zien71] [DeAb72] or the method of characteristics

[BrPi73] [ReSu70] for solutions of PDE,s are employed. Those methods need to calculate the solution u(x,t) over the entire region even though only the a set of solutions (x_j,t) at the x_j equivalent to the measuring stations are required.

By contrary, as an alternative approach to the solution of PDE's, the Monte Carlo method [Waso51] [MeYe54] [King51] [Klah60] which is based on probability theory is employed to calculate the solutions $u(x_j,t)$ at only the x_j equivalent to the measuring stations. As an example, let us consider a simple two dimensional elliptic partial differential equation which describes a steady state aquifer flow system

$$a_x\frac{\partial^2 c}{\partial x^2} + a_y\frac{\partial^2 c}{\partial y^2} + b_x\frac{\partial c}{\partial x} + b_y\frac{\partial c}{\partial y} = 0 \quad (x,y) \in D \tag{5.2-1}$$

with boundary condition

$$c(x_b,y_b) = V(x_b,y_b) \quad (x_b,y_b) \in \partial D. \tag{5.2-2}$$

It is assumed that the solution $c(x_1,y_1)$ within D is required. The solution is carried out in the following steps :

(1) The original partial differential equation (5.2-1) is approximated by the finite difference method so that the domain D is discretized into regular grid points at which the difference equation is derived as follows :

$$c(x,y) = P_1c(x+\Delta x,y) + P_2c(x-\Delta x,y) + P_3c(x,y+\Delta y) + P_4c(x,y-\Delta y) \tag{5.2-3}$$

where

$$p_1 = \frac{1}{A}(\frac{a_x}{\Delta x^2}+\frac{b_x}{\Delta x}), \tag{5.2-4}$$

$$p_2 = \frac{1}{A}(\frac{a_x}{\Delta x^2}), \tag{5.2-5}$$

$$p_3 = \frac{1}{A}(\frac{a_y}{\Delta y^2}+\frac{b_y}{\Delta y}), \tag{5.2-6}$$

$$p_4 = \frac{1}{A}(\frac{a_y}{\Delta y^2}), \tag{5.2-7}$$

$$A = (\frac{2a_x}{\Delta x^2}+\frac{2a_y}{\Delta y^2}+\frac{b_x}{\Delta x}+\frac{b_y}{\Delta y}), \tag{5.2-8}$$

and the following equation is also satisfied :

$$p_1+p_2+p_3+p_4 = 1. \tag{5.2-9}$$

This finite difference equation can be regarded as a stochastic process with transition probabilities p_1 through p_4.

(2) A series of random walks (based on a random number generator) starting from location (x_1,y_1), which corresponds to the location of the measuring station are taken. The direction of each step of the random walks is determined by the random number ξ which is a member of a uniform distribution from 0 to 1 and by the transition probabilities p_1 through p_4 which are characterize by the stochastic equation shown in step 1, such that

ξ	Direction
$0 \leq \xi \leq p_1$	$(x,y) \rightarrow (x+\Delta x,y)$
$p_1 < \xi \leq p_1+p_2$	$(x,y) \rightarrow (x-\Delta x,y)$
$p_1+p_2 < \xi \leq p_1+p_2+p_3$	$(x,y) \rightarrow (x,y+\Delta y)$
$p_1+p_2+p_3 < \xi \leq 1$	$(x,y) \rightarrow (x,y-\Delta y)$

Each random walk is continued until it arrives at some point (x_b,y_b) on the boundary ∂D.

(3) When a random walk arrives at some point on the boundary, then the boundary value $V(x_b,y_b)$ on this point is recorded and accumulated.

(4) Steps (1) through (3) are repeated for a large number of iterations N.

(5) The average of the accumulated boundary values yields an approximate solution $u(x_1,y_1)$ such as

$$u(x_1,y_1) \approx \frac{1}{N}\sum_{i=1}^{N} V(x_b,y_b) \tag{5.2-10}$$

The Monte Carlo approach is computationally attractive because

(1) solutions for the entire space are not needed. In another words, calculations are required for only a few points equivalent to the measuring locations, and

(2) The random walks are independent of each other for different series of solutions and for different locations, so that concurrent calculations are

possible, and

(3) as shown below residual randomness in the computed solutions may be suppressed further when correlation functions and power spectra are calculated.

The Monte Carlo method has been most useful for the solution of two dimensional problems governed by Laplace's and Poisson's equations with Dirichlet boundary conditions [Waso51] [Lall54] [Shre66]. However, this method is equally applicable to parabolic partial differential equations [King51] [Shre66]. A generalized Monte Carlo method can be combined with the solution of a parabolic differential equations which describes a general pollution system

$$\frac{\partial u}{\partial t} = \sum_{j=1}^{N} \frac{\partial}{\partial x_j}\left[\sum_{i=1}^{N} a_{ij}\frac{\partial u}{\partial x_i} + a_j u\right] + \sum_{j=1}^{N} b_j \frac{\partial u}{\partial x_j} + bu + F \qquad (5.2\text{-}11)$$

with initial condition

$$u(x,t_0) = u_0(x) \quad x \in D \qquad (5.2\text{-}12)$$

and boundary condition

$$u(x_b,t) = u_b(t) \quad x_b \in \partial D \qquad (5.2.13)$$

The above procedure must be modified as follows :

(1) If the field parameters a_{ij}, a_j, b_j and b are functions of space x and t, their transition probabilities are functions of x and t. Therefore, transition probabilities must be calculated for the entire region within the boundary area.

(2) If the input function F exists, the values of the F at (x,t) at each step of the random walk falls must be accumulated.

(3) Because of the term $\frac{\partial u}{\partial t}$, each random walk must take the initial condition into consideration. In another words, a random walk must be stopped when it arrived at either the boundary or the initial condition t=0.

5.3 The Monte Carlo Method for River Pollution Systems

In order to derive the computer algorithm of the Monte Carlo method for the river pollution problem characterized by the one-dimensional partial differential equation

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + F(x,t) \qquad \text{for } x \epsilon D \tag{5.3-1}$$

where u(x,t) is the BOD, with initial condition

$$u(x,t_0) = u_0(x) \qquad \text{for } x\epsilon D \tag{5.3-2}$$

and boundary condition

$$u(x_b,t) = u_b(t) \qquad \text{for } x\epsilon\partial D, \tag{5.3-3}$$

Finite difference approximations are employed to obtain the difference equation as follows :

$$\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} \approx a\ \frac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{dx^2}$$
$$- b(x)\frac{u(x,t)-u(x-\Delta x,t)}{\Delta x} - cu(x,t) + F(x,t)$$

(5.3-4)

Here, we define $u(x,t)=u_k(x)$ where $t=k*DELTAt$. Therefore, Eqn. (5.3-4) can be expressed as:

$$u_{k+1}(x) = \left(\frac{a}{\Delta x^2}\right)\Delta t u_k(x+\Delta x) + \left(\frac{a}{\Delta x^2}+\frac{b(x)}{\Delta x}\right)\Delta t u_k(x-\Delta x)$$

$$+\left(\frac{1}{\Delta t}-\frac{2a}{\Delta x^2} - \frac{b(x)}{\Delta x} - c\right)\Delta t u_k(x) + \Delta t F_k(x) \qquad (5.3\text{-}5)$$

Moreover, Eqn. (5.3-5) can simply be expressed

$$u_{k+1}(x) = D\ \{p_1(x)u_k(x+\Delta x) + p_2(x)u_k(x-\Delta x) + p_3(x)u_k(x)\}$$

$$+ \Delta t F_k(x) \qquad (5.3\text{-}6)$$

where

$$D = \frac{a}{\Delta x^2}\Delta t + \left(\frac{a}{\Delta x^2} + \frac{b(x)}{\Delta x}\right)\Delta t + \left(\frac{1}{\Delta t} - \frac{2a}{\Delta x^2} - \frac{b(x)}{\Delta x} - c\right)\Delta t$$

$$= 1 - c\Delta t \qquad (5.3\text{-}7)$$

And, $p_1(x)$,$p_2(x)$ and $p_3(x)$ are expressed

$$p_1(x) = \frac{1}{D}\left(\frac{a}{\Delta x^2}\right)\Delta t \qquad (5.3\text{-}8)$$

$$p_2(x) = \frac{1}{D}\left(\frac{a}{\Delta x^2} + \frac{b(x)}{\Delta x}\right)\Delta t \qquad (5.3\text{-}9)$$

$$p_3(x) = \frac{1}{D}\left(\frac{1}{\Delta t} - \frac{2a}{\Delta x^2} - \frac{b(x)}{\Delta x} - c\right)\Delta t \qquad (5.3\text{-}10)$$

And where

$$p_1(x) + p_2(x) + p_3(x) = 1 \tag{5.3-11}$$

Therefore, equation (5.3-6) can be regarded as a stochastic process with transition probabilities (5.3-8) through (5.3-10). Based on the stochastic equation (5.3-7) and the transition probabilities (5.3-8) through (5.3-10), the computer algorithm can be developed as follows :

Suppose the solution $u(x_I,t_K)$ at $x=I\Delta x$ and $t=K\Delta t$ is required.

0) Calculate the transition probabilities $p_1(x)$, $p_2(x)$ and $p_3(x)$ at all x in the area under consideration and store in a table.

1) Set the time step index k at K. Set the space step index i at I so that $x=I\Delta x$. Set the resister Z to 0.

2) Generate the sequence of random number, ξ which is distributed from 0 to 1.

3) Start a random walk from point x as determined by $p_1(x)$,$p_2(x)$,$p_3(x)$ and ξ, and proceed to the neighboring grid point according to the following table

ξ	i	k	Direction
$0 \le \xi \le p_1$	$i \to i+1$	$k \to k-1$	$(x,t) \to (x+\Delta x, t-\Delta t)$
$p_1 < \xi \le p_1+p_2$	$i \to i-1$	$k \to k-1$	$(x,t) \to (x-\Delta x, t-\Delta t)$
$p_1+p_2 < \xi \le 1$	$i \to i$	$k \to k-1$	$(x,t) \to (x, t-\Delta t)$

and accumulate the input function value $\Delta t F_k(x)$ in resister Z.

$$Z \text{ <----- } \Delta t F_k(x) D^{(K-k)}$$

4) If the random walk arrives at a boundary point x_b for k > 0, then the boundary potential $u_b(t)$ is accumulated.

$$Z \text{ <----- } u_b(t) D$$

If the random walk is found to be an internal node for k = 0, then the initial value $u_0(x)$ is accumulated.

$$Z \text{ <----- } u_{0(x)D}$$

5) Continue the process (steps 2 to 4) for a large number of iterations N.

6) Calculate the average value of N trials and print it. This result is the approximate solution.

$$u(x_I, t_K) \approx \frac{Z}{N}$$

The entire process (1) through (6) can be repeated for different times t = Kdt. This process can be also applied for two or three dimensional river pollution systems by only changing transition probabilities.

5.4 The Monte Carlo Method for Aquifer Systems

The Monte Carlo method can be also applied to aquifer systems. Let us consider the two-dimensional parabolic partial differential equation.

$$S\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left(K_x\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial x}\left(K_y\frac{\partial h}{\partial y}\right)+Q(x,y,t) \tag{5.4-1}$$

with initial condition

$$h(x,y,t_0) = h_0(x,y) \tag{5.4-2}$$

and boundary condition

$$h(x_b,y_b,t) = h_b(t) \tag{5.4-3}$$

By the finite difference method, Eqn. (5.4-1) is approximated as a difference equation

$$\begin{aligned} h_{k+1}(x,y) &= p_1(x,y)h_k(x+\Delta x,y)+p_2(x,y)h_k(x-\Delta x,y) \\ &+p_3(x,y)h_k(x,y+\Delta y)+p_4(x,y)h_k(x,y-\Delta y) \\ &+p_0(x,y)h_k(x,y)+\Delta tQ_k(x,y) \end{aligned} \tag{5.4-4}$$

where

$$p_1(x,y) = \frac{\Delta t}{\Delta x^2}K_{x+\Delta x} \tag{5.4-5}$$

$$p_2(x,y) = \frac{\Delta t}{\Delta x^2} K_{x-\Delta x} \tag{5.4-6}$$

$$p_3(x,y) = \frac{\Delta t}{\Delta y^2} K_{y+\Delta y} \tag{5.4-7}$$

$$p_4(x,y) = \frac{\Delta t}{\Delta y^2} K_{y+\Delta y} \ . \tag{5.4-8}$$

and

$$p_0+p_1+p_2+p_3+p_4 = 1 \tag{5.4-9}$$

Equation (5.4-4) can be regarded as a stochastic process with transition probabilities p_0 through p_4. Therefore, based on Eqn.(5.4-4) through (5.4-9), the computer algorithm can be developed similar to that of the river pollution problem.

Suppose the solution $h(x_I,y_J,T)$ at $x=I\Delta x, y=J\Delta y$ and $t=T=K\Delta t$ is required.

0) First calculate the transition probabilities p_0 through p_4 at all x,y in the domain D under consideration and store it in a table.

1) Set the time step index k to K. Set the space step indices (i,j) to (I,J) so that $x=I\Delta x$ and $y=J\Delta y$. Set the resister Z to 0.

2) Generate the sequence of random numbers ξ which distributed uniformly from 0 to 1.

3) Start a random walk from the point x,y such that the direction of the first step is determined by

ξ	i,j	k	Direction
$0\le\xi\le p_1$	$(i,j)\to(i+1,j)$	$k\to k-1$	$(x,y,t)\to(x+\Delta x,y,t-\Delta t)$
$p_1<\xi\le p_{12}$	$(i,j)\to(i-1,j)$	$k\to k-1$	$(x,y,t)\to(x-\Delta x,y,t-\Delta t)$
$p_{12}<\xi\le p_{13}$	$(i,j)\to(i,j+1)$	$k\to k-1$	$(x,y,t)\to(x,y+\Delta y,t-\Delta t)$
$p_{13}<\xi\le p_{14}$	$(i,j)\to(i,j-1)$	$k\to k-1$	$(x,y,t)\to(x,y+\Delta y,t-\Delta t)$
$p_{14}<\xi\le 1$	$(i,j)\to(i,j)$	$k\to k-1$	$(x,y,t)\to(x,y,t-\Delta t)$

where

$$p_{12} = p_1 + p_2 \qquad (5.4\text{-}10)$$

$$p_{13} = p_1 + p_2 + p_3 \qquad (5.4\text{-}11)$$

$$p_{14} = p_1 + p_2 + p_3 + p_4 \qquad (5.4\text{-}12)$$

and accumulate the input function value $\Delta t Q_k(x,y)$ in resister Z.

Z <----- $\Delta t Q_k(x,y)$

4) If the random walks arrives at a boundary point (x_b,y_b) for k > 0 , the boundary potential $u_b(t)$ is accumulated.

Z <----- $h_b(t)$

If the random walk arrives at a initial point (x,y) for k = 0, then the initial value $h_0(x,y)$ is accumulated.

$$Z \longleftarrow h_0(x,y)$$

5) Continue the process (steps 2 to 4) for a large number of iterations N.

6) Calculate the average value of N trials and print it. This result is the approximate solution at $x=x_I$,$y=y_J$ and $t=t_K$,

$$u(x_I,y_J,t_K) \approx \frac{Z}{N}$$

The entire process (1) through (6) can be repeated for different time $t=K\Delta t$. This process can be applied for both aquifer flow problems and quality problems with two or three dimensional equations

5.5 Errors in the Monte Carlo Method

There are several errors involve in the Monte Carlo method. In particular the following four sources of errors are significant [Shih74] [Thud65] [Shre67] to analyze the accuracy of the solution of partial differential equations.

(1) Local truncation error : E_l

(2) Statistical error : E_s

(3) Random number error : E_r

(4) Gross error : E_g

The local truncation error, E_l, is caused by the finite difference approximation of the original partial differential equations. Since E_l is equivalent to the truncation error which involves in the finite difference method, the error can be analytically evaluated. In general E_l can be reduced by decreasing the space step size Δx and time step size Δt.

Statistical error, E_s, is due to the fact that the number of random walk trials, N, are finite. Since the Monte Carlo method is based on the low of probability, a large number of random walks should be taken in order to obtain the solutions of partial differential equations. E_s for the finite number of trials can be evaluated by using the central limit theory as follows:

Let Z_i (equivalent to the value in the resister Z for i^{th} trial) be random variables with the expected value

$$E[Z_i] = m \tag{5.5-1}$$

and variance

$$Var[Z_i] = b^2 \tag{5.5-2}$$

As discussed in Sec. (5.2), the solution $u(x,y)$ at (x_1,y_1) of the two dimensional partial differential equation in Eqn.(5.41) by the Monte Carlo Method is given by

$$u(x_1,y_1) \approx \frac{1}{N}\sum_{i=1}^{N} Z_i \tag{5.5-3}$$

By the central limit theorem of the theory of probability, the distribution of $u(x_1,y_1)$ for large number of N can be approximately expressed by Gaussian distribution with the expected value

$$E[u(x_1,y_1)] = m \tag{5.5-4}$$

and variance

$$Var[u(x_1,y_1)] = \sigma^2 = \frac{b^2}{N} \tag{5.5-5}$$

Therefore, by applying the rule of "Three Sigma" to the distribution of $u(x_1,y_1)$, the following equation is obtained.

$$P\{m-3\sigma < u(x_1,y_1) < m+3\sigma\} = 0.997 \tag{5.5-6}$$

By substituting Eqn.(5.5-5) into Eqn.(5.5-6), the following equation is obtained.

$$P\{m-\frac{3b}{\sqrt{N}} < u(x_1,y_1) < m+\frac{3b}{\sqrt{N}}\} = 0.997 \tag{5.5-7}$$

Eqn. (5.5-7) can be rewritten as follow :

$$P\{|u(x_1,y_1)-m|<\frac{3b}{\sqrt{N}}\} = 0.997 \tag{5.5-8}$$

Eqn. (5.5-8) means that the probability that the absolute value of the statistical error, E_s, is smaller than $3b/\sqrt{N}$ is almost 1. In another words, E_s can be restricted by both b and N as follows :

$$E_s = |\, u(x_1,y_1) - m| < \frac{3b}{\sqrt{N}} \tag{5.5-9}$$

In general, it is impossible to exactly calculate the variance b^2 because it depends on the transition probabilities, the boundary regions and the values. However upper bound of the variance can be approximately restricted by selecting the maximum possible value of Z_i, max$|Z_i|$, although this is not best tight bound. Therefore, upper bound of the statistical error of the solution by the Monte Carlo method can be restricted as follows :

$$E_s < 3\frac{max|Z_i|}{\sqrt{N}} \qquad (5.5\text{-}10)$$

It is noted E_s cannot be so much improved even the number of the trials N is increased because of deviation of $\sqrt{N}$.

Random number error, E_r is originated by the non-uniform distribution or biased ditribution of ramdom number which is generated by ramdom number generater of computer subroutine. This kind of error can be avoided by changing the seed of ramdom number generater after enough number of generation or selecting the different randum number generater. It is important to evaluate the statistical nature of the generated random number in order to confirm whether the distribution of the generated random number is uniform and unbiased.

Gross error, E_g, is essential error of the Monte Carlo method which uses the finite step size of random walks and the restricted directions (i.e., four neighbourhood grid points). This kind of error can be reduced by selecting the smaller value of the step size and/or high order finite difference approximation of the original equation. On the other hand, the computation time increases as the number of grid points is incresed.

5.6 Numerical Implementations

In this section, numerical implementation of the Monte Carlo method to solve the parial differential equation for the two-dimensional steady state aquifer flow system and the one-diemnsional river pollution system is presented. The solutions by the Monte Carlo method are compared with the solutions by the finite difference method with convergence and computation speed.

5.6.1 Two-Dimentional Aquifer flow Systems

In order to demonstrate the implementation of the Monte Carlo method for the solution of the aquifer flow system, the follwing simple two-dimensional steady state equation in Eqn. (5.2-1) is employed.

$$a_x \frac{\partial^2 u}{\partial x^2} + a_y \frac{\partial^2 u}{\partial y^2} + b_x \frac{\partial u}{\partial x} + b_y \frac{\partial u}{\partial y} = 0 \qquad (x,y) \in D \tag{5.6-1}$$

with the boundary region

$$0 \le x \le x_b, \qquad 0 \le y \le y_b \tag{5.6-2}$$

boundary condition

$$\begin{aligned} u(x,y_b) &= 1.0 \quad \text{for } 0 \le x \le x_b \\ \textit{otherwise } u(x,y) &= 0.0 \end{aligned} \tag{5.6-3}$$

As the numerical values of the parameters, the boundary region and the space step size, the following dada are employed :

$$a_x = 0.35 \tag{5.6-4}$$

$$a_y = 0.30 \tag{5.6-5}$$

$$b_x = 0.25 \tag{5.6-6}$$

$$b_y = 0.10 \tag{5.6-7}$$

$$x_b = 2.0 \tag{5.6-8}$$

$$y_b = 2.0 \tag{5.6-9}$$

$$\Delta x = \Delta y = 0.05 \tag{5.6-10}$$

From Eqn. (5.2-3), the finite difference equation can be expressed as follows :

$$u(x,y) = 0.35u(x+\Delta x,y) + 0.3u(x-\Delta x,y) + 0.25u(x,y+\Delta y) + 0.1u(x,y-\Delta y) \tag{5.6-11}$$

It is assumed that the solution $u(0.3,0.3)$ is required. The solution u(0.3,0.3) is calculated by both the Monte Carlo method and the Jacobi method which is classified as iterative finite difference method. The solution by the Jacobi method is as follows :

$$u^{(i+1)}(x,y)=0.35u^{(i)}(x+\Delta x,y)+0.3u^{(i)}(x-\Delta x,y)+0.25u^{(i)}(x,y+\Delta y)+0.1u^{(i)}(x,y-\Delta y) \tag{5.6-12}$$

with the initial starting value

$$u^{(0)}(x,y) = 0 \qquad \text{for } 0 \leq x \leq x_b, \; 0 \leq y \leq y_b \tag{5.6-13}$$

Fig.5.6.1 presents the solution u(0.3,0.3) by the Monte Carlo method and Jacobi method for changing the number of random trials, M, and the number of iteration, I. The solution by the Monte Carlo method gives rough solution for relative small number of radom trails (more than 200 trials). On the other hand, the solution by the Jacobi method needsmore than 1000 iteration to give the same accuracy as the Monte Carlo method. However, the Jacobi method converge at more than 2000 while the Monte Carlo method need more than 10000 random trials. Fig.5.6.2 presents the computation time for the calculations of u(0.3,0.3) in Fig.5.6.1 by both methods. Obviousely, the Monte Carlo method is about ten time faster than the Jacobi method for the same number of iterations. As results, the Monte Carlo method can calcualte rough solutions with high speed. As reference, Fig.5.6.3 presents the computation time for the solutions of the three-

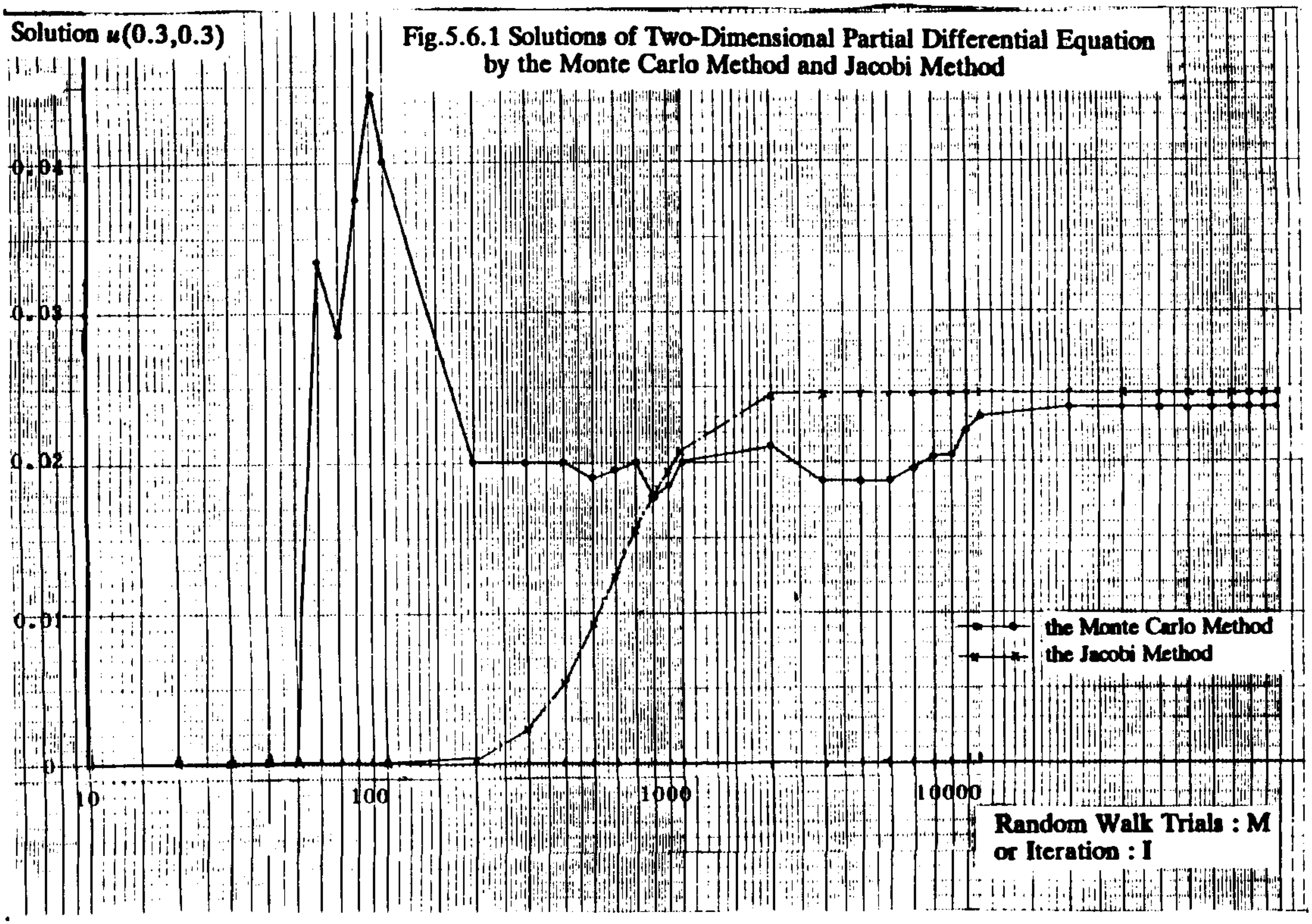


Fig.5.6.1 Solutions of Two-Dimensional Partial Differential Equation by the Monte Carlo Method and Jacobi Method

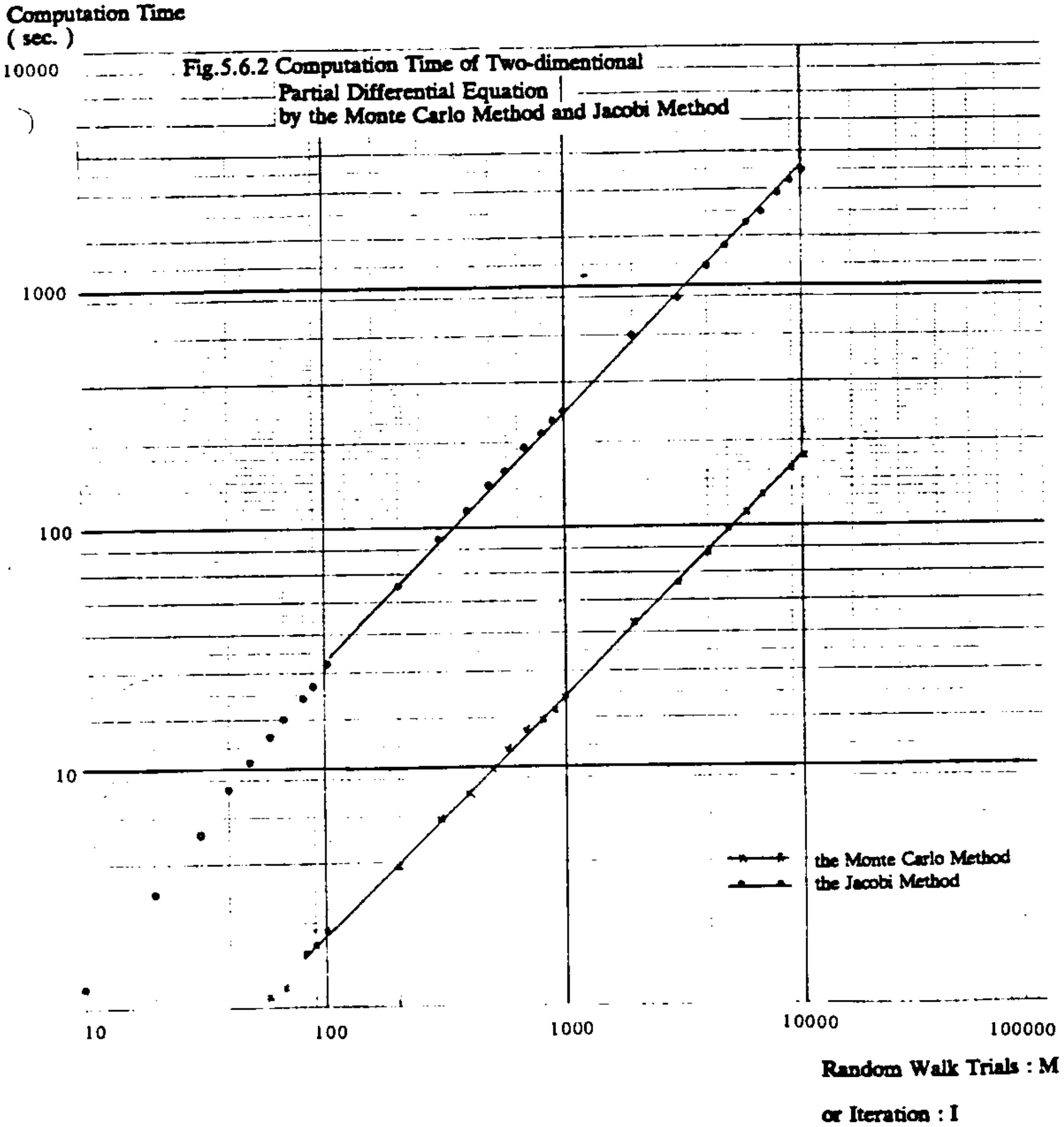


Fig.5.6.2 Computation Time of Two-dimentional Partial Differential Equation by the Monte Carlo Method and Jacobi Method

dimensional aquifer flow equation by the both methods. In this calculation, the parameter values and boundary region and boundary values are the same as in case of the two-dimensional case. It is obviouse that the computation time by the Monte Carlo method is only twice as the time for two-dimensional eqation, whress, the computation time by the Jacobi method extremely increases compared with in the case of the two-dimensional equation. Therefore, the Monte Carlo method is more powerful for the calculations of high-dimensional partial differential equations.

5.6.2 One Dimentional River Pollution System

The numerical values for one dimensional river pollution system is assumed as follows :

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + F(x,t) \quad \text{for } x \epsilon D \qquad \text{for } x \epsilon D \tag{5.6-14}$$

with initial condition

$$u(x,t_0) = u_0(x) \quad \text{for } x \epsilon D \tag{5.6-15}$$

and boundary condition

$$u(x_b,t) = u_b(t) \quad \text{for } x \epsilon \partial D, \tag{5.6-16}$$

Finite difference approximations are employed to obtain the difference equation as follows :

$$\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} \approx a\,\frac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{dx^2}$$

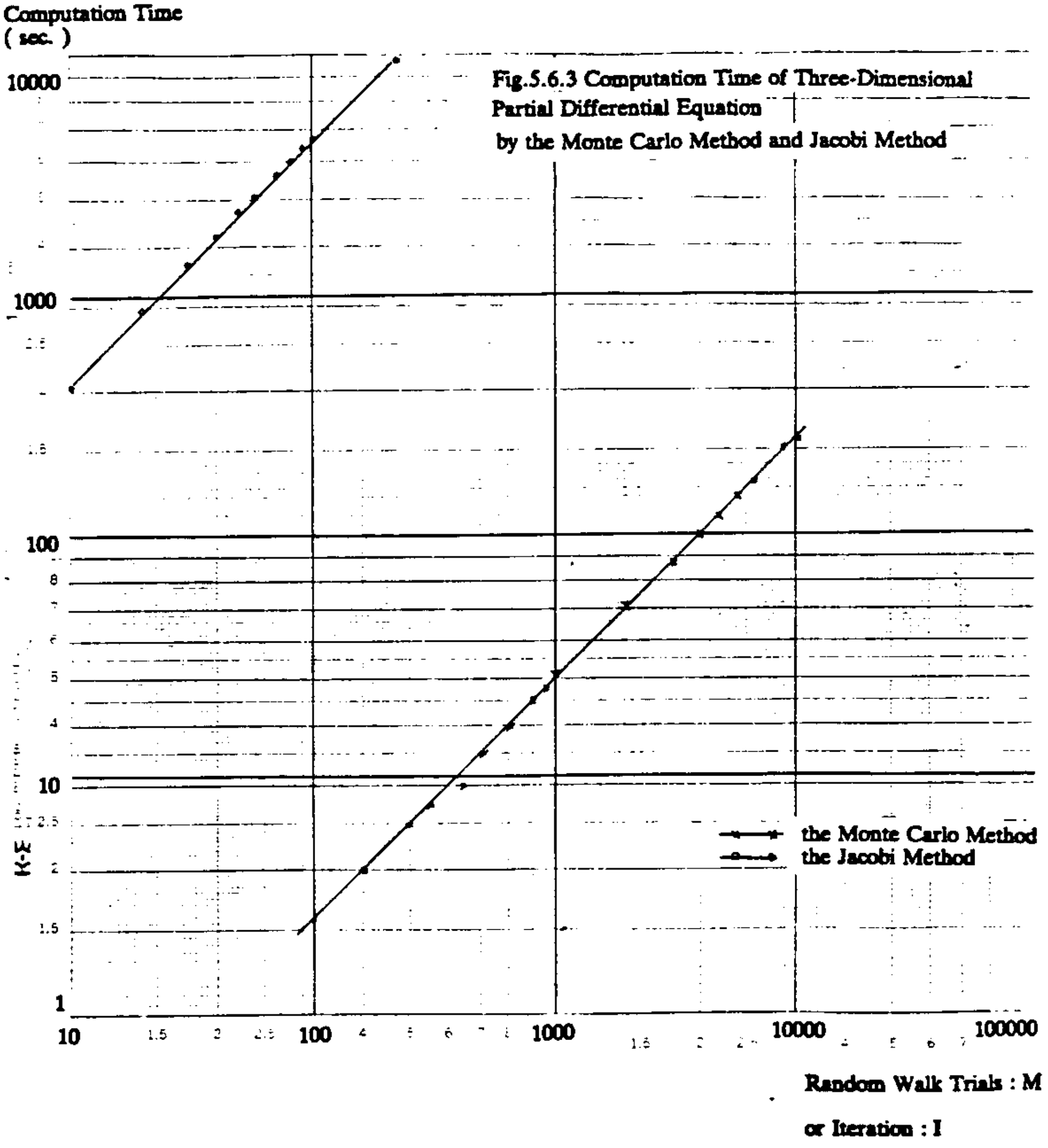


Fig.5.6.3 Computation Time of Three-Dimensional Partial Differential Equation by the Monte Carlo Method and Jacobi Method

$$- \overset{*}{b}(x)\frac{u(x,t)-u(x-\Delta x,t)}{\Delta x} - cu(x,t) + F(x,t) \tag{5.6-17}$$

Here, we define u(x,t)=$u_k(x)$ where t = k*Δt. Therefore,

$$u_{k+1}(x) = D\ \{p_1(x)u_k(x+\Delta x) + p_2(x)u_k(x-\Delta x) + p_3(x)u_k(x)\} \\ + \Delta t F_k(x,t) \tag{5.6-18}$$

The numerical values of the system are assumed as follows :

$$a = 0.5 \tag{5.6-19}$$

$$b = 0.5 \tag{5.6-20}$$

$$c = 0.1 \tag{5.6-21}$$

$$F(x) = sin(x) \tag{5.6-22}$$

$$u_0(x) = 0.0 \tag{5.6-23}$$

$$u_{x_0} = 0.0 \tag{5.6-24}$$

$$u_{x_f} = 0.0 \tag{5.6-25}$$

The step size and the boundary region are

$$\Delta x = 0.1\pi \tag{5.6-26}$$

$$\Delta t = 0.02 \tag{5.6-27}$$

The time and the space regions are

$$0 \leq t \leq 20.0 \qquad 0 \leq x \leqslant 4.0\pi \tag{5.6-28}$$

Therefore, the number of the space step, N, and the number of the time step, K are

$$N=40 \tag{5.6-29}$$

$$K = 500 \tag{5.6-30}$$

It is assumed to solve the solution $u(x_m,T)$ where

$$x_m = 1.5\pi \quad T = 10.0 \tag{5.6-31}$$

Using these numerical values, the transition probabilities are as follows :

$$p_1 = 0.10152 \tag{5.6-32}$$

$$p_2 = 0.13342 \tag{5.6-33}$$

$$p_3 = 0.76606 \tag{5.6-34}$$

As the random number generation, the conguentional multiply method was employed because of its simplicty and fast generation. The computational results are presented in Fig.5.6.3 which shows the solution $u(1.5\pi,10.0)$ by both the finite difference method and the Monte Carlo method for different number of random walk trials, M. From the computational stability condition of the explicit finite difference method, the time step size must be satisfied :

$$\Delta t < \frac{1}{\frac{a}{\Delta x^2} + \frac{b}{\Delta x} - c} \tag{5.6-35}$$

This condition is also required for the Monte Carlo method in order that the all the transition probabilities P_i for i=1 to 3 are

$$P_i < 1.0 \text{ for } i=1,3 \tag{5.6-36}$$

It can be seen that the solution by the Monte Carlo method graduateley approaches to the solution by the explicit finite difference method as the number of random walk trials, M is increaed. The relative error for the solution by the

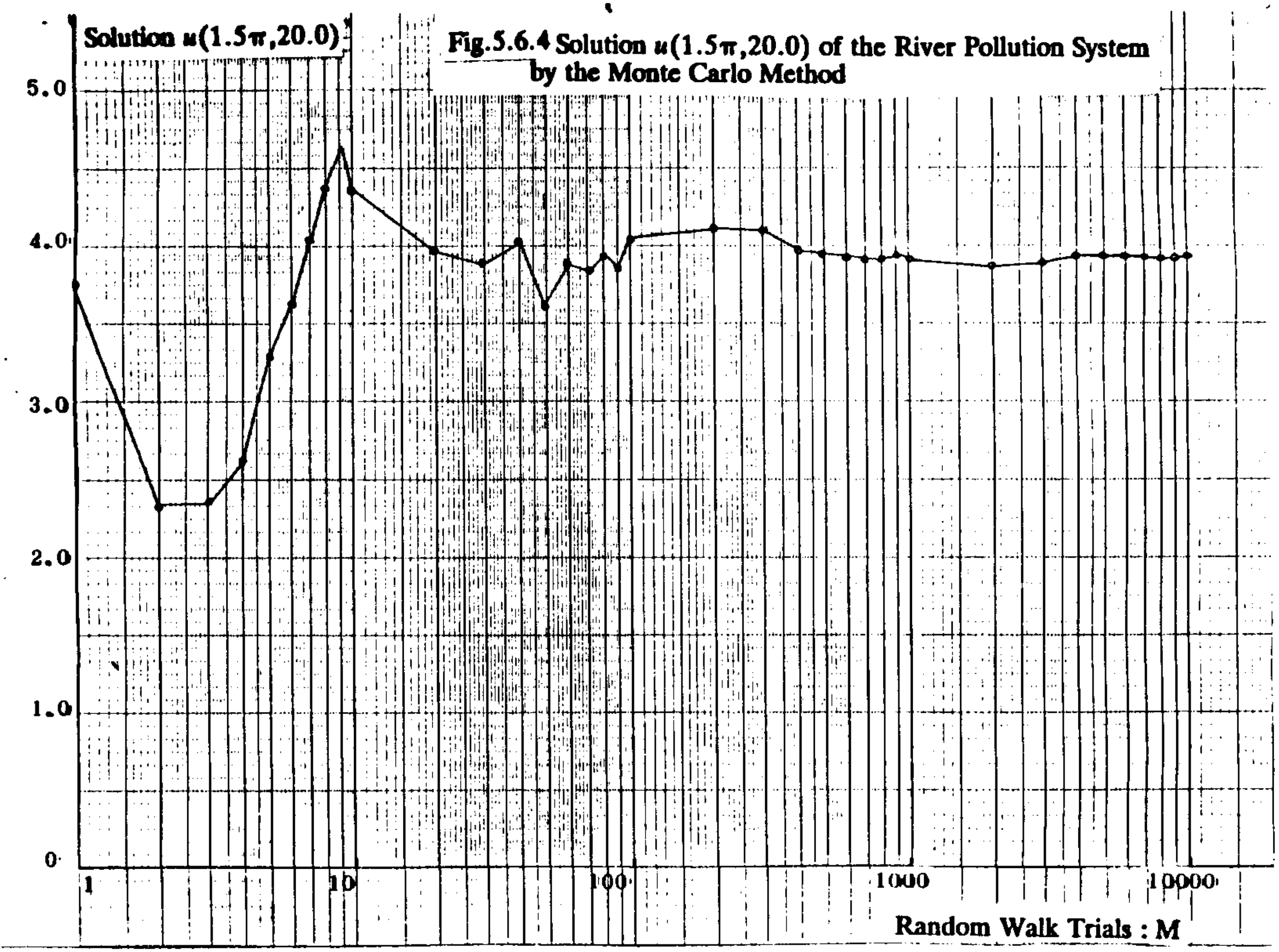


Fig.5.6.4 Solution $u(1.5\pi,20.0)$ of the River Pollution System by the Monte Carlo Method

finite difference method can be attained within 1.0 % for more than 100 random walk trials. Fig.5.6.4 presents the CPU time of the VAX-11/750 for calcultions of the solution of the $u(1.5\pi,10.0)$ by both the explicit finite difference method and the Monte Carlo method. In order to calculate the solution $u(1.5\pi,10.0)$ by the finite difference method, the finite difference calculation must be executed not only for all the grid points x_j ($0 \leq x_j \leq 4.0\pi$) but also for all the time steps t_i ($0 \leq t_i \leq 10.0$). In the Monte Carlo method, on the other hand, the solution $u(1.5\pi,10.0)$, which is independent to the other grid points and other time steps, can be directly calculated. Fig.5.6.5 presents the solutions of $u(1.5\pi,t)$ for each time. For the transient period, the solutions by the Monte Carlo method are almost same to the solution by the finite differece method, on the other hand, for a steady state, the solutions behave like Gaussian distribution with the mean which is equivalent to the solution by the finite difference method.

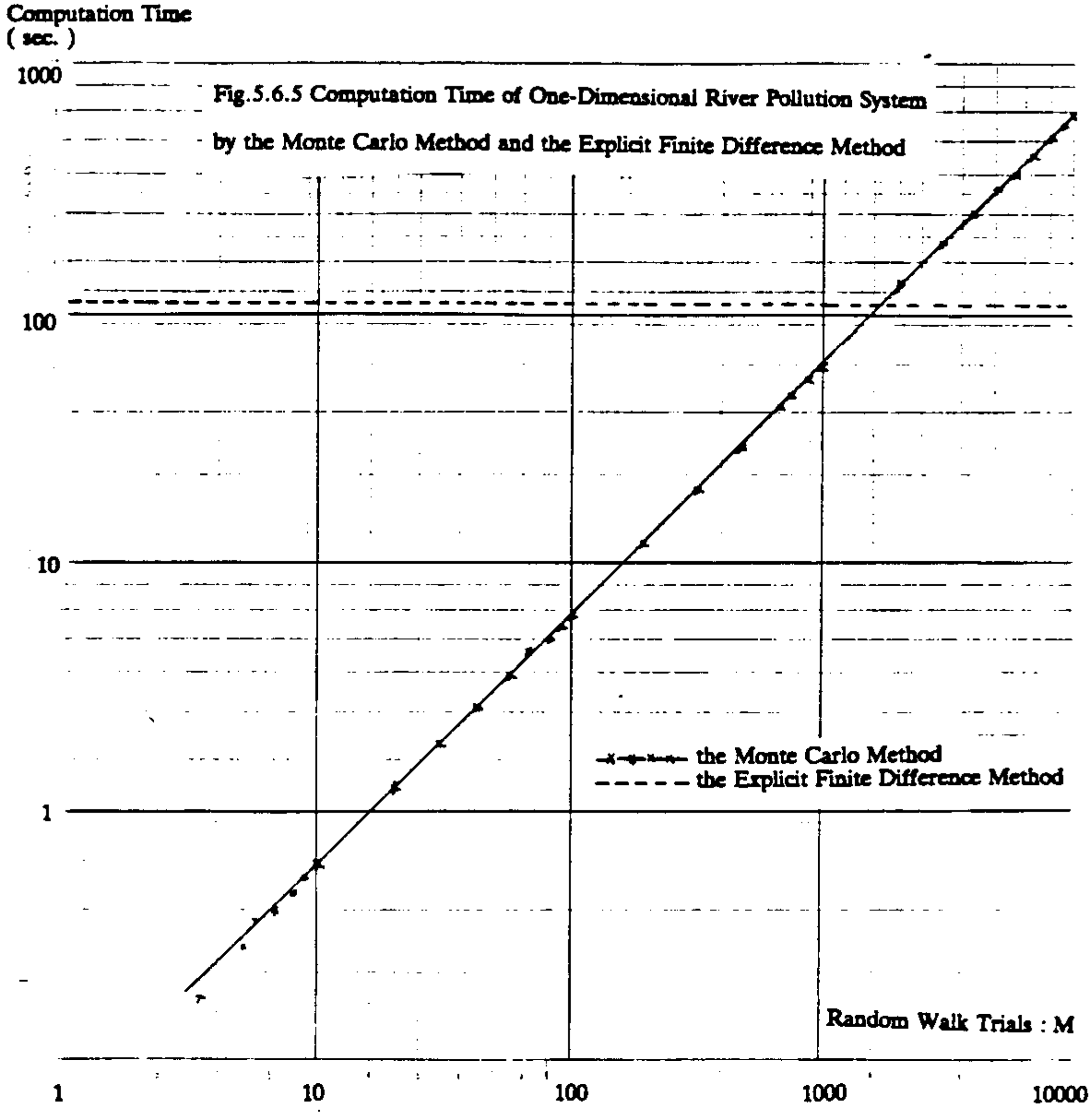


Fig.5.6.5 Computation Time of One-Dimensional River Pollution System by the Monte Carlo Method and the Explicit Finite Difference Method

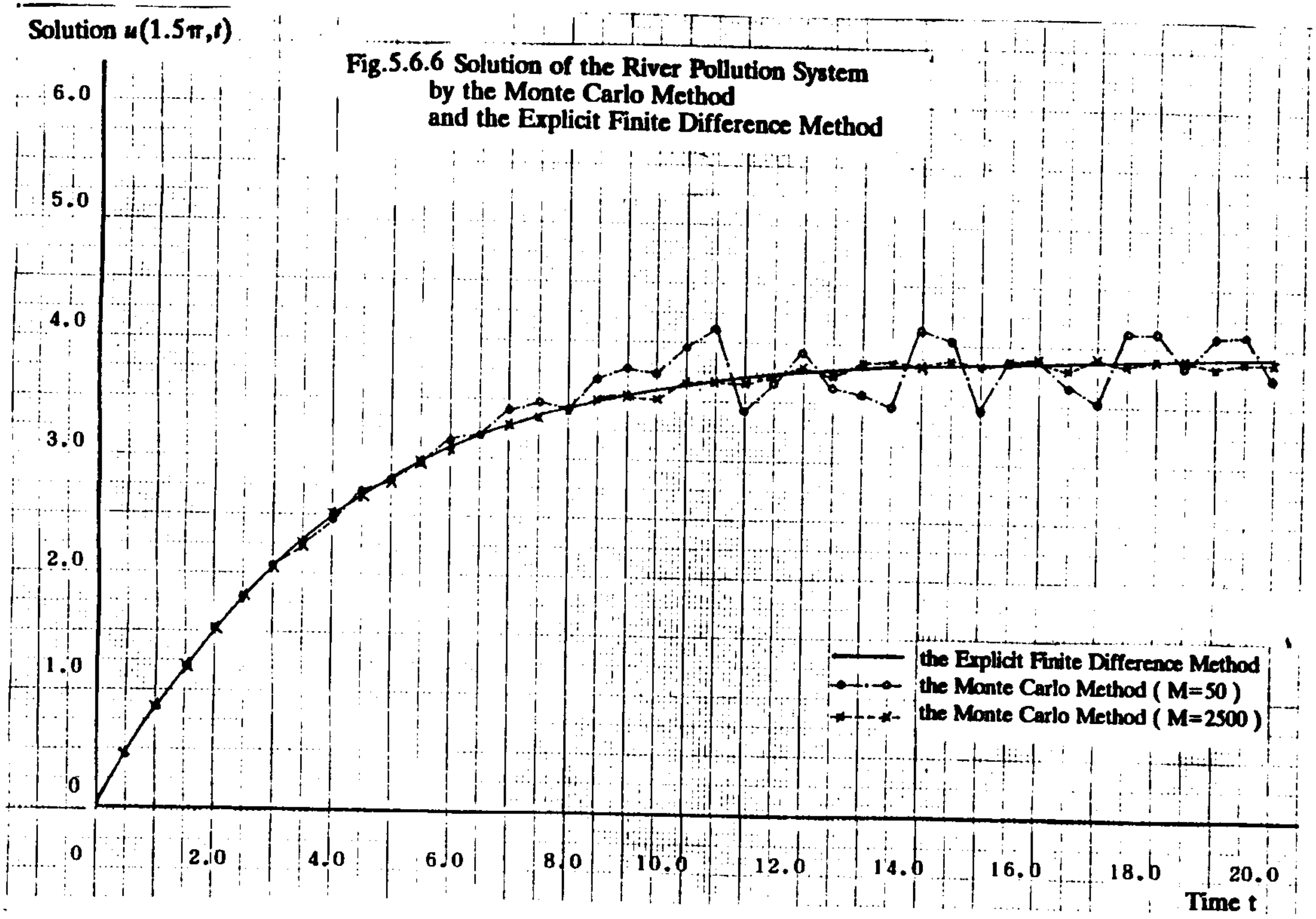


Fig.5.6.6 Solution of the River Pollution System by the Monte Carlo Method and the Explicit Finite Difference Method

CHAPTER 6

The Entire Identification Process

6.1 Introduction

This chapter organizes the entire identification procedures and combines the pattern recognition approach with possible conventional identification methods to determine the unknown pollution sources. First the pattern recognition approach discussed in Chapter 4 is employed to characterize an intermediate structure of pollution sources. Then various function identification or parameter identification methods discussed in Chapter 3 are applied.

6.2 Combination of Pattern Recognition and Conventional Methods

Using the pattern recognition approach, the mathematical structure of pollution source can be characterized. In addition, in order to identify the pollution sources more exactly, various conventional identification methods can be used to advantage. Therefore, the entire identification method can be a combination of the pattern recognition method and the conventional methods such as the least square method, the cross-validation, the Kalman filtering or the dynamic programming method etc.

The first step of a total identification of pollution sources is to decide whether the pollution sources can be regarded as a finite number of point sources such as

$$F(x,t)=\sum_{i=1}^{K}\delta(x-x_{i_i})f_i(t) \tag{6.2-1}$$

or distributed source type such as

$$F(x,t)=g(x)f(t). \tag{6.2-2}$$

This classification is based on on a priori knowledge of the actual situation such as :

1) pollutant area, namely, whether it is an industrial area or an agricultural area or a town area, etc.,

2) the number of suspected pollution sources,

3) predicted pollution pattern, i.e., daily pattern, weekly pattern, monthly pattern, seasonly pattern, etc.

Next, the pattern recognition approach is introduced to characterize the pollution sources by an intermediate structure. Suitable subclasses of pollution sources are then generated. For example, for distributed sources, time patterns with an explicit periodity of f(t) might be nominated. Alternatively, for point sources, the locations of pollution sources are nominated by considering the actual environment. Then features are extracted in order to enhance the similarities and differences between the original pollution sources $F_{ACT}(x,t)$ and the generated pollution subclasses $F_i(x,t)$. Next, the discriminant function d_i is formulated, and a decision is made as to which subclass of the generated pollution source is the most close to the original pollution source. In this way, the intermediate pollution structure is characterized. After characterizing the pollution source structure, the specific pollution sources are determined by specifying their functions or their parameters using suitable conventional identification methods. Finally, the evaluation of the entire identification procedure is carried out by implementing simulation using the mathematical model. The total identification procedure is indicated in Fig.6.1. In the following sections, the entire identification processes for each type of pollution source is discussed.

6.3 Entire Identification for Distributed Pollution Sources

In this section, it is assumed that the pollution sources to be identified can be regarded as the distributed type of pollution sources, expressed as

$$F(x,t) = g(x)f(t) \tag{6.3-1}$$

where both the space function g(x) and the time function f(t) are assumed to be unknown. As one basic approach to the total identification procedure, the time pattern function f(t) in explicit form is first determined while the space function g(x) is assigned any non-zero value ; then the space function g(x) in implicit form is found by the conventional identification methods.

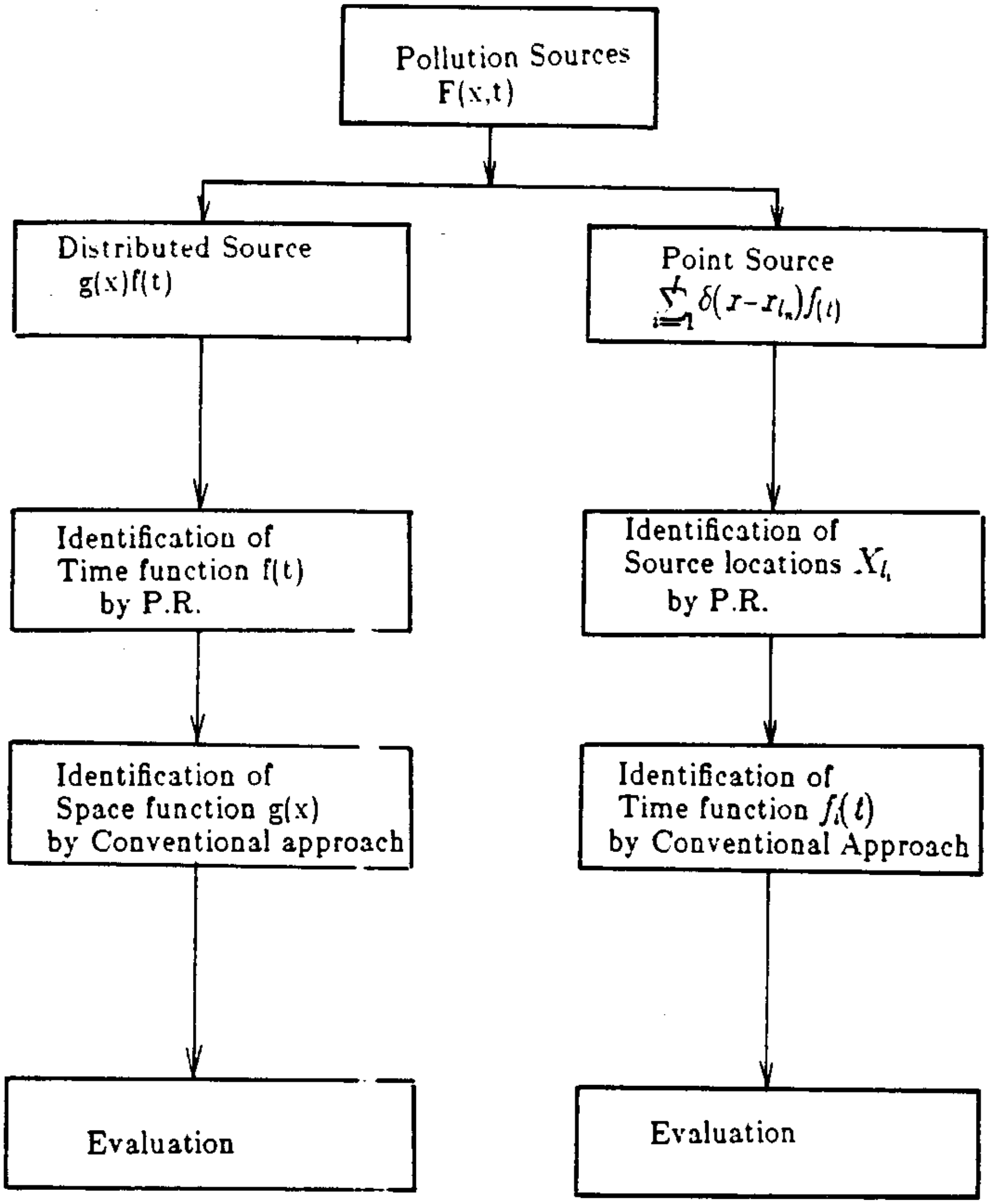


Fig.6.1 The Entire Identification Process

6.3.1 Case 1 : River Pollution System with BOD Measurement

Let us consider the river pollution system which is discussed in section 3.2 and characterized by the one-dimensional parabolic partial differential equation with BOD index $u(x,t)$

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + g(x)f(t) \tag{6.3-2}$$

where a and c are assumed to be constant, and b(x) is a function of x, the initial condition

$$u(x,0) = u_0(x) \tag{6.3-3}$$

and the boundary conditions

$$u(x_0,t) = u_{x_0}(t) \tag{6.3-4}$$

$$u(x_f,t) = u_{x_f}(t). \tag{6.3-5}$$

The BOD measurement data can be obtained from actual measuring stations at x_{m_j} with measurement noise

$$z(x_{m_j},t_k) = u(x_{m_j},t_k) + n(x_{m_j},t_k) \quad \text{for } j=1,..,M \tag{6.3-6}$$

where M is the number of measuring stations. Now the problem is to identify the unknown pollution source F(x,t)=g(x)f(t) from the BOD measurement data (6.3-6), using the mathematical model (6.3-2), the initial condition (6.3-3) and the boundary conditions (6.3-4) (6.3-5). The basic approach to this problem is to identify first the time function f(t) by the pattern recognition approach and then to identify the space function g(x) by conventional identification methods as presented in Fig.6.1.

The first step of the pattern recognition approach is to generate the subclasses S_i of the time pattern $f_i(t)$. Possible pollution time pattern functions are nominated

$$S_i : \quad f_i(t) = 1 + \sum_{n=1}^{i} \{A_n \cos n\omega t + B_n \sin n\omega t\} \tag{6.3-7}$$

or

$$= 1 + \sum_{n=1}^{i} \{a_n \cos(n\omega t + \phi_n)\} \quad \omega = \frac{2\pi}{T} \tag{6.3-8}$$

The basic period T of the time pattern is generated based on a priori knowledge, such as daily, weekly or seasonal patterns. If the pollution source can be regarded as having a daily time pattern, the basic period T is taken as 1 day. Therefore, the nominated pollution pattern can be expressed as a summation of multiple basic frequency components, $\frac{2n\pi}{T}$, as in Eqn.(6.3-7) or (6.3-8). It is noted that there is an important relationship between the S_i below :

$$S_1 \subset S_2 \subset \cdots \subset S_i \subset \cdots \subset S_I \tag{6.3-9}$$

Therefore as increasing the index "i" of subclass S_i, the generated time pattern f_i can close to the original time function f_{ACT}. Depending upon the generated time pattern, Eqn. (6.3-1) is solved numerically by the finite difference or the finite element method in conjunction with the Monte Carlo method. The calculated solutions $u_i(x_{m_j},t_k)$ which are equivalent to the measurement location data are stored as output data. In this case, the space function g(x) can be simply replaced by a non-zero value, because resulting feature vectors are not affected by g(x).

The term g(x) can be identified later by the conventional identification methods.

The second step is to select these features which can serve to enhance the similarities and differences between the original time function $f_{ACT}(t)$ and the generated subclass of the time pattern $f_i(t)$. As suggested in Section 4.4, the correlation function or the coherence function can be employed to advantage. Here, the normalized correlation function is used as an example. Before calculating the output $u_i(x_{m_j},t)$, the parameter values a_i and ϕ_i must be specified as initial starting values. Then these parameter values are optimized in order to maximize the features. This procedure is readily executed by conventional parameter optimization approaches, such as the steepest decent method, the Newton-Raphson method, the Gauss-Newton method, the conjugate direction method, etc. After the determination of the optimal parameter values a_i and ϕ_i the feature vector $\gamma_i(x_{m_j})$ based on the subclass S_i is obtained ~~as~~ follows :

$$\gamma_i = [c_i(x_{m_1}),c_i(x_{m_2}), \ldots ,c_i(x_{m_M}),1] \tag{6.3-10}$$

The next step is to calculate the discriminant function $d(\gamma_i)$. Here, we can employ the linear discriminant function which was introduced in Section 4.4 such as

$$d(\gamma_i) = W\gamma_i^T \tag{6.3-11}$$

where the weighting vector W is

$$W = \left[\frac{1}{M},\frac{1}{M}, \ldots ,\frac{1}{M},-e\right] \tag{6.3-12}$$

where M is the number of measurement stations. The weighting vector W is somewhat hueristic, but quite reasonable because the calculated feature elements are equally weighted, summed, and subtracted from the threshold values e. Thus, if

$$d(\gamma_i) > 0, \tag{6.3-13}$$

S_i is sufficient to describe the original time function $f_{ACT}(t)$, i.e. to determine the optimal time function $f_i(t)=f_{OPT}(t)\approx f_{ACT}(t)$. Otherwise, the other subclasses S_i for $(i \leftarrow i+1)$ are tried until Eqn. (6.3-13) is satisfied.

Once the time function has been identified by the pattern recognition approach, the next step is to determine the space function g(x). At this stage several conventional identification methods can be employed as discussed in Chapter 3. Here as an example, the least square method is employed to determine the nodal vector G of the space function of g(x). In order to formulate the least square method, a state vector equation which is derived from the original equation (6.3-1) by discretizing the space coordinate x into a finite number of the discrete points $j\Delta x$ for j=1,..,N, is derived as described in Section 3.6. The vector equation can be written as

$$U_{k+1} = AU_k + Gf_k \tag{6.3-14}$$

The output equation is

$$Y_k = CU_k \tag{6.3-15}$$

and the measurement equation is

$$Z_k = Y_k + n_k \tag{6.3-16}$$

where matrix A is the transition matrix determined by the field-parameters a,b(x),c. C is the output matrix determined by the measurement locations x_{m_i}. G represents unknown space function vectors to be identified

$$G = [g(0),g(\Delta x),g(2\Delta x),\ldots,g(N\Delta x)]^T \tag{6.3-17}$$

and $f_k = f_{opt}(t_k)$ is the optimal time function that was determined by the pattern recognition approach. The space function vector G can be determined by minimizing the cost function

$$J = \sum_{k=1}^{L}(Y_k - Z_k)^T(Y_k - Z_k) \tag{6.3-18}$$

As shown in Chapter 3, the least square methods can give the optimal value of G by calculating the following sequential expressions

$$G(n+1) = G(n) + K(n)[Z(n+1) - X(n+1)G(n)] \tag{6.3-19}$$

$$P(n+1) = P(n) - K(n)X(n+1)P(n) \tag{6.3-20}$$

$$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1} \tag{6.3-21}$$

with initial starting values $G(0)$ and P(0). As indicated in Eqn. (3.3-36),

$$X(n+1) = C(A^n f_0 + A^{n-1} f_1 + .. + If_n) \tag{6.3-22}$$

However, X(n) can alternatively be calculated by the state vector equations

$$S(n+1) = AS(n) + If(n) \tag{6.3-23}$$

$$X(n) = CS(n) \tag{6.3-24}$$

By calculating the sequential algorithm of the least square method from (6.3-19) through (C with initial starting values G(0) and P(0), the optimal space function value G can obtained. As a result, the entire identification procedure can be summarized in Fig.6.2.

6.3.2 Case 2 : Pollution Identification from DO Measurement Data

In this section, the entire identification procedure of a distributed pollution source identification using dissolved oxygen (DO) measurement data is derived. It is recognized that since DO measurement data have less measurement noise and since sampling and measurement are easier than in the case of the BOD, a higher accuracy of the identified function can be expected. On the other, the identification of pollution sources from DO measurement makes it necessary to calculate both the BOD and DO equations simultaneously since the pollution source F(x,t) affects the BOD index, but does not affect directly the DO index. The DO equation can be expressed as

$$\frac{\partial v}{\partial t} = a\frac{\partial^2 v}{\partial x^2} - b(x)\frac{\partial v}{\partial t} - K_r(v - v_s) - K_d u \tag{6.3-25}$$

with initial condition

$$v(x,0) = v_0(x) \tag{6.3-26}$$

and boundary conditions

$$v(x_0,t) = v_{x_0}(t) \tag{6.3-27}$$

$$v(x_f,t) = v_{x_f}(t) \tag{6.3-28}$$

The DO measurement data from the environment can be expressed as

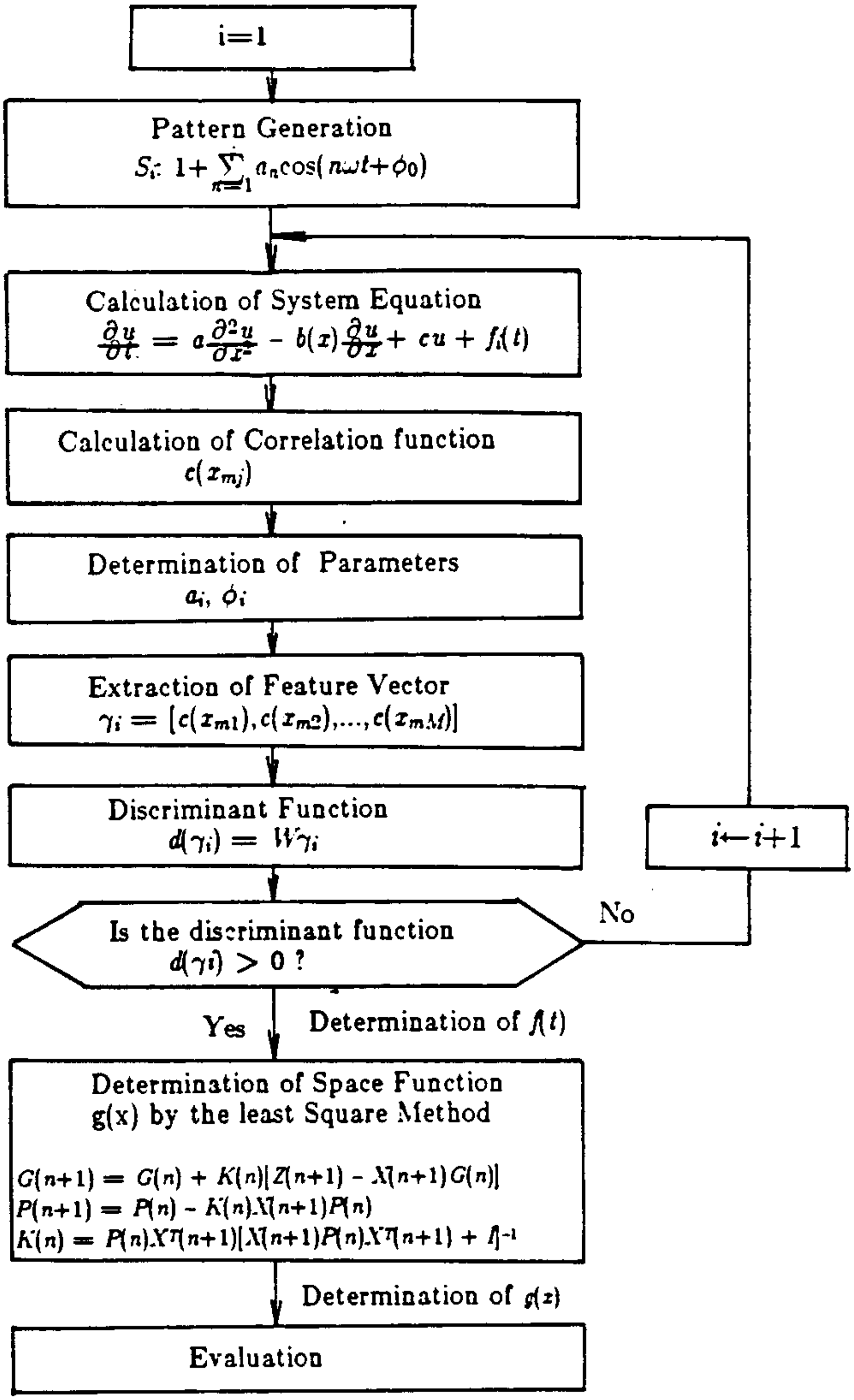

i=1
Pattern Generation
S_i: $1+\sum_{n=1} a_n\cos(n\omega t+\phi_0)$
Calculation of System Equation
$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} + cu + f_i(t)$
Calculation of Correlation function
$c(x_{mj})$
Determination of Parameters
a_i, ϕ_i
Extraction of Feature Vector
$\gamma_i = [c(x_{m1}), c(x_{m2}), \ldots, c(x_{mM})]$
Discriminant Function
$d(\gamma_i) = W\gamma_i$
$i \leftarrow i+1$
No
Is the discriminant function
$d(\gamma i) > 0$?
Yes
Determination of $f(t)$
Determination of Space Function
g(x) by the least Square Method
$G(n+1) = G(n) + K(n)[Z(n+1) - X(n+1)G(n)]$
$P(n+1) = P(n) - K(n)X(n+1)P(n)$
$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1}$
Determination of $g(x)$
Evaluation

Fig.6.2 Distributed Pollution Source Identification from BOD Data

$$w(x_{m_j},t_k) = v(x_{m_j},t_k) + \xi(x_{m_j},t_k) \qquad \text{for } j=1,\ldots,M \tag{6.3-29}$$

where M is the number of measurement stations. The problem is to identify the distributed pollution source F(x,t)=g(x)f(t) from the DO measurement data with the aid of the two mathematical models Eqn.(6.3-1) and Eqn. (6.3-25) along with their initial conditions and boundary conditions. The basic identification procedure is same as that for the BOD except that

1) the BOD equation (6.3-1) and the DO equations (6.3-25) must be solved simultaneously to calculate the $v(x_{m_j},t_k)$ equivalent to the measurement data at the pattern generation step, and

2) the state vector equation must contain both the BOD state vector as well as the DO state vector when applying the least square method.

The first step of the pattern recognition approach is to generate the subclass S_i of time patterns. Possible pollution time pattern functions $f_i(t)$ are nominated in the general form of Eqn.(6.3-7) or (6.3-8). Based on the generated time pattern $f_i(t)$ with g(x) taken as 1, the outputs $v_i(x_{m_j},t_k)$ are calculated and stored. Next the features (the correlation function or the coherence function) are selected. The feature vectors γ_i are formulated by optimizing the parameter values a_i and ϕ_i in Eqn. (6.3-8). Then the linear discriminant function $d_i(\gamma)$ is calculated. Thus, if $d_i(\gamma) > 0$, then S_i is sufficient to describe the original time function $f_{ACT}(t)$. Otherwise, another subclass S_i for $(i \leftarrow i+1)$ is tried until Eqn. (6.3-13) is satisfied.

Once the time function has been identified by the pattern recognition approach, it is necessary to determine the space function g(x). In order to formulate the least square method, a state vector equation, which includes both the BOD state and the DO state must be derived. Since the BOD state equation has already been derived in the proceeding section, only the DO state equation is formulated below. For simplicity, we define a new DO index $v(x,t)=v_s-v(x,t)$, so that Eqn. (6.3-25) can be replaced by

$$\frac{\partial v}{\partial t} = a\frac{\partial^2 v}{\partial x^2} - b(x)\frac{\partial v}{\partial x} - K_r v - K_d u \tag{6.3-30}$$

The new DO equation is discretized by the finite difference method so that

$$\frac{v(x,t+\Delta t) - v(x,t)}{\Delta t} \approx$$

$$a\frac{v(x+\Delta x,t) - 2v(x,t)+v(x-\Delta x,t)}{\Delta x^2}$$

$$- b(x)\frac{v(x,t)-v(x-\Delta x,t)}{\Delta x} - K_r v(x,t) + K_d u(x,t) \tag{6.3-31}$$

Or

$$v(x,t+\Delta t)\mathit{make} = p_1(x)v(x+\Delta x,t) + p_2(x)v(x,t)$$

$$+ p_3(x)v(x-\Delta x,t)+\Delta t\cdot K_d u(x,t) \tag{6.3-32}$$

For $x=i\Delta x$ and $t=k\Delta t$, Eqn. (6.3-32) can be written as

$$v_{k+1}(i) = p_1(i)v_1(i+1)+p_2(i)v(i)+p_3(i)v(i-1)+\Delta t K_d u(i) \tag{6.3-33}$$

Where the coefficients $p_1(i)$,$p_2(i)$ and $p_3(i)$ are defined as :

$$p_1(i) = \frac{a\Delta t}{\Delta x^2} \tag{6.3-34}$$

$$p_2(i) = \left\{1 - \frac{2a\Delta t}{\Delta x^2} - \frac{b(i)\Delta t}{\Delta x} - K_r\Delta t\right\} \tag{6.3-35}$$

$$p_3(i) = \left\{\frac{a\Delta t}{\Delta x^2} - \frac{b(i)\Delta t}{\Delta x}\right\} \tag{6.3-36}$$

Equation (6.3-33) can be expressed by the state vector equation as

$$V_{k+1} = AV_k + RU_k \tag{6.3-37}$$

where

$$V_k = [v_k(0), v_k(\Delta x), v_k(2\Delta x), \ldots, v_k(N\Delta x)]^T \tag{6.3-38}$$

$$A = \begin{bmatrix} p_2(0) & p_1(0) & & & & \\ p_3(1) & p_2(1) & p_1(1) & & & \\ & p_3(2) & p_2(2) & p_1(2) & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & p_3(N) & p_2(N) \end{bmatrix} \tag{6.3-39}$$

$$R = \begin{bmatrix} \Delta tK_d & & & & & \\ & \Delta tK_d & & & & \\ & & \Delta tK_d & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \Delta tK_d \end{bmatrix} \tag{6.3-40}$$

Therefore, the new DO equation and the BOD equation are calculated simultaneously as

$$U_{k+1} = AU_k + Gf_k \tag{6.3-41}$$

$$V_{k+1} = AV_k + RU_k \tag{6.3-42}$$

Then these two equations are merged into one vector equation

$$\begin{bmatrix} U_{k+1} \\ V_{k+1} \end{bmatrix} = \begin{bmatrix} A & O \\ R & A \end{bmatrix} \begin{bmatrix} U_k \\ V_k \end{bmatrix} + \begin{bmatrix} G \\ O \end{bmatrix} f_k \tag{6.3-43}$$

or

$$V'_{k+1} = A'V'_k + G'f_k \tag{6.3-44}$$

The DO measurement data can likewise be expressed as

$$W_k = \begin{bmatrix} O & C \end{bmatrix} \begin{bmatrix} U_k \\ V_k \end{bmatrix} + \xi_k \tag{6.3-45}$$

or

$$W_k = C'V'_k + \xi_k \tag{6.3-46}$$

The term X' which is referred to in Eqn.(6.3-23) can be expressed as

$$S'(n+1) = A'S(n) + If(n) \tag{6.3-47}$$

$$X'(n) = C'S(n) \tag{6.3-48}$$

Therefore, the formulation of the least square algorithm from DO measurement is the same as for BOD measurement i.e. :

$$G'(n+1) = G'(n) + K'(n)[W(n+1) - X'(n+1)G'(n)] \tag{6.3-49}$$

$$P'(n+1) = P'(n+1) - K'(n)X'(n+1)P'(n) \tag{6.3-50}$$

$$K'(n+1) = P'(n)X'^{T}[X'(n+1)P'(n)X'^{T}(n+1) + I]^{-1} \qquad (6.3\text{-}51)$$

with initial guessed values $G'(0)$ and $P'(0)$. By carrying out the sequential algorithm of the least square method from Eqn.(6.3-47) through (6.3-51) with the initial starting values, the optimal space function value G' can be obtained. The entire identification procedure is summarized in Fig.6.3.

6.3.3. Case 3 : Pollution Identification in Aquifer Pollution Systems

In this section, the entire identification procedure of aquifer pollution systems with a distributed pollution source is derived. The mathematical expression of a one-dimensional aquifer pollution systems can be characterized by the three equations, i.e., the ground water flow equation :

$$\frac{\partial}{\partial x}(K\frac{\partial h}{\partial x}) = S\frac{\partial h}{\partial t} + Q \qquad (6.3\text{-}52)$$

with initial conditions

$$h(x,t_{0)} = h_0(x) \qquad x \epsilon D \qquad (6.3\text{-}53)$$

and boundary conditions

$$h(x_0,t) = h_{x_0}(t) \qquad (6.3\text{-}54)$$

$$h(x_f,t) = h_{x_f}(t) \qquad x \epsilon \delta D \qquad (6.3\text{-}55)$$

where

h : hydraulic head

Q : volume flow rate : constant

K : hydraulic conductivity tensor : a function of x

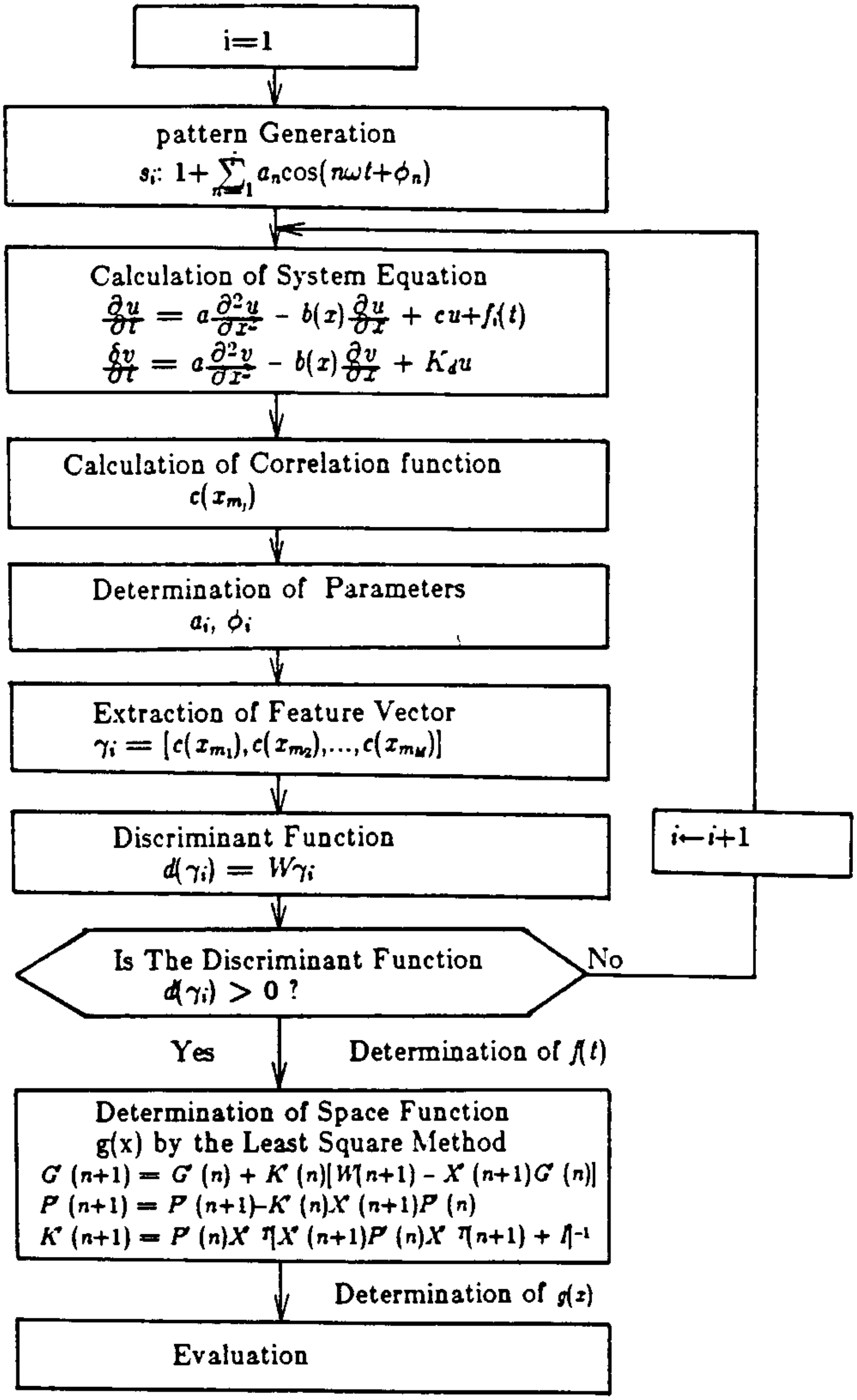

i=1
pattern Generation
s_i: $1+\sum_{n=1}^{r} a_n \cos(n\omega t+\phi_n)$
Calculation of System Equation
$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} + cu+f_i(t)$
$\frac{\delta v}{\partial t} = a\frac{\partial^2 v}{\partial x^2} - b(x)\frac{\partial v}{\partial x} + K_d u$
Calculation of Correlation function
$c(x_{m_j})$
Determination of Parameters
a_i, ϕ_i
Extraction of Feature Vector
$\gamma_i = [c(x_{m_1}), c(x_{m_2}), \ldots, c(x_{m_M})]$
Discriminant Function
$d(\gamma_i) = W\gamma_i$
$i \leftarrow i+1$
Is The Discriminant Function
$d(\gamma_i) > 0$?
No
Yes
Determination of $f(t)$
Determination of Space Function
g(x) by the Least Square Method
$G'(n+1) = G'(n) + K'(n)[W'(n+1) - X'(n+1)G'(n)]$
$P'(n+1) = P'(n+1) - K'(n)X'(n+1)P'(n)$
$K'(n+1) = P'(n)X'^T[X'(n+1)P'(n)X'^T(n+1) + I]^{-1}$
Determination of $g(x)$
Evaluation

Fig.6.3 Distributed Pollution Source Identification from DO Data

S : specific storage : constant

the dispersion equation

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}(D\frac{\partial c}{\partial x}) - \frac{\partial}{\partial x}(vc) - \frac{cQ}{n} + F(x,t) \qquad (6.3\text{-}56)$$

with initial conditions

$$c(x,0) = c_0(x) \quad x \epsilon D \qquad (6.3\text{-}57)$$

and boundary conditions

$$c(x_0,t) = c_{x_0}(t) \qquad (6.3\text{-}58)$$

$$c(x_f,t) = c_{x_f}(t) \quad x_0\, x_f\, \epsilon \delta D \qquad (6.3\text{-}59)$$

and the Darcy equation

$$v = \frac{K}{n}\frac{\partial h}{\partial x} \qquad (6.3\text{-}60)$$

where

c : pollutant concentration

D : coefficient of dispersion : constant

n : effective porosity (dimensionless)

F = g(x)f(t) : pollution source source or sink fluid

Observations of pollutant concentration are made directly at observation wells at a finite number of locations where the pollution concentrations can be observed at selected times in the presence of noise and error. Therefore, measured data can be expressed as :

$$z(x_{m_j},t_k) = c(x_{m_j},t_k) + e(x_{m_j},t_k) \tag{6.3-61}$$

for j=1,...,M where $e(x_{m_j},t_k)$ are measurement noise and error.

Now the identification problem is to determine the pollution sources F(x,t)=g(x)f(t) from measured data $z(x_{m_j},t_k)$ and the mathematical models. The basic procedure of the entire identification of aquifer pollution source is same in the case of the river pollution systêms except that

1) the ground water flow equation and the dispersion equation as well as Darcy's velocity equation must be solved to calculate the pollutant concentration $c(x_{m_j},t_k)$ equivalent to the measurement data at the step of pattern generation, and

2) the state vector equation must contain both the flow equation and dispersion equation in order to permit conventional identification.

Fist step of pattern recognition approach is to generate the subclasses S_i of time pattern. Possible pollution time pattern functions $f_i(t)$ are nominated in the general form (6.3-7) or (6.3-8). Based on the generated time pattern $f_i(t)$ and replacing the g(x) by 1, the outputs $c_i(x_{m_j},t_k)$ are calculated and stored. Next, the features using the correlation function or the coherence function are selected and the feature vectors γ_i are formulated by optimizing the parameter values a_i and ϕ_i. Then the linear discriminant function $d(\gamma_i)$. is calculated. Thus, if $d_i(\gamma) > 0$, S_i is sufficient describe the original time function $f_{ACT}(t)$. Otherwise, the other subclasses S_i for $(i{\leftarrow}i+1)$ ares tried until Eqn. (6.3-13) is satisfied. The final step is to determine the space function g(x). In order to formulate the least square method, a state vector equation is derived from the ground water flow equation (6.3-52) and the dispersion equation (6.3-56) as well as Darcy's equation (6.3-60). By discreterizing the space coordinate x into finite number of the discrete points $j\Delta x$, the aquifer flow equation is derived as :

$$h(x,t+\Delta t) \approx p_1(x)h(x+\Delta x,t) + p_2(x)h(x,t) + p_3(x)h(x-\Delta x,t) + \Delta tQ \tag{6.3-62}$$

where

$$p_1(x) = \frac{\Delta t}{S\Delta x^2}K(x+0.5\Delta x) \tag{6.3-63}$$

$$p_2(x) = 1 - \frac{\Delta t}{S\Delta x^2}\Big[K(x+0.5\Delta x) + K(x - 0.5\Delta x)\Big] \tag{6.3-64}$$

$$p_3(x) = \frac{\Delta t}{S\Delta x^2}K(x-0.5\Delta x) \tag{6.3-65}$$

The dispersion equation is,

$$\begin{aligned} c(x,t+\Delta t) \approx\ & q_1(x)c(x+\Delta x,t) + q_2(x)c(x,t) \\ & + q_3(x)c(x-\Delta x,t) + \Delta tg(x)f(t) \end{aligned} \tag{6.3-66}$$

where

$$q_1(x,t) = \frac{\Delta t}{\Delta x^2}D(x+0.5\Delta x) \tag{6.3-67}$$

$$\begin{aligned} q_2(x,t) = 1 & - \frac{\Delta t}{\Delta x^2}\Big[D(x+0.5\Delta x) + D(x - 0.5\Delta x)\Big] \\ & - \frac{\Delta t}{\Delta x}v(x-\Delta x,t) \end{aligned} \tag{6.3-68}$$

$$q_3(x,t) = \frac{\Delta t}{\Delta x^2}D(x-0.5\Delta x) + \frac{\Delta t}{\Delta x}v(x-\Delta x,t)$$

(6.3-69)

And Darcy's velocity is

$$v(x,t+\Delta t) = \frac{K}{n\Delta x}\Big[h(x+\Delta x,t) - h(x,t)\Big] \tag{6.3-70}$$

For $x=j\Delta x$ and $t=k\Delta t$, these equation can be merged into a state vector equation as

$$C_{k+1} = A_k C_k + G f_k \tag{6.3-71}$$

where

$$C_k = [c_k(0),c_k(\Delta x),\ldots,c_k(N\Delta x)]^T \tag{6.3-72}$$

$$G = [g(0),g(\Delta x),\ldots,g(N\Delta x)]^T \tag{6.3-73}$$

$$f_k = f(k\Delta t) \tag{6.3-74}$$

$$A(k) = \begin{bmatrix} B_{0,k} & C_{0,k} & & & & \\ A_{1,k} & B_{1,k} & C_{1,k} & & & \\ & A_{2,k} & B_{2,k} & C_{2,k} & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & & A_{N,k} & B_{N,k} \end{bmatrix} \tag{6.3-75}$$

where

$$A_{i,k} = q_3(h(i\Delta x,k\Delta t),i\Delta x,k\Delta t) \tag{6.3-76}$$

$$B_{i,k} = q_2(h(i\Delta x,k\Delta t),i\Delta x,k\Delta t) \tag{6.3-77}$$

$$C_{i,k} = q_1(h(i\Delta x,k\Delta t),i\Delta x,k\Delta t) \tag{6.3-78}$$

The output vector equation can be expressed as

$$Y_k = HC_k \tag{6.3-79}$$

The measurement vector equation can be written as

$$Z_k = Y_k + n_k \tag{6.3-80}$$

The space function vector G can be determined by minimizing the cost function J,

$$J = \sum_{k=1}^{L} (Y_k - Z_k)^T (Y_k - Z_k) \tag{6.3-81}$$

As derived in chapter 3, the least square method can give the optimal value of G by calculating the sequential equation

$$G(n+1) = G(n) + K(n)[Z(n+1) - X(n+1)G(n)] \tag{6.3-82}$$

$$P(n+1) = P(n) - K(n)X(n+1)P(n) \tag{6.3-83}$$

$$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1} \tag{6.3-84}$$

with initial starting values $G(0)$ and P(0). As indicated above, X(n+1) can be expressed as

$$S(n+1) = A(n)S(n) + If(n) \tag{6.3-85}$$

$$X(n+1) = HS(n+1) \tag{6.3-86}$$

By executing the sequential algorithm of the least square method from (6.3-81) through (6.3-85) with initial starting values, the optimal space function value G can obtained. The entire identification procedure is summarized in fig.6.4.

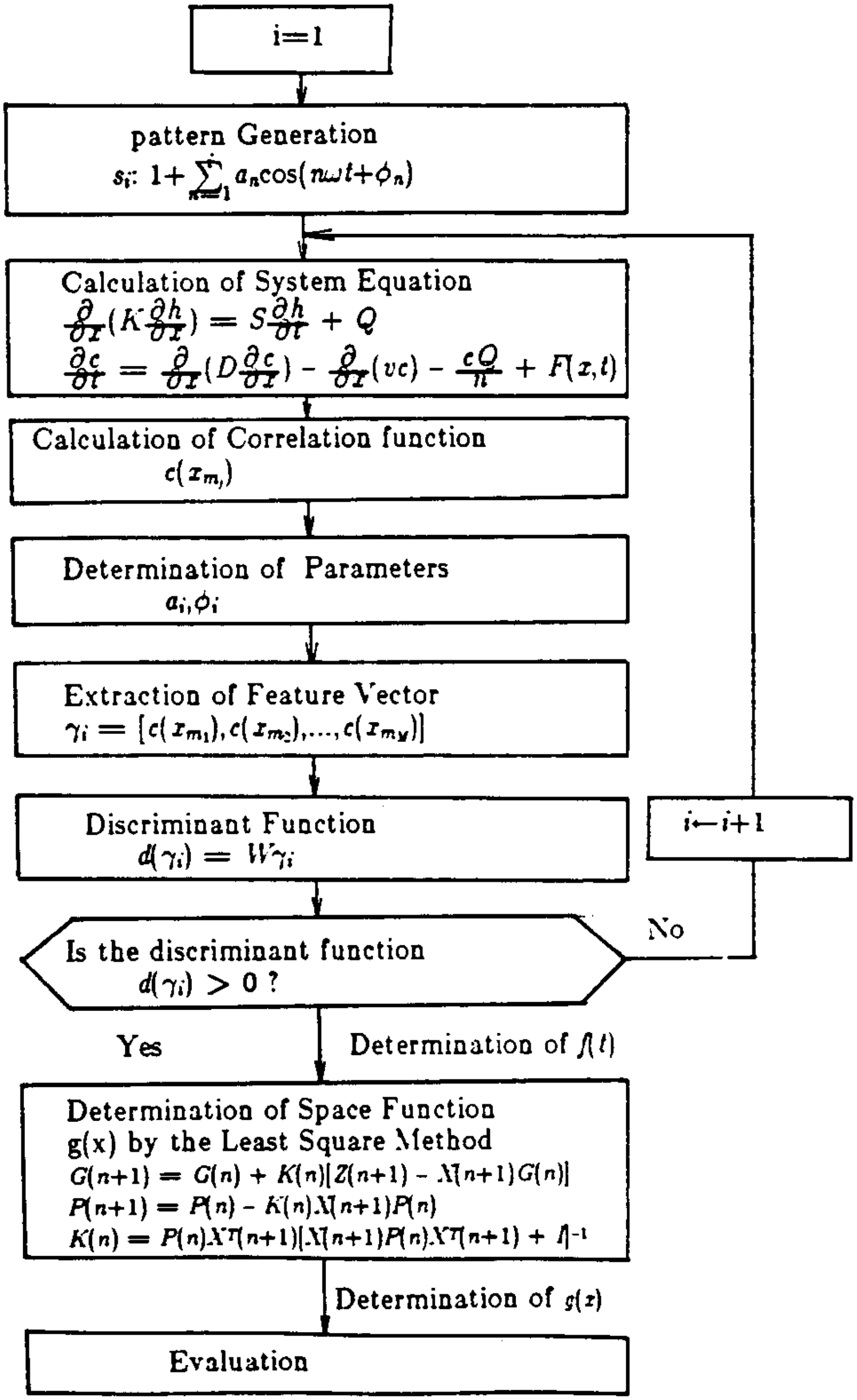

i=1
pattern Generation
s_i: $1+\sum_{n=1}^{r} a_n\cos(n\omega t+\phi_n)$
Calculation of System Equation
$\frac{\partial}{\partial x}(K\frac{\partial h}{\partial x}) = S\frac{\partial h}{\partial t} + Q$
$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}(D\frac{\partial c}{\partial x}) - \frac{\partial}{\partial x}(vc) - \frac{cQ}{n} + F(x,t)$
Calculation of Correlation function
$c(x_{m_i})$
Determination of Parameters
a_i,ϕ_i
Extraction of Feature Vector
$\gamma_i = [c(x_{m_1}),c(x_{m_2}),\ldots,c(x_{m_N})]$
Discriminant Function
$d(\gamma_i) = W\gamma_i$
$i \leftarrow i+1$
Is the discriminant function
$d(\gamma_i) > 0$?
No
Yes
Determination of $f(t)$
Determination of Space Function
g(x) by the Least Square Method
$G(n+1) = G(n) + K(n)[Z(n+1) - X(n+1)G(n)]$
$P(n+1) = P(n) - K(n)X(n+1)P(n)$
$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1}$
Determination of $g(x)$
Evaluation

Fig.6.4 Distributed Pollution Source Identification in Aquifer Pollution Systems

6.4 Entire Identification for Point Pollution Sources

In this section, we consider the case that the pollution sources to be identified are considered to b point type pollution sources which are expressed as

$$F(x,t) = \sum_{n=1}^{l} \delta(x-x_{l_n}) f_n(t) \tag{6.4-1}$$

where the pollution locations x_{l_i} and the time functions $f_i(t)$ are assumed to be unknown. As the basic approach of the total identification procedure, first, the pollution source locations x_{l_i} are determined by the pattern recognition approaches, then the time function $f_i(t)$ are determined by the conventional identification method, particularly, the dynamic programming method.

6.4.1 Case 4 : Point Source Identification from BOD measurements

Let us consider again the one-dimensional river pollution system with BOD index which is characterized by Eqn. (6.3-1) to (6.3-4). The identification problem is to determine the pollution source $F(x,t)$ which is expressed by Eqn.(6.4-1), (i.e., by their space and their time functions) from BOD data using the mathematical model Eqn.(6.3-1) to (6.3-4).

In the first step of the pattern recognition approaches, the pollutant area x (ϵ D) is classified into nominated pollution source locations x_{l_i} as subclasses S_i. At some of these nominated source locations x_{l_i}, the pollution sources actually may be injecting pollutants into the river ; at others locations there may be no actual sources. The purpose of the pattern recognition approach is to determine the actual pollution source locations x_{ACT} from the generated locations x_{l_i}. The mathematical expression of pollution sources along the river is

$$\begin{aligned} F(x_{l_i},t) &= \delta(x - x_{l_i})f_i(t) \\ &= f_i(t) \quad \text{for } i=1,\ldots,I \end{aligned} \tag{6.4-2}$$

where I is the number of nominated pollution sources. The next step is to select the features. Although there are several feature expressions, one of the most suitable possibilities is to use an accumulated value of the source intensity equivalent to the time function $f_i(t)$ as

$$\alpha_i = \int_{t_0}^{t_f} f_i(t)dt \tag{6.4-3}$$

or using a discrete time expression

$$\alpha_i = \sum_{k=0}^{L-1} f_i(t_k) \tag{6.4-4}$$

where $[t_0,t_f]$ is some measuring period and L is the number of measurement data. In fact, for all x_{l_i}, if $f_i(t) > 0$, x_{l_i} is an actual pollution location. If $f_i(t) = 0$ then x_{l_i} is not an actual pollution source location. However, the magnitude of the injected pollution source depends on the total pollution over some period $[t_0,t_f]$ rather than upon the pollution intensity at some instant of time. In another words, the responsible pollution sources can be determined by evaluating the integrated pollution time history $f_i(t)$ using expressed by Eqn. (6.4-3) or (6.4-4). Then α_i can be derived from the subclass of suspected location x_{l_i} and the measurement data $z(x_{m_j},t_k)$. Instead of the (6.3-1), the BOD vector equation which is equivalent to Eqn. (6.3-14) is employed as follows :

$$U_{k+1} = AU_k + BF_k \tag{6.4-5}$$

where

$$F_k = [f_1(t_k), f_2(t_k), \ldots f_i(t_k), \ldots f_I(t_k)]^T \tag{6.4-6}$$

and the matrix B is determined by the nominated pollution source location x_{l_i}. The output equation is

$$Y_k = CU_k \tag{6.4-7}$$

and the measurement equation is

$$Z_k = Y_k + n_k \tag{6.4-8}$$

Initially both sides of Eqn.(6.4-5) are summed from k=0 to L-1 so that

$$\sum_{k=0}^{L-1} U_{k+1} = \sum_{k=0}^{L-1} AU_k + \sum_{k=0}^{L-1} BF_k \tag{6.4-9}$$

or

$$\sum_{k=0}^{L-1} U_k + U_L - U_0 = A\sum_{k=0}^{L-1} U_k + B\sum_{k=0}^{L-1} F_k \tag{6.4-10}$$

This equation can be rewritten as

$$\sum_{k=0}^{L-1} U_k = (I - A)^{-1}(U_0 - U_L) + (I - A)^{-1}B\sum_{k=0}^{L-1} F_k \tag{6.4-11}$$

Similarly, the output equation (6.4-2) can be summed as

$$\sum_{k=0}^{L-1} Y_k = C\sum_{k=0}^{L-1} U_k \tag{6.4-12}$$

Substituting Eqn. (6.4-12) into Eqn.(6.4-11), we obtain

$$\sum_{k=0}^{L-1} Y_k = C(I - A)^{-1}(U_0 - U_L) + C(I - A)^{-1}B\sum_{k=0}^{L-1} F_k \tag{6.4-13}$$

$$= D + G\sum_{k=0}^{L-1} F_k \tag{6.4-14}$$

where we define

$$D = C(I - A)^{-1}(U_0 - U_L) \tag{6.4-15}$$

$$G = C(I - A)^{-1}B \tag{6.4-16}$$

Therefore, $\sum_{k=0}^{L-1} F_k$ can be written as :

$$\sum_{k=0}^{L-1} F_k = \begin{bmatrix} \sum_{k=0}^{L-1} f_1(t_k) \\ \sum_{k=0}^{L-1} f_2(t_k) \\ \cdots \\ \sum_{k=0}^{L-1} f_i(t_k) \\ \cdots \end{bmatrix} . \tag{6.4-17}$$

$$= [G^T G]^{-1} G^T [\sum_{k=0}^{L-1} Y_k - D] \tag{6.4-18}$$

By taking mean operation on Eqn. (6.4-8), we obtain

$$\frac{1}{L}\sum_{k=0}^{L-1} Z_k = \frac{1}{L}\sum_{k=0}^{L-1} Y_k + \frac{1}{L}\sum_{k=0}^{L-1} n_k \tag{6.4-19}$$

If the measurement noise n_k has zero mean, the second term of the right side can be eliminated.

$$\sum_{k=0}^{L-1} Z_k = \sum_{k=0}^{L-1} Y_k \tag{6.4-20}$$

Therefore, using Eqn.(6.4-20), Eqn.(6.4-18) can be expressed as

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdots \\ \alpha_i \\ \cdots \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{L-1} f_1(t_k) \\ \sum_{k=0}^{L-1} f_2(t_k) \\ \cdots \\ \sum_{k=0}^{L-1} f_i(t_k) \\ \cdots \end{bmatrix}. = [G^T G]^{-1} G^T [\sum_{k=0}^{L-1} Z_k - D] \tag{6.4-21}$$

Using Eqn.(6.4-21), the feature α_i can be calculated from the BOD measurement data. This kind of feature vector, based on the accumulated observation data, is particularly advantageous because

1) the measurement noise term can essentially be neglected,

2) the calculation of the feature vector by Eqn. (6.4-21) is very simple, and

3) the calculations are suitable for vector or parallel processors.

After calculating the feature vector, the discriminant function is formulated so as to make a decision whether or not the generated subclass of pollution locations is one of the actual source location. This decision making can be carried out by evaluating the linear discriminant function with some threshold value e because

1) the original partial differential equation is approximated by a finite difference methods so that the truncation error of the numerical calculation

of the the vector equation (6.4-5) must be compensated by some bias, and

2) measurement data from the actual environment are involved. The noise signal do not have an non-zero mean value, so that, the noise bias cannot be neglected.

Therefore, specifying a threshold value for the linear discriminant function d is quite reasonable. Then the decision making is carried out according to :

$$d(\gamma_i) = W\gamma_i^T \tag{6.4-22}$$

$$\gamma_i = [\alpha_i, -1] \tag{6.4-23}$$

$$W = [\frac{1}{L}, -e] \tag{6.4-24}$$

The feature vectors Eqn. (6.4-21) and the discriminant functions Eqn. (6.4-22) are calculated for all nominated pollution locations, and the actual pollution source locations are thereby extracted by the decision rule in Eqn. (4.6-5). Once the pollution locations are determined, their time patterns $f_i(t)$ are determined by conventional methods such as dynamic programming as follows :

Problem Statement

Given the state equation

$$U_{k+1} = AU_k + BF_k, \tag{6.4-25}$$

the output equation

$$Y_k = CU_k, \tag{6.4-26}$$

and the observation system

$$Z_k = CU_k + n_k, \tag{6.4-27}$$

find the optimal time functions $\hat{F}_k$ that minimize the cost function

$$J = (Z_L - CU_L)^T(Z_L - CU_L)$$

$$+ \sum_{i=1}^{L-1}\{(Z_k - CU_k)^T(Z_k - CU_k) + (F_k^T R F_k)\} \tag{6.4-28}$$

where R is weighting matrix. From the dynamic programming method, the following iterative equations concerning with the optimal control values $\hat{F}_k$ can be derived as indicated in Appendix B.

$$\hat{F}(L-k) = K(L-k)[Z(L+k-1) - CAU(L-k)] \tag{6.4-29}$$

$$K(L-k) = [R + B^T C^T P(k-1) CB]^{-1} C^T B^T P(k-1) \tag{6.4-30}$$

$$P(k) = P(k-1) - P(k-1)CB[R + B^T C^T P(k-1)CB]^{-1} B^T C^T P(k-1) \tag{6.4-31}$$

and

$$P(0) = I. \tag{6.4-32}$$

These equations can be calculated iteratively as indicated in Fig 3.5. First, the term P(1) is calculated using (6.4-31) with the initial condition (6.4-32). Then K(L-1) is calculated using (6.4-30) and P(1). The calculated K(L-1) is stored. For k=2, P(2) is calculated, then K(L-2) is stored, and so on. This calculation is continued until k=L. Next the optimal time function F_k is calculated using the stored K(L-k) and the state vector equation (6.4-25). The entire identification procedure is illustrated in Fig.6.5.

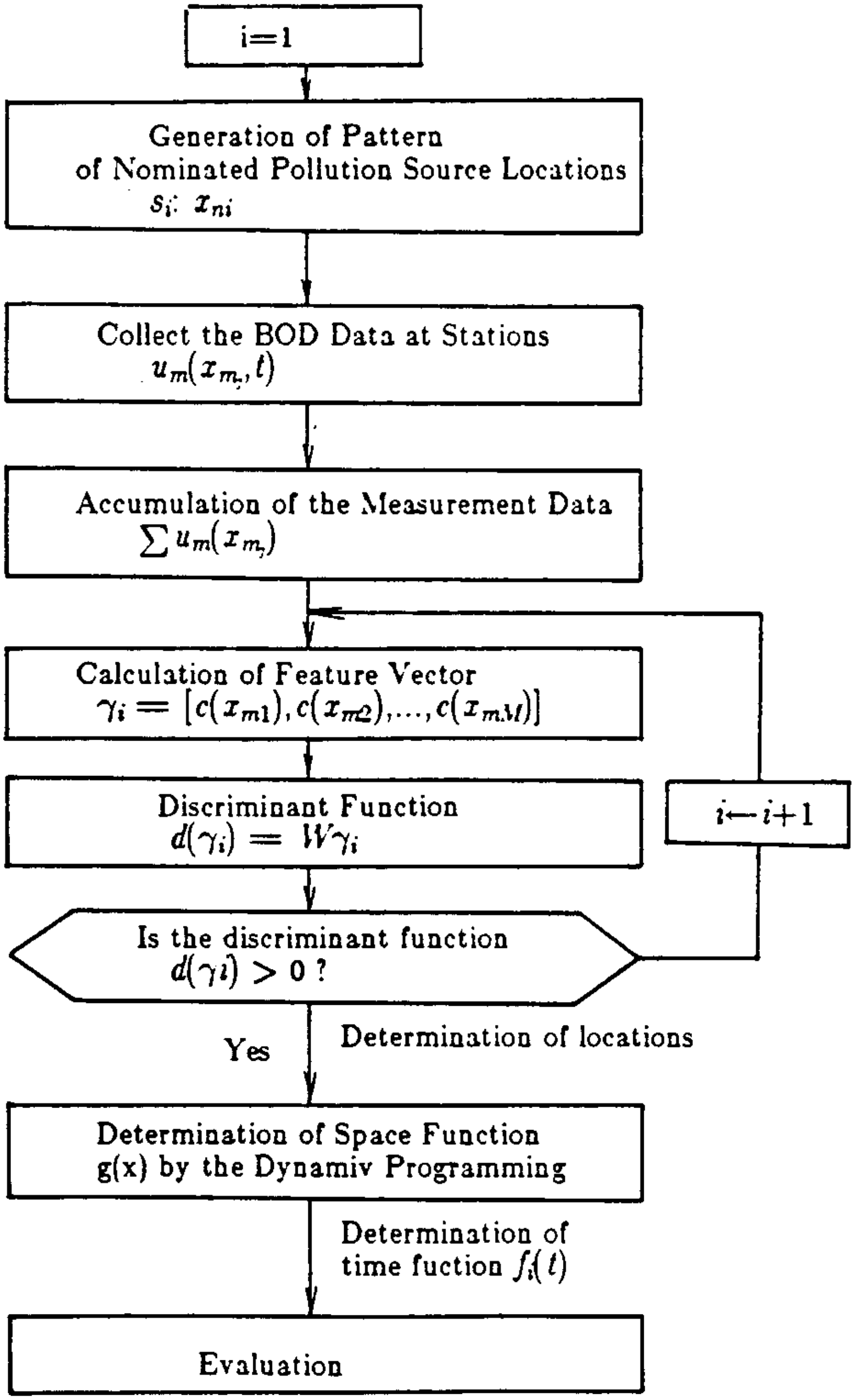

i=1
Generation of Pattern
of Nominated Pollution Source Locations
s_i: x_{ni}
Collect the BOD Data at Stations
$u_m(x_m, t)$
Accumulation of the Measurement Data
$\sum u_m(x_m)$
Calculation of Feature Vector
$\gamma_i = [c(x_{m1}), c(x_{m2}), \ldots, c(x_{mM})]$
Discriminant Function
$d(\gamma_i) = W\gamma_i$
$i \leftarrow i+1$
Is the discriminant function
$d(\gamma i) > 0$?
Yes
Determination of locations
Determination of Space Function
g(x) by the Dynamiv Programming
Determination of
time fuction $f_i(t)$
Evaluation

Fig.6.5 Point Pollution Source Identification from BOD Data

6.4.2. Case 5 : Point Source Identification from DO measurement

In this section, the entire identification of point sources from the DO measurement is formulated. The basic procedure of the identification is almost the same as in the case of BOD measurement data except that both the BOD and the DO indices must be included as state vectors. First the nominated pollution source locations x_{l_i} are generated as subclass S_i. Then the feature α_i is formulated as

$$\alpha_i = \sum_{k=1}^{L-1} f_i(t_k) \tag{6.4-31}$$

In order to formulate α_i, the state vector concerned with the BOD and DO indices are rewritten from Eqn. (6.3-43) as :

$$\begin{bmatrix} U_{k+1} \\ V_{k+1} \end{bmatrix} = \begin{bmatrix} A & O \\ R & A \end{bmatrix} \begin{bmatrix} U_k \\ V_k \end{bmatrix} + \begin{bmatrix} B \\ O \end{bmatrix} F_k \tag{6.4-32}$$

or simply,

$$V'_{k+1} = A'V'_{k+1} + B'F_k \tag{6.4-33}$$

where the F_k is same as in Eqn.(6.4-6) and B' is determined by x_{l_i}. The measurement equation is

$$W_k = C'V'_{k+1} + \xi_k \tag{6.4-34}$$

Using the same procedure from (6.4-9) through (6.4-21), the feature vector can be derived as

$$(\alpha_i) = \sum_{k=0}^{L-1} F_k$$

(6.4-35)

$$= [G'^T G']^{-1} G'^T [\sum_{k=0}^{L-1} W_k - D'] \tag{6.4-36}$$

where

$$G' = C'[I - A']^{-1} B' \tag{6.4-37}$$

$$D' = C'[I - A']^{-1}(V_0' - V_L') \tag{6.4-38}$$

After calculating the feature vector α_i, the discriminant function $d(\gamma_i)$ is formulated as in Eqn. (6.4-27). Then decision making is carried out to determine whether $d(\gamma_i) > 0$ or not. If so, the nominated location x_{l_i} can be regarded as one of the actual pollution source locations x_{ACT}. Otherwise the nominated location x_{l_i} is discarded. This process is tried for all of the nominated locations x_{l_i} for i=1,..I. In this way, the actual pollution source locations are all determined.

Once the pollution locations are determined, their time pattern $f_i(t)$ must be determined. This can be carried out by the dynamic programming method as introduced in the previous section as :

$$\hat{F}(L-k) = K(L-k)[W(L+k-1) - C'A'V'(L-k)] \tag{6.4-39}$$

$$K(L-k) = [R + B'^T C'^T P(k-1) C'B']^{-1} C'^T B'^T P(k-1) \tag{6.4-40}$$

$$P(k) = P(k-1) - P(k-1)C'B'[R + B'^T C'^T P(k-1)C'B']^{-1} B'^T C'^T P(k-1) \tag{6.4-41}$$

and

$$P(0) = I. \tag{6.4-42}$$

These equations can be calculated iteratively as presented in Fig.3.5. The entire identification procedure is illustrated in Fig.6.6.

6.4.3 Case 6 : Point Source Identification in Aquifer Pollution Systems

In this section, the entire identification of point pollution sources in aquifer pollution systems is formulated. The basic procedure is almost the same as for river pollution systems except that the state vector to formulate the features and the dynamic programming method contain the time varying transition matrix A_k.

Again let us consider the one-dimensional aquifer pollution system which is characterized by Eqn. (6.3-52) to (6.3-61). First the nominated pollution source locations x_{l_i} are generated as subclass S_i. Then the feature α_i is formulated such that

$$\alpha_i = \sum_{k=0}^{L-1} f_i(t_k) \tag{6.4-43}$$

In order to formulate α_i, the state vector concerned with the pollutant concentration Eqn. (6.4-5) is taken as :

$$C_{k+1} = A_k C_k + BF_k \tag{6.4-44}$$

The measurement equation from (6.3-42) is

$$Z_k = HC_k + \xi_k \tag{6.4-45}$$

Since this vector equation is a time varying, the direct formulation of $\sum_{k=0}^{L-1} F_k$ is impossible. However, by taking the average of the transition matrix A_k, the same

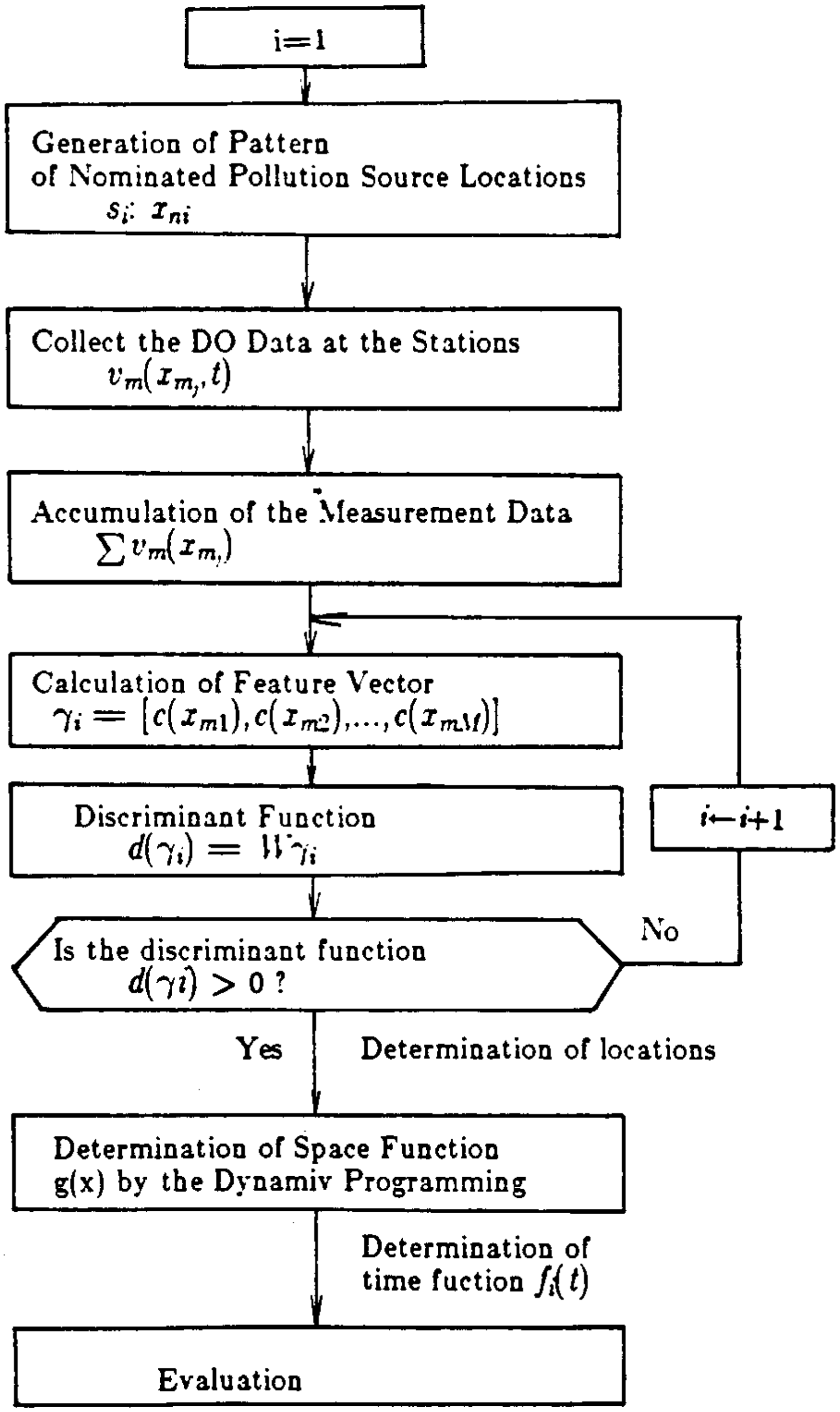

i=1
Generation of Pattern
of Nominated Pollution Source Locations
s_i: x_{ni}
Collect the DO Data at the Stations
$v_m(x_{m_j}, t)$
Accumulation of the Measurement Data
$\sum v_m(x_{m_j})$
Calculation of Feature Vector
$\gamma_i = [c(x_{m1}), c(x_{m2}), \ldots, c(x_{mM})]$
Discriminant Function
$d(\gamma_i) = W\gamma_i$
$i \leftarrow i+1$
Is the discriminant function
$d(\gamma i) > 0$?
No
Yes
Determination of locations
Determination of Space Function
g(x) by the Dynamiv Programming
Determination of
time fuction $f_i(t)$
Evaluation

Fig.6.6 Point Pollution Source Identification from DO Data

equation as in the proceeding section can be derived approximately as

$$\alpha_i = \sum_{k=0}^{L-1} F_k \tag{6.4-46}$$

$$= [G^T G]^{-1} G^T [\sum_{k=0}^{L-1} Z_k - D] \tag{6.4-47}$$

where

$$G = H[I - \bar{A}]^{-1} B \tag{6.4-48}$$

$$D = H[I - \bar{A}]^{-1} (C_0 - C_L) \tag{6.4-49}$$

where $\bar{A}$ represents the mean value of A_k. After calculating the feature, the discriminant function $d(\gamma_i)$ is formulated as in Eqn. (6.4-27). Then decision making is carried out to determine whether $d(\gamma_i) > 0$ or not. If so, the nominated locations x_{l_i} can be regarded as one of the actual pollution source locations x_{ACT}. Otherwise the nominated location x_{l_i} is discarded. This process is tried for all of the nominated location x_{l_i} for i=1,..I. Thereby, the actual pollution source locations are all determined. Then their time patterns $f_i(t)$ are determined by dynamic programming as introduced in the previous section as :

$$\hat{F}(L-k) = K(L-k)[Z(L+k-1) - HA(L-k)C(L=k)] \tag{6.4-50}$$

$$K(L-k) = [R + B^T H^T P(k-1) HB]^{-1} H^T B^T P(k-1) \tag{6.4-51}$$

$$P(k) = P(k-1) - P(k-1)HB[R + B^T H^T P(k-1) HB]^{-1} B^T H^T P(k-1) \tag{6.4-52}$$

and

$$P(0) = I. \tag{6.4-53}$$

These equations can be solved iteratively. The entire identification procedure is illustrated in Fig.6.7.

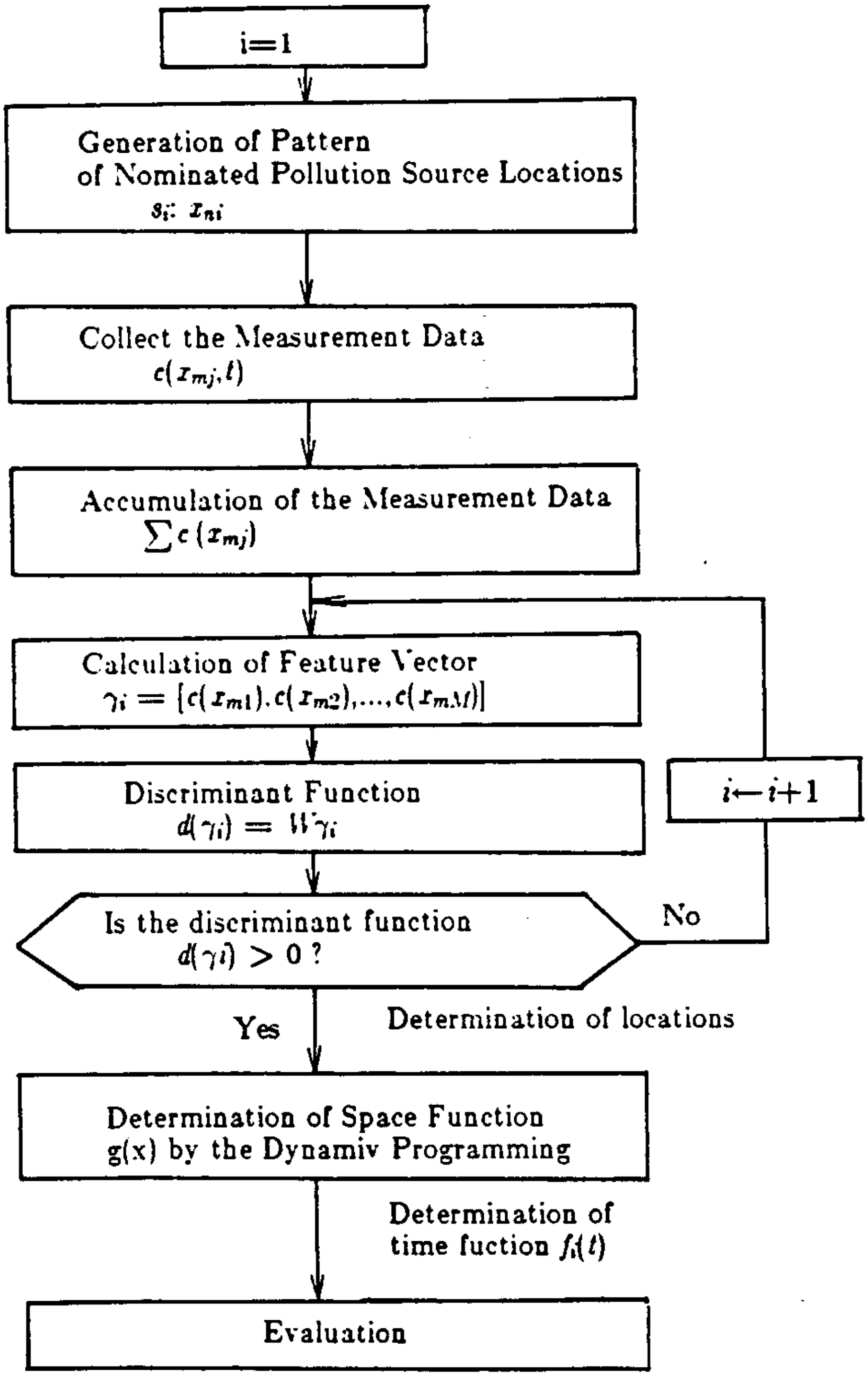

i=1
Generation of Pattern
of Nominated Pollution Source Locations
s_i: x_{ni}
Collect the Measurement Data
$c(x_{mj}, t)$
Accumulation of the Measurement Data
$\sum c\,(x_{mj})$
Calculation of Feature Vector
$\gamma_i = [c(x_{m1}), c(x_{m2}), \ldots, c(x_{mM})]$
Discriminant Function
$d(\gamma_i) = W\gamma_i$
$i \leftarrow i+1$
Is the discriminant function
$d(\gamma i) > 0$?
No
Yes
Determination of locations
Determination of Space Function
g(x) by the Dynamiv Programming
Determination of
time fuction $f_i(t)$
Evaluation

Fig.6.7 Point Pollution Source Identification in Aquifer Pollution Systems

CHAPTER 7

Computer Implementation Using A Peripheral Array Processor

7.1 Introduction

This chapter is devoted to a discussion of the numerical implementation for the entire proposed identification process including the pattern recognition approach as well as the conventional identification methods using high-speed array processors. Particular attention is focused upon implementations by peripheral array processors, which are connected to a general purpose computer and can perform high-speed computations at a modest cost. Several identification tasks are executed both using a conventional computer, the VAX-11/750 * and a combination of the VAX-11/750 and a peripheral array processor, the Mini-Map **. Performance evaluations of the host-plus-peripheral array processor used for source identification are presented.

7.2 Numerical implementation by peripheral array processors

The implementation of the identification process including the pattern recognition approach and the conventional identification methods employ highly

*VAX-11/750 is a trade mark of DEC Inc.

**Mini-Map is a trade mark of CSPI Inc.

computation intensive algorithms. High speed is needed for :

1) for the solution of the partial differential equations,

2) for signal processing, including the determination of the correlation and the coherence function in the course of feature selection and extraction,

3) for functional optimization or parameter identification using conventional identification methods.

It has long been recognized that conventional sequential computers are inefficient and expensive where relatively simple operations, such as additions and multiplications, need to be performed on very large sets of data elements. Recent development in computer architectures has led to special purpose processors which are basically different from conventional computers. So called supercomputers, or vector processors as well as peripheral array processors, belong to this category. Supercomputers, such as the STAR-100 and CRAY-1, were developed to solve partial differential equations. Such supercomputers have a very high speed, but they are very expensive. Peripheral array processors, on the other hand, have been primarily developed for signal processing applications. Modern peripheral array processors can provide very high data throughput at a modest cost [Karp77] [Karp80] [Karp81]. Array processors generally were designed as a peripheral device for conventional host computers and enhance the performance of the host computer in specific numerical computing tasks through extensive parallelism and/or pipelining. They include arithmetic sections containing at least one adder and one multiplier capable of operating in parallel. Peripheral array processors are very attractive for the implementation of full identification algorithms because they combine advantageous features of :

1) vector processors which excel in solving partial differential equations and

2) digital signal processors which process time series data by FFT or another signal processing algorithms.

Therefore, by using the peripheral array processors connected to a general purpose host computer, the implementation of our identification problems can be carried out at a higher speed than by conventional computers and at relatively modest cost.

7.3 The Peripheral Array Processor : Mini-Map

Over fifteen companies market peripheral array processors, each with unique architectures and facilities. The Mini-Map array processor developed by CSPI Inc. is a peripheral array processor designed to be used with a variety of host computers including the Digital Equipment Co. VAX series. The host computer controls the Mini-Map by providing the Mini-Map with program and data through the UNIBUS, by starting the operation of the Mini-Map, and by retrieving the results as presented in Fig.7.1. The host computer and the Mini-Map can run in parallel when suitably synchronized. The host computer and the Mini-Map can communicate with each other by means of a "shared memory window", a block of 8K bytes within the Mini-Map physical memory as presented in Fig.7.2, accessible to the host computer.

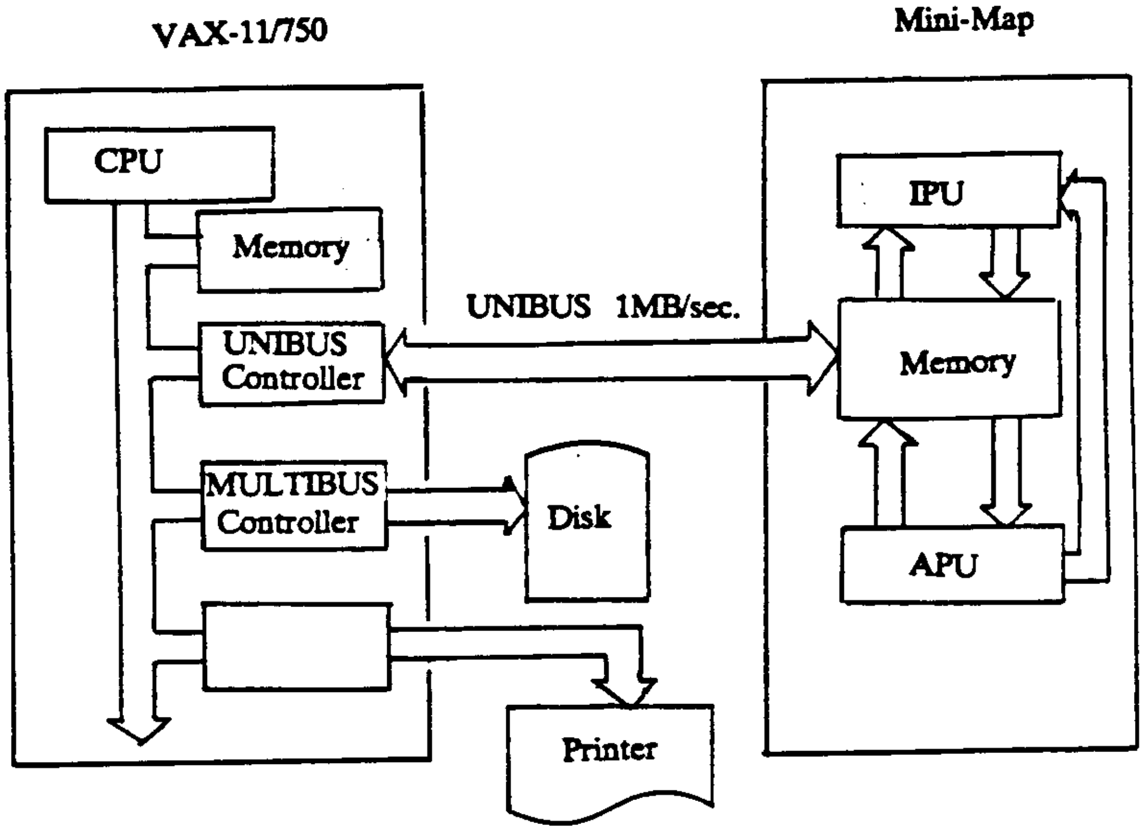


Fig.7.1 VAX-11/750 and Mini-Map Array Processor

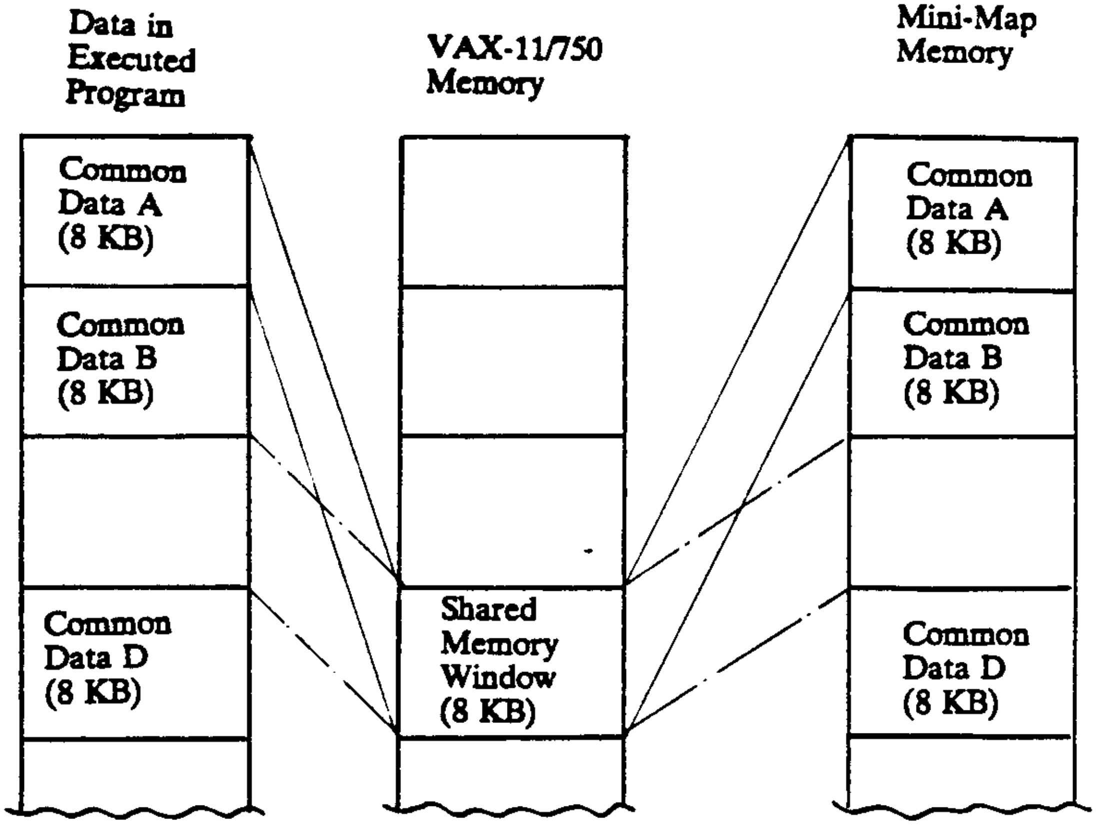


Fig.7.2 Data Communication Between VAX-11/750 and Mini-Map

The Mini-Map arithmetic unit consists of two processors : the Integer Processor Unit (IPU) and the Arithmetic Processor Unit (APU) as shown in Fig.7.3. The IPU performs the address calculations as well as data control by sending data to the Input Queue (IQ) for the APU, and by returning the results from the Output Queue (OQ) to the memory. The APU performs the arithmetic calculation on the data stream from the IQ and sends the results to the OQ. The IPU and APU run in parallel in synchronism. The internal architecture of the APU consists of a pipelined multiplier and a pipelined adder, running in parallel as presented in Fig.7.4. The architecture of the Mini-Map favors vector operations. The actual programming by microcode or assembly language is very difficult because of the complexity of the operations and their synchronization. For this reason, a very useful Scientific Subroutine Library (SSL), in which typical vector-matrix operations and many signal processing operations have been pre-programmed, is provided. By combining the SSL and user-written programs, the computational speed can be improved without an excessive programming effort.

7.4 Array Processors for Partial Differential Equations

Partial differential equations are usually solved numerically with the aid of finite differential methods or finite element methods. In the finite difference method, the original partial differential equation is approximated by a set of finite difference equations, one for each grid points. For example, in the river pollution system, the original equation (6.3-1) is approximated by the explicit finite difference equation.

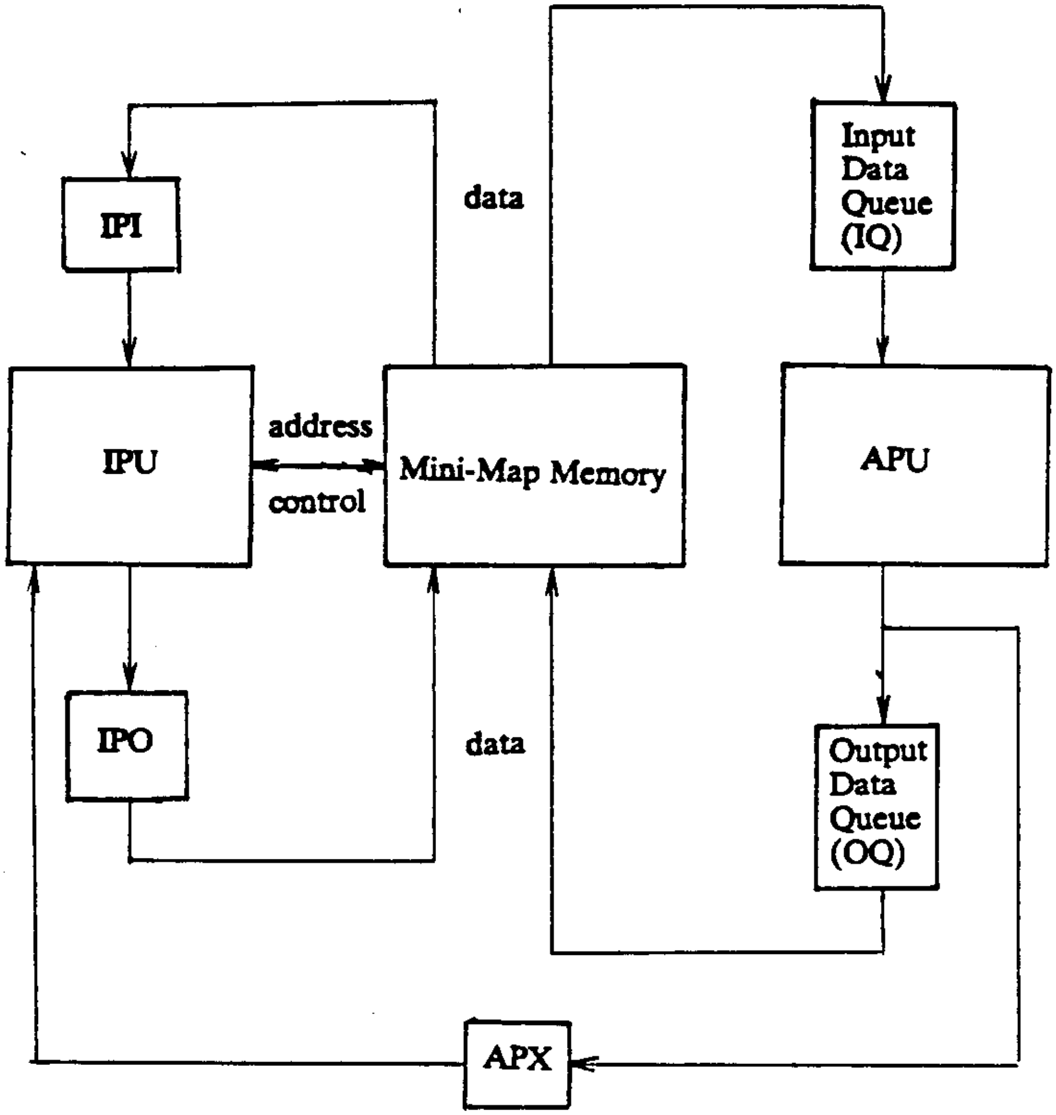


Fig.7.3 Data Flow Between the IPU,APU and Mini-Map Memory

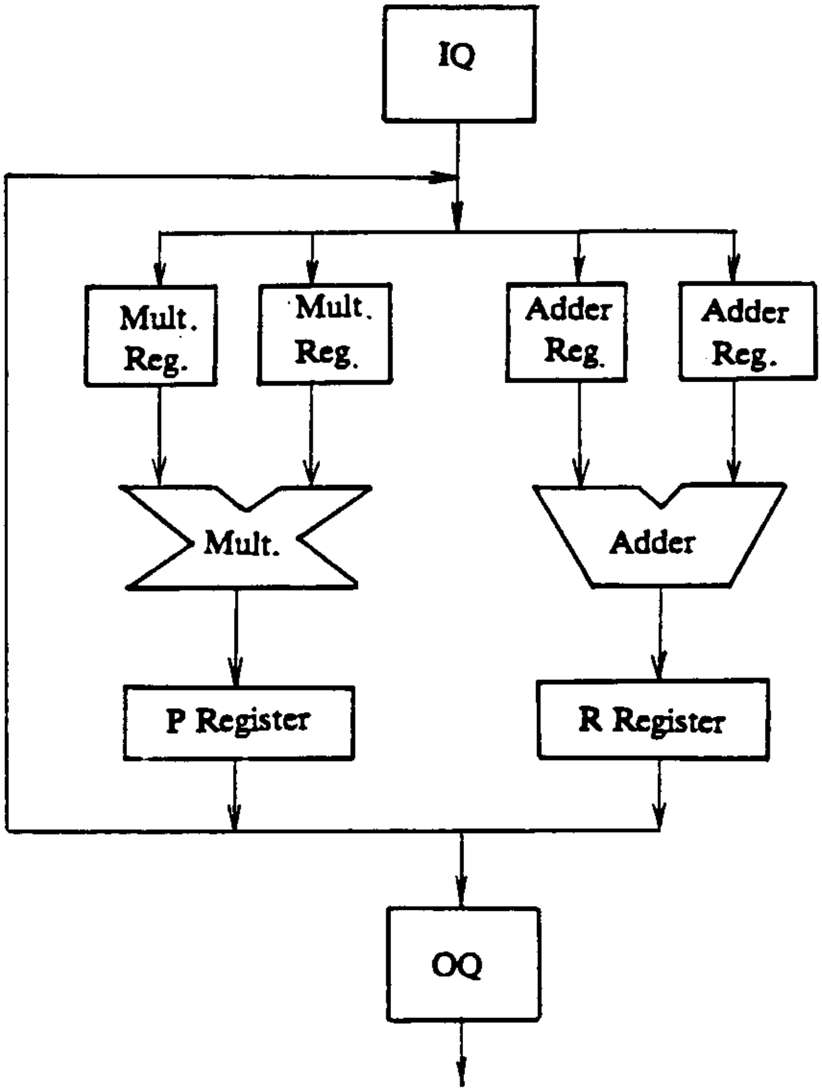


Fig.7.4 The APU and Queue of Mini-Map Array Processor

$$u(x,t+\Delta t) \approx A(x)u(x+\Delta x,t) + B(x)u(x,t) + C(x)u(x-\Delta x,t) + \Delta t g(x)f(t) \tag{7.7-1}$$

This algebraic system is solved easily and directly on the Mini-Map because it contains only vector multiplications and additions, where the length of the vector is equal to the number of internal grid points in the x domain. By contrast, the Mini-Map is not efficient for implicit schemes, such as

$$u(x,t-\Delta t) \approx A(x)u(x+\Delta x,t) + B(x)u(x,t) + C(x)u(x-\Delta x,t) + \Delta t g(x)f(t) \tag{7.7-2}$$

In this case a tridiagonal system of algebraic equations must be solved recursively at each time level, which requires considerable computation time. On the other hand, Eqn.(7.7-2) is computationally stable, while in Eqn.(7.7-1), Δt is constrained by the stability criterion.

7.5 Array Processors for Feature Vector Calculations

In the determination of the pattern recognition feature vectors, correlation or coherence functions are calculated for each generated subclass pattern. These calculations involve the FFT. Since array processors are particularly well suited for the calculation of the FFT, the feature vectors can be realized with a small programming effort. The Mini-Map has a specific subroutine for FFT computation, for the correlation and for convolution. By using these subroutines as well as user-created subroutines, feature vectors can be easily calculated.

7.6 Array Processors for Conventional Identification

The conventional identification methods which were discussed in Chapter 3 are employed to determine the specific parameter values or functional values of the pollution sources. These methods, such as the least square method, the Kalman filtering and the dynamic programming usually entail vector and matrix calculations. For example, the least square method can be expressed as

$$G(n+1) = G(n) + K(n)[Z(n+1)-X(n+1)G(n)] \tag{7.6-1}$$

$$P(n+1) = P(n) - K(n)X(n)P(n) \tag{7.6-2}$$

$$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1} \tag{7.6-3}$$

These equations contain only matrix and vector calculations ; therefore vector processing is efficient for these calculation, especially when the number of the discrete points and the measurement data are relatively large.

7.8 Numerical Implementation by Array processor

In this section, numerical implementations of the entire identification of pollution source are discussed. Iff particular, the main calculation of the entire identification procedure including : calculation of partial differential equations, calculation of feature vectors and calculation of conventional identification methods. The computational speed of the Mini-Map-VAX-11/750 combination is compared with the time required to solve the same problem using only the VAX-11/750.

7.8.1 Calculations of Partial Differential Equations

In this section, numerical solutions for partial differential equations including, the BOD equation, the DO equation and the aquifer quality equation, are calculated by the finite difference method.

BOD System Equation

The one dimensional BOD system equation is expressed as follows :

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + F(x,t) \tag{7.8-1}$$

Then explicit finite difference method is employed to obtain the finite difference equation :

$$u(x,t+\Delta t) = A(x)u(x+\Delta x,t) + B(x)u(x,t) + C(x)u(x-\Delta x,t) + \Delta t F(x,t) \tag{7.8-2}$$

where

$$A(x) = \frac{a\Delta t}{\Delta x^2} \tag{7.8-3}$$

$$B(x) = 1 - \frac{2a\Delta t}{\Delta x^2} - \frac{b(x)\Delta t}{\Delta x} - c\Delta t \tag{7.8-4}$$

$$C(x) = \frac{2a\Delta t}{\Delta x^2} + \frac{b(x)\Delta t}{\Delta x} \tag{7.8-5}$$

The following parameter value are assumed :

$$a = 0.5 \tag{7.8-6}$$

$$b(x) = 0.25(sin(x)+1.0) \tag{7.8-7}$$

$$c = 0.1 \tag{7.8-8}$$

$$F(x,t) = sin(x)+1.0 \tag{7.8-9}$$

with the initial condition

$$U(x,0) = 0 \tag{7.8-10}$$

and the boundary conditions

$$U(0,t) = 0 \tag{7.8-11}$$

$$U(x_N,t) = 0 \tag{7.8-12}$$

and simulation domain and step size,

$$\Delta x = 0.02\pi \tag{7.8-13}$$

$$\Delta t = 0.001 \tag{7.8-14}$$

$$x_N = N\Delta x \quad (N=5 - 500) \tag{7.8-15}$$

$$t_K = K\Delta t \quad (K=5 - 1000) \tag{7.8-15}$$

The implementation of the algorithm on the host/peripheral system involves the following steps:

1. Initialization : input data including the field parameters, input functions, initial condition and boundary conditions are read from the host computer into the array processor.

2. Calculation of the coefficients A(x), B(X), and C(x).

3. Calculation of U(x,t) in Eqn.(7.8-1).

4. Output: the solution values are transmitted from the Mini-Map to the VAX-11/750

For purpose of comparison, the entire algorithm was also implemented on the VAX-11/750. Computational Result 1 is presented in Table.7.8.1 which shows the computation time in seconds required by the Mini-Map and VAX-11/750 respectively using 100 steps in the time domain and a number of steps in the space domain ranging from 5 to 500. In Table.7.8.1, the number of space steps is maintained constant at 100 while the number of steps in the time domain is varied from 5 to 1000. The final column in last table presents the acceleration achieved by using the Mini-Map. The FORTRAN program for the MCL program for the Mini-Map are provided in the Appendix E. As can be seen, the computational time using a combination of the VAX-11/750-Mini-Map is about ten times faster than usinf only the VAX-11/750.

DO System Equation

The one dimensional system equation is expressed :

$$\frac{\partial v}{\partial t} = a\frac{\partial^2 v}{\partial x^2} - b(x)\frac{\partial v}{\partial x} - k_r(v-v_s) - k_d u \qquad (7.8\text{-}2\text{-}16)$$

By the explicit finite difference method, the following finite difference equation is obtained

$$v_{k+1} = A(x)v_k(x+\Delta x) + B(x)v_k(x) + C(x)v_k(x-\Delta x) - \Delta t K_d u(x) \qquad (7.8\text{-}17)$$

In order to calculate the DO index, both the DO equation and the BOD equation must be solved simultaneously. Therefore, the total number of difference equations becomes as twice large as in the case of the BOD index. The other numerical conditions for the entire calculations are the same as in the case of the

Table.7.8.1 Computational Evaluation of River Pollution System with BOD index by The Array Processor, Mini-Map

Computational Result 1

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
100	5	0.0321	0.0802	2.50
	10	0.0321	0.1250	3.89
	20	0.0321	0.2226	6.93
	30	0.0625	0.3164	5.06
	40	0.0625	0.3984	6.37
	50	0.0625	0.5000	8.00
	60	0.0703	0.6054	8.61
	70	0.0703	0.6835	9.72
	80	0.0703	0.7851	11.17
	90	0.0780	0.8671	11.12
	100	0.0780	0.9765	12.52
	200	0.1406	1.969	14.00
	300	0.1953	2.828	14.48
	400	0.2500	3.731	14.92
	500	0.3203	4.664	14.56

Computational Result 2

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
5	100	0.0079	0.0468	5.91
10		0.0102	0.0923	9.09
20		0.0217	0.1859	8.53
30		0.0281	0.3008	10.70
40		0.0312	0.3789	12.10
50		0.0391	0.5112	13.10
60		0.0539	0.6094	11.31
70		0.0645	0.6448	10.00
80		0.0703	0.7773	11.06
90		0.0800	0.9028	11.29
100		0.0780	0.9765	12.52
200		0.1678	2.0195	12.03
300		0.2578	2.9414	11.40
400		0.3516	3.8008	10.81
500		0.4258	4.7353	11.12
1000		0.8523	9.6230	11.30

BOD index. The computational results are presented in Table.7.8.2.

Aquifer Pollution System

The two dimensional aquifer pollution system is characterized by the following equations :

The aquifer flow equation is

$$S\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left(K_x(x,y)\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y(x,y)\frac{\partial h}{\partial y}\right) - Q \tag{8.4-1}$$

The aquifer quality system is

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D_x(x,y,t)\frac{\partial c}{\partial x}\right) + \left(D_y(x,y,t)\frac{\partial c}{\partial y}\right)$$

$$- \frac{\partial}{\partial x}\left(v_x(x,y,t)c\right) - \frac{\partial}{\partial y}\left(v_y(x,y,t)c\right) + \frac{cQ}{n} + F(x,y,t) \tag{8.4-2}$$

The Darcy equations are

$$v_x(x,y,t) = \frac{K_x(x,y)}{b}\frac{\partial h}{\partial x} \tag{8.4-3}$$

$$v_y(x,y,t) = \frac{K_y(x,y)}{b}\frac{\partial h}{\partial y} \tag{8.4-4}$$

The diffusion coefficients are

$$D_x(x,y,t) = e_x v_x(x,y,t) \tag{8.4-5}$$

$$D_y(x,y,t) = e_y v_y(x,y,t) \tag{8.4-6}$$

where e_x,e_y are dispersion coefficients. Assume the numerical values for these parameters

Table.7.8.2 Computational Evaluation of River Pollution System with DO index by The Array Processor, Mini-Map

Computational Result 3

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
100	5	0.0685	0.1718	2.51
	10	0.0703	0.2188	3.11
	20	0.0801	0.3906	4.88
	30	0.0898	0.6094	6.79
	40	0.1044	0.7969	7.63
	50	0.1172	0.9921	8.47
	60	0.1289	1.1641	9.03
	70	0.1406	1.4023	9.97
	80	0.1482	1.5664	11.14
	90	0.1602	1.7469	10.90
	100	0.1718	1.9414	11.30
	200	0.2813	4.2695	15.17
	300	0.3984	6.4609	16.21
	400	0.5117	8.7500	17.10
	500	0.6172	10.9571	17.75

Computational Result 4

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
5	100	0.0146	0.1016	5.91
10		0.0195	0.1992	9.09
20		0.0313	0.3398	8.53
30		0.0468	0.6016	10.70
40		0.0703	0.7852	12.10
50		0.0898	1.0117	13.10
60		0.1054	1.2073	11.31
70		0.1210	1.4180	10.00
80		0.1328	1.5859	11.06
90		0.1602	1.8203	11.29
100		0.1679	1.9414	12.52
200		0.3398	3.9336	12.03
300		0.5117	5.9726	11.40
400		0.6785	7.9179	10.81
500		0.8398	9.6230	11.45
1000		1.6875	19.725	11.68

$$K_x(x,y) = K_y(x,y) = 31000.0 - 2.25(x+y) \tag{8.4-7}$$

$$S = 0.015 \tag{8.4-8}$$

$$e_x = e_y = 50.0 \tag{8.4-9}$$

$$Q = 500.0 \tag{8.4-10}$$

and initial condition

$$h(x,y,0) = 200.0 \quad x,y \in D \tag{8.4-11}$$

$$c(x,y,0) = 0.0 \qquad x,y \in D \tag{8.4-12}$$

and boundary conditions

$$h(x_b,y_b,t) = 200.0 \quad x_b,y_b \in \partial D \tag{8.4-13}$$

$$c(x_b,y_b,t) = 0 \tag{8.4-14}$$

where the space region ,D, is

$$0 \leq x \leq 10000.0 \qquad 0 \leq y \leq 10000.0 \tag{8.7-15}$$

and time region, T, is

$$0 \leq t \leq 2000.0 \tag{8.4-16}$$

For numerical computations the following space step size and time step size are specified:

$$\Delta x = \Delta y = 500.0 \tag{8.4-17}$$

$$\Delta\, t = 20.0 \tag{8.4-18}$$

To solve these partial differential equations, the explicit finite difference approximation method is employed.

Table.7.8.4 Computational Evaluation of Aquifer Pollution System by Array Processor, Mini-Map

Computational Result 5

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
100	5x5	1.602	4.218	2.63
	6x6	1.930	6.406	3.32
	7x7	2.270	8.887	3.91
	8x8	2.609	11.242	4.31
	9x9	2.972	14.813	4.98
	10x10	3.332	18.140	5.44
	20x20	7.359	64.348	8.74
	30x30	12.141	127.01	10.46
	40x40	17.668	228.04	12.91

Computational Result 6

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
5	20x20	0.371	1.960	5.28
10		0.738	4.738	6.42
20		1.468	11.168	7.61
30		2.207	17.109	7.75
40		2.949	24.590	8.34
50		3.679	30.332	8.25
60		4.422	36.520	8.41
80		5.891	49.563	8.44
100		7.359	64.348	8.74
500		36.80	349.47	9.50

In this section, feature vectors are calculated by the array processor system using coherence function. Assume BOD measurement data $u_m(x_{m_i},t_k)$ in a river pollution system which were observed at four measuring stations. The sampling interval is ΔT, and the total number of the measurement data at each measuring station is k=N, so that the total observation time is $N\Delta t$. As presented in Section 4.4, the following steps are involved for the calculations of the coherence functions :

1) tapering by window function

2) FFT transform

3) normalization

4) ensemble averaging or frequency averaging

The above steps must be executed to calculate the power spectra density and cross-power of for all measurement data for the different measuring locations. Furthermore, the same procedure must be executed for the calculation BOD $u(x_{m_i},t_k)$, which is equivalent to the measurement stations generated by the subclassified pollution source P_j. As numerical data, the calculated BOD data in Sec.7.8.2 are employed. In order to evaluate the computational performance of the array processor system, the number of the measurement data N is changed from 16 to 1024 while the number of measuring stations is kept at 4. The computational results are presented in Table.7.8.5. As can be seen from the results, the array processor system is significantly faster in In particular, a speed up of more than twenty is realized for N greater than 60.

Table.7.8.5 Computational Evaluation of Feature Vector Calculations

by The Array Processor, Mini-Map

Computational Result 7

I (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
4	9-16	0.008	0.161	13.75
	17-32	0.015	0.277	18.47
	33-64	0.026	0.519	19.96
	65-128	0.050	1.078	21.56
	129-256	0.106	2.160	20.38
	253-512	0.206	4.839	23.49
	513-1024	0.416	10.801	25.96

Calculation of The Least Square Method

In this section, as one of the conventional method for the identification of space function, the least square method is implemented on the array processor system. The least square method is formulated in the following :

$$G(n+1) = G(n) + K(n)[Z(n+1) - X(n+1)G(n)] \tag{7.8-19}$$

$$P(n+1) = P(n) - K(n)X(n+1)P(n) \tag{7.8-20}$$

$$K(n) = P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1) + I]^{-1} \tag{7.8-21}$$

$$X(n+1) = C(A^n f_0 + A^{n-1} f_1 + .. + If_k) \tag{6.3-22}$$

or

$$S(n+1) = AS(n) + If(n) \tag{6.3-23}$$

$$X(n) = CS(n) \tag{6.3-24}$$

with the initial values

$$G(0) = 0.0 \tag{6.3-25}$$

$$P(0) = I \tag{6.3-26}$$

$$S(0) = O \tag{6.3-27}$$

where

P(n)	: N x N matrix,
K(n)	: M x M matrix,
A	: N x N matrix,
C	: M x N matrix,

S(n) : N x N matrix,

X(n) : M x N matrix,

I : N x N matrix,

G(n) : N x 1 vector,

Z(n) : M x 1 vector,

f(n) : scalar time function,

N : the number of the space step,

M : the number of the measurement stations,

These equations are calculated simultaneously until $t = \Delta t K$ to solve the space function G(n+1) with initial starting values $G(0)$ and P(0). For numerical implementation, the following data and parameter values are assumed :

$$N = 5-40 \tag{7.8-30}$$

$$K = 5-500 \tag{7.8-31}$$

$$M = 4 \tag{7.8-32}$$

$$G = [0,0,\ldots.,0] \tag{7.8-33}$$

$$P(n) = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & 1 \end{bmatrix} \tag{7.8-34}$$

The computational results are presented in Table.7.8.6. It is clear that array processor can execute the least square method about thirty times faster than the host computer. Therefore it can be inferred that numerical implementations by the array processor will achieve similar speed-ups for the other conventional

Table.7.8.6 Computational Evaluation of Conventional Identification by The Array Processor, Mini-Map

Computational Result 8

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
100	5	0.5780	4.2187	7.2987
	6	0.5780	5.2891	9.1506
	7	0.6875	6.7185	9.9724
	8	0.6875	8.9179	12.971
	9	0.8359	10.871	13.777
	10	0.7891	13.410	16.427
	20	1.6484	53.730	32.595
	30	3.1719	140.03	44.146
	40	6.2830	296.39	47.171

Computational Result 9

K (No.)	N (No.)	Mini-Map (sec.)	VAX-11/750 (sec.)	Speed ratio (VAX/Mini-Map)
5	20	0.0703	2.6328	37.451
10		0.1484	5.3008	35.719
20		0.3203	10.660	33.282
30		0.4844	16.141	33.321
40		0.6484	21.332	32.899
50		0.8203	26.699	32.548
60		0.9844	32.109	32.618
70		1.1563	37.481	32.414
80		1.3203	43.031	32.592
90		1.4766	48.339	32.732
100		1.6406	53.472	32.593
200		3.3125	107.45	32.593
300		4.9688	161.57	32.517
400		6.6625	215.92	32.411
500		9.0380	293.57	32.480

methods because these entail the same matrix-vector calculations.

CHAPTER 8

Simulation and Analysis

8.1 Introduction

In this chapter, several numerical examples are presented to evaluate the entire identification process formulated in the preceding chapters. The first several numerical examples involve distributed pollution sources in river pollution in which the pollution sources are identified from BOD and DO. The second set of examples deals with point pollution sources in both river pollution and aquifer pollution systems. The computational results are evaluated by studying the manner in which the proposed identification process is capable of identifying the original pollution sources in the presence of increasing levels of measurement noise.

8.2 Example 1

In this section, a distributed pollution source in a river pollution system is identified from BOD measurement data. The river under study is assumed to flow from an upstream lake through an industrial region and eventually into the sea. The pollutant concentration u(x,t) of BOD is given by:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + F(x,t) \tag{8.2-1}$$

where a is the dispersion coefficient (assumed constant), b(x) is the flow velocity (assumed to be a function of the width of the river) and, c is the decay coefficient (assumed constant). Pollutant sources, F(x,t) are located at unknown points along the banks of the river. BOD data are obtained by sampling the river water at four measuring locations. Assume the following numerical values for the parameters:

$$a=0.5 \tag{8.2-2}$$

$$b=0.25(\,sin(x)+1.0\,) \tag{8.2-3}$$

$$c=0.1 \tag{8.2-4}$$

and initial condition

$$u(x,0)=0 \qquad x_0 \le x \le x_f \tag{8.2-5}$$

and boundary conditions

$$u(x_0,t)=0 \tag{8.2-6}$$

$$u(x_f,t)=0 \tag{8.2-7}$$

where

$$0 \le x \le \pi\,, \qquad 0 \le t \le 25.6 \tag{8.2-8}$$

The measurement data of BOD are obtained at four stations within the region under consideration and involve measurement noise as follows :

$$u_m(x_{m_1},t) = u(x_{m_1},t)+e_1(t) \;\; at\; x_{m_1}=0.3\pi \tag{8.2-9}$$

$$u_m(x_{m_2},t) = u(x_{m_2},t)+e_2(t) \quad at\ x_{m_2}=0.5\pi \tag{8.2-10}$$

$$u_m(x_{m_3},t) = u(x_{m_3},t)+e_3(t) \quad at\ x_{m_3}=0.7\pi \tag{8.2-11}$$

$$u_m(x_{m_4},t) = u(x_{m_4},t)+e_4(t) \quad at\ x_{m_4}=0.9\pi \tag{8.2-12}$$

where $e_i(t)$ for i=1,4 are measurement noise.

Objective

To identify the distributed pollution sources F(x,t) from measurement data (8.2-9) to (8.2-12) as well as the mathematical model (8.2-1) to (8.2-10).

For numerical computations the following space step size and time step size are specified:

$$\Delta x = 0.1\pi \tag{8.2-13}$$

$$\Delta t = 0.1 \tag{8.2-14}$$

To solve the parabolic partial differential equation, the following two methods are employed separately and their pattern recognition results are compared with each other.

(1) Finite Difference method (explicit method).

(2) Monte Carlo method (50 random walks per time step).

Pattern of Pollution Sources F(x,t)

The distributed pollution sources F(x,t) are functions of time and space and can be generally expressed as :

$$F(x,t) = g(x)y(t) \qquad (8.2\text{-}15)$$

By taking advantage of a priori knowledge of pollution sources, that is, the locations of factories or their daily or monthly or yearly operation patterns, etc., explicit subclasses of F(x,t) can specified as the nominated pollution patterns. For example, the following five subclasses might be generated from a priori information :

$$P1 : F_1(x,t) = (\sin(\omega t) + 1.0)sin(x) \quad \omega = \frac{2\pi}{3.2} \qquad (8.2\text{-}16)$$

$$P2 : F_2(x,t) = \sin^2(\omega t)sin(x) \qquad (8.2\text{-}17)$$

$$P3 : F_3(x,t) = sin(x) \qquad (8.2\text{-}18)$$

$$P4 : F_4(x,t) = (TRI(\omega t))sin(x) \qquad (8.2\text{-}19)$$

$$P5 : F_5(x,t) = (sin(\omega t) + 1.0) \qquad (8.2\text{-}20)$$

where the function $TRI(\omega t)$ is a triangular function with a period 1/3.2 and a maximum value of 1.0 at t=1/6.4. The pattern recognition approach is used to determine which of these subclasses the pollution sources resemble most closely.

Feature Vector Extraction

In order to extract the feature of the subclasses, coherence functions $r_j(x_i,f)$ are employed as the elements of the feature vectors γ_j.

$$r_j^2(x_{m_i},f) = \frac{|\overline{C_j(x_{m_i},f)}|^2}{Bm(x_{m_i},f)B_j(x_{m_i},f)} \qquad 0 \le r_j(x_{m_i},f) \le 1.0 \qquad (8.2\text{-}21)$$

where

$\overline{Bm(x_{m_i},f)}$: power spectra density function of measurement data $u_m(x_{m_i},t)$

$\overline{B_j(x_{m_i},f)}$: power spectra density function of the solution $u_j(x_{m_i},t)$ based on the subclass $F_j(x,t)$

$\overline{C_j(x_{m_i},f)}$: cross spectra density function between $u_j(x_{m_i},t)$ and $u_m(x_{m_i},t)$

i (=4) : the number of the measurement station

j (=5) : the number of the subclass of $F_j(x,t)$

Note that if the measurement data $u_m(x_{m_i},t)$ are quite similar to the calculated output $u_j(x_{m_i},t)$ based on one of the subclasses $F_j(x,t)$, the coherence function $r_j(x_{m_i},f)$ approaches 1.0, while if they are quite different, the coherence function approaches 0. The feature vectors can be expressed as:

$$\gamma_j = [\ r_j^2(x_{m_1},f), r_j^2(x_{m_2},f), r_j^2(x_{m_3},f), r_j^2(x_{m_4},f)\ ] \tag{8.2-22}$$

Classification

The classification process involves making a decision as to which pollution pattern among the subclasses is most close to the actual pollution sources. A discriminant function based on the feature vector, together with a suitable threshold, is used to that end.

$$d(\gamma_j) = W\gamma_j^T \tag{8.2-23}$$

$$= \frac{1}{4}\sum_{i=1}^{4} r_j^2(x_{m_i},f) \tag{8.2-24}$$

where

$$W = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}] \tag{8.2-25}$$

Simulation Results

For simulation purposes, "measurement data" at the four BOD measurement locations were generated by solving Eqn.(8.2-2), using for F(x,t) which is identical to subclass P_1, Eqn.(8.2-2), with added Gaussian noise with zero mean and standard deviation $\sigma = 0.1$. These data are presented in Table 8.2.1.

Next these measurement data were transformed into frequency domain data $u_m(x_i,f)$, and their power spectra were calculated using conventional signal processing techniques involving:

1) tapering by window functions

2) Fourier transform by FFT

3) calibration for truncation

4) smoothing and averaging

The power spectra are presented in Table 8.2.2. Note that in each case there are only one or two frequency components with substantial amplitudes, while all other frequency components are negligible. The non-negligible frequency components therefore constitute useful feature vectors.

To test the classification procedure, solutions at the four measuring points based on each of the five subclasses of input functions, were calculated using both the finite difference method and the Monte Carlo method. The resulting solutions $u_j(x_{m_i},t)$ were transformed into the frequency domain using the FFT. Finally, feature vectors γ_j were calculated using Eqn.(8.2-21) and the discriminant functions according to Eqn.(8.2-22). These computational results are presented in Table 8.2.1 to 8.2.3.

It can be seen that the value of the discriminant function $d(gamma_j)$ based on the subclass P_1, which is the same as the input function of the measurement data, is almost 1.0 for both cases, while the value of the discriminant function for subclasses P_2, P_3, and P_4 is very small, especially for f = 0.313. It can be concluded that the feature vector can extract the original input function by considering the values of the discriminant function even when measurement data are corrupted by measurement noise. However, the discriminant function based on subclass P_5 is also nearly equal to 1.0, even though P_1 contains the space functions sin(x), while P_5 does not. Additional processing is therefore required to resolve this conflict. As discussed in Chapter 6, this conflict can be resolved by identifying the space function g(x) using the conventional identification after having identified the time function f(t).

Fig.8.2.1 shows the discriminant function of the pattern subclass P_1 under different noise levels. From this result, it is recognized that the proposal pattern recognition method using the coherence function as the feature vectors is consistent even in the presence of a high level of noise.

Table 8.2.1 Measurement Data of BOD

Time	$Um(x_1,t)$	$Um(x_2,t)$	$Um(x_3,t)$	$Um(x_4,t)$
.00	.00000	.00000	.00000	.00000
.10	-.06837	.14044	.01681	.18805
.20	.12984	.27414	.13714	.06653
.30	.24895	.17292	.21014	.06199
.40	.23116	.43267	.31748	-.00542
.50	.55846	.42695	.15273	.06560
.60	.58433	.66985	.56732	-.01199
.70	.71495	.81777	.48912	.19413
.80	.46224	.80098	.63335	.22780
.90	.54207	.87090	.57039	.23134
.00	.66104	.85548	.87939	.33950
1.10	.82871	.93568	.91973	.35089
1.20	1.08080	.85653	.85739	.35456
1.30	.71317	1.10326	1.08942	.38330
1.40	.83123	1.24830	.95779	.27574
1.50	.92521	1.20580	1.18857	.50928
1.60	.78763	1.40927	1.04599	.34653
1.70	.85815	1.37364	1.13288	.55769
1.80	1.00183	1.38885	1.16371	.58895
1.90	1.24178	1.43656	1.23719	.38548
2.00	1.09468	1.52861	1.48446	.34686
2.10	1.17943	1.80422	1.32773	.50419
2.20	1.26096	1.55279	1.39403	.52162
2.30	1.24369	1.70517	1.47923	.55274
2.40	1.24621	1.76079	1.65599	.82600
2.50	1.44435	1.59432	1.69279	.65015
2.60	1.45212	1.91876	1.53748	.69082
2.70	1.55319	1.93545	1.74557	.70211
2.80	1.33082	2.09776	1.87607	.79277
2.90	1.37192	2.16524	1.77545	.66055
3.00	1.43067	2.27401	1.97905	.72727
3.10	1.48419	2.12177	1.83952	.83706
3.20	1.62831	2.16612	2.02369	.78590

Table 8.2.2 Power Spectra Density of Measurement Data

Frequ	$Bm(x_1,f)$	$Bm(x_2,f)$	$Bm(x_3,f)$	$Bm(x_4,f)$
.000	.2242e+01	.4450e+01	.3506e+01	.5591e+00
.313	.1282e+00	.2160e+00	.2002e+00	.3713e-01
.625	.1193e-01	.2129e-01	.3034e-01	.8880e-02
.938	.2608e-02	.2161e-01	.8339e-02	.6638e-03
1.250	.3161e-02	.1296e-01	.5270e-02	.2746e-02
1.563	.1050e-01	.2534e-03	.3381e-02	.1080e-02
1.875	.2841e-02	.1758e-02	.2236e-02	.1778e-03
2.188	.4426e-03	.8582e-03	.4317e-04	.5259e-03
2.500	.4607e-02	.5166e-03	.4196e-02	.3056e-03
2.813	.4641e-03	.2058e-02	.5061e-02	.1170e-02
3.125	.2970e-02	.4633e-03	.1024e-02	.1135e-02
3.438	.1138e-02	.6937e-03	.3147e-03	.3880e-03
3.750	.3895e-03	.2554e-02	.4389e-03	.7977e-03
4.063	.5472e-02	.1180e-02	.3487e-02	.2962e-03
4.375	.2354e-02	.1972e-02	.1747e-02	.6448e-03
4.688	.3640e-03	.3083e-02	.4828e-02	.9588e-03
5.000	.9049e-05	.2263e-02	.4113e-02	.8220e-04

Table.8.2.3 Feature Vectors and Discriminant Function

1) Soulution by Finite Difference method (explicit method)

subclass	f	$r^2(x_1,f)$	$r^2(x_2,f)$	$r^2(x_3,f)$	$r^2(x_4,f)$	$d(\gamma_i)$
P_1	.0	.9997	.9999	.9999	.9998	.9999
	.313	.9929	.9976	.9929	.9751	.9896
P_2	.0	.5866	.5794	.5744	.5743	.5711
	.313	.5613	.5561	.5613	.5610	.5599
P_3	.0	.6950	.6854	.6864	.6852	.6880
	.313	.5056	.5124	.5188	.5162	.5140
P_4	.0	.5709	.5658	.5654	.5640	.5665
	.313	.0791	.0822	.0718	.0707	.0759
P_5	.0	.9997	.9998	.9998	.9969	.9990
	.313	.9929	.9976	.9976	.9710	.9891

2) Solution by Monte Carlo method

subclass	f	$r^2(x_1,f)$	$r^2(x_2,f)$	$r^2(x_3,f)$	$r^2(x_4,f)$	$d(\gamma_i)$
P_1	.0	.9965	.9980	.9984	.9942	.9968
	.313	.9654	.9905	.9587	.8451	.9399
P_2	.0	.4830	.4859	.4886	.4878	.4863
	.313	.4542	.4742	.4746	.4640	.4725
P_3	.0	.6813	.6791	.6759	.6742	.6776
	.313	.3519	.4557	.4137	.4090	.4076
P_4	.0	.6562	.6765	.6811	.6816	.6738
	.313	.0584	.1192	.0973	.0926	.0918
P_5	.0	.9965	.9972	.9989	.9968	.9973
	.313	.9625	.9914	.9597	.8704	.9460

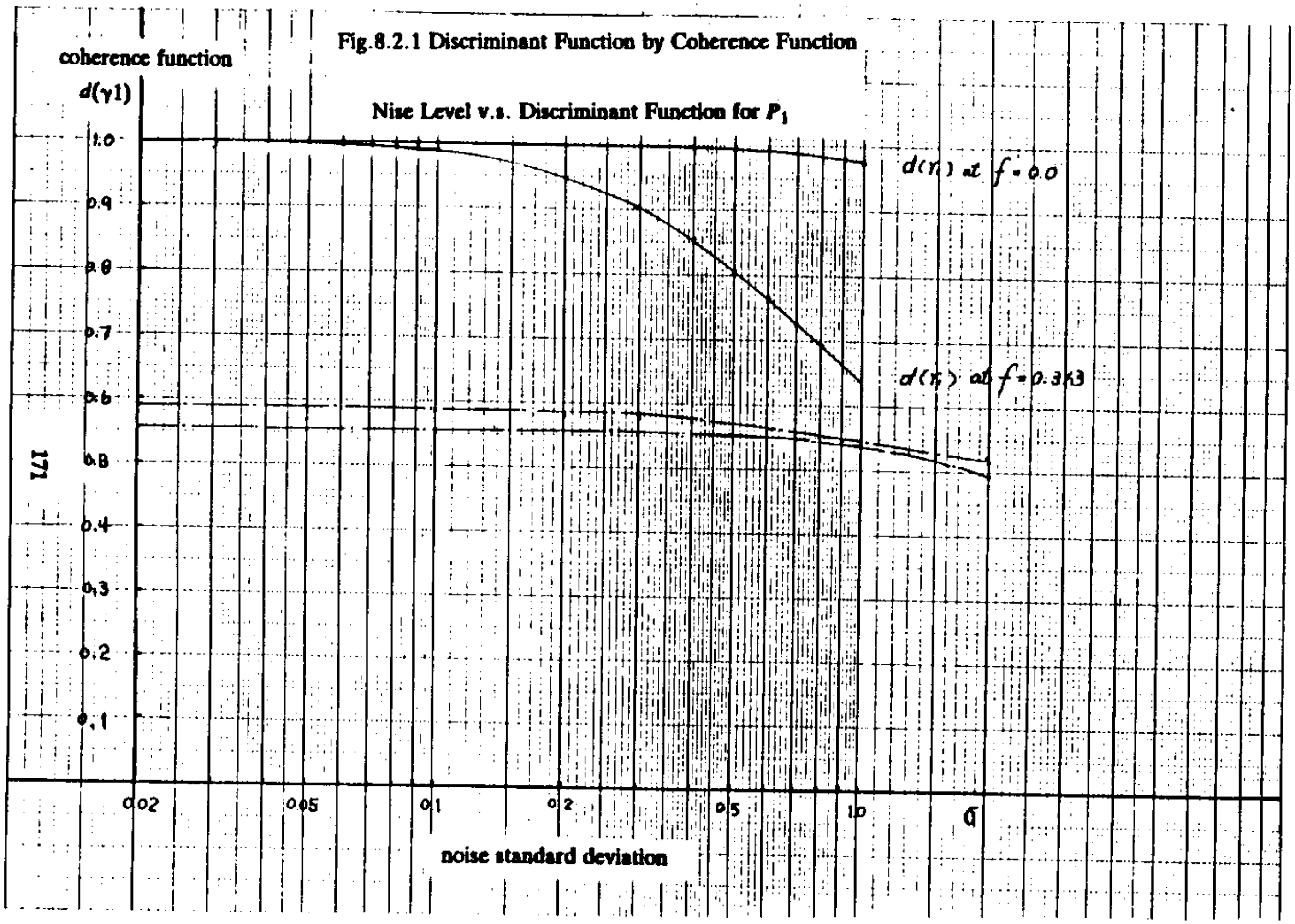


Fig.8.2.1 Discriminant Function by Coherence Function

Nise Level v.s. Discriminant Function for P_1

8.3 Example 2

In this section, a distributed pollution source in a river pollution system is identified from DO measurement. The distributed pollution source to be identified is more general in that the time function f(t) of the pollution source F(x,t) is only assumed to be Fourier series. As formulated in Section 6.3, first, the time function f(t) is identified by the pattern recognition approach. Then, the space function g(t) is identified by the least square method. In order for the pattern recognition approach to be applied to more general situations, Fourier series with a basic frequency of ω are employed to describe the distributed pollution source. Again, coherence functions are employed as the elements of the feature vectors. The mathematical model with BOD index is :

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + F(x,t) \tag{8.3-1}$$

The mathematical model with DO index :

$$\frac{\partial v}{\partial t} = a\frac{\partial^2 v}{\partial x^2} - b(x)\frac{\partial v}{\partial x} - K_r(v_s - v) + K_d u \tag{8.3-2}$$

Assume the following conditions and numerical values for the parameters:

$$a = 0.01 \tag{8.3-3}$$

$$b = 0.25(\ sin(\pi x/x_f) + 1.0\ ) \tag{8.3-4}$$

$$c = 0.1 \tag{8.3-5}$$

$$K_r = 0.1 \tag{8.3-6}$$

$$K_d = 0.5 \tag{8.3-7}$$

$$v_s = 5.0 \tag{8.3-8}$$

and initial conditions

$$u(x,0) = 0 \quad x_0 \leq x \leq x_f \tag{8.3-9}$$

$$v(x,0) = v_s \tag{8.3-10}$$

and boundary conditions

$$u(x_0,t) = 0 \quad 0 \leq t \tag{8.3-11}$$

$$u(x_f,t) = 0 \tag{8.3-12}$$

$$v(x_0,t) = v_s \tag{8.3-13}$$

$$v(x_f,t) = v_s \tag{8.3-14}$$

where

$$0 \leq x \leq 4.0 \quad 0 \leq t \leq 25.6 \tag{8.3-15}$$

The other conditions are assumed to be the same as in Section 8.2. The measurement data of DO are obtained as four stations within the region under consideration and involve measurement noise as follows :

$$z(x_{m_1},t_k) = v(x_{m_1},t_k) + e_1(t_k) \quad at\ x_{m_1} = 1.0 \tag{8.3-16}$$

$$z(x_{m_2},t_k) = v(x_{m_2},t_k) + e_2(t_k) \quad at\ x_{m_2} = 2.0 \tag{8.3-17}$$

$$z(x_{m_3},t_k) = v(x_{m_3},t_k) + e_3(t_k) \quad at\ x_{m_3} = 3.0 \tag{8.3-18}$$

$$z(x_{m_4},t_k) = v(x_{m_4},t_k) + e_4(t_k) \quad at\ x_{m_4} = 3.5 \tag{8.3-19}$$

where $e_i(t_k)$ for i=1,4 are measurement noise.

Objective

To identify the distributed pollution source F(x,t) from the measurement data (8.3-16) to (8.3-19) as well as the mathematical model (8.3-1) to (8.3-14).

First of all, let us identify the time pattern of the pollution source. For numerical computations the following space and time step sizes are specified :

$$\Delta x = 0.1 \tag{8.3-20}$$

$$\Delta t = 0.1 \tag{8.3-21}$$

To solve the two partial differential equations, Eqn.(8.3-1) and (8.3-2), the explicit finite difference method is employed.

Pattern of pollution source F(x,t)

The distributed pollution source F(x,t) which is function of time and space can be generated as

$$F(x,t) = g(x)f(t) \tag{8.3-22}$$

As discussed in Section 6.3, first, the time function f(t) is identified by the pattern recognition. Then, the space function g(x) is identified by conventional identification approach. Therefore, only subclasses of the time function f(t) are generated for the pattern recognition approach. By taking advantage of a priori knowledge of the pollution source, daily, seasonally or yearly patterns can be assumed. For example, the following subclasses, which are expressed by Fourier series, might be generated from a priori information as :

$$P_0 : f_1(t) = a_0 \tag{8.3-23}$$

$$P_1 : f_2(t) = a_0 + a_1 cos(\omega t + \phi_1) \qquad \text{(8.3-24)}$$

$$P_2 : f_3(t) = a_0 + a_1 cos(\omega t + \phi_1) + a_2 cos(2\omega t + \phi_2) \qquad \text{(8.3-25)}$$

.

.

$$P_n : f_n(t) = a_0 + \sum_{i=1}^{n} a_i cos(i\omega t + \phi_i) \qquad \text{(8.3-26)}$$

where $\omega(=2\frac{\pi}{T})$, or T (=basic time pattern period) can be roughly estimated by a priori information or posteriori measurement data behaver.

Feature Vector Extraction

In order to extract the feature of the subclasses, coherence function $r_j(x_{m_i},f)$ are employed as the elements of the feature vectors γ_j.

$$r_j^2(x_{m_i},f) = \frac{|\overline{C_j(x_{m_i},f)}|^2}{Bm(x_{m_i},f)B_j(x_{m_i},f)} \qquad \text{(8.3-28)}$$

where

$\overline{Bm(x_{m_i},f)}$: power spectra density function of measurement data $z(x_{m_i},t)$

$\overline{B_j(x_{m_i},f)}$: power spectra density function of the solution $v_j(x_{m_i},t)$ based on the subclass $F_j(x,t)$

$\overline{C_j(x_{m_i},f)}$: cross spectra density function between $v_j(x_{m_i},t)$ and $z(x_{m_i},t)$

i (=4) : the number of the measurement station

j (=n) : the number of the subclass of $F_j(x,t)$

The feature vectors can be easily expressed as:

$$\gamma_j = [\ r_j^2(x_{m_1},f), r_j^2(x_{m_2},f), r_j^2(x_{m_3},f), r_j^2(x_{m_4},f)\ ] \tag{8.3-29}$$

Classification

The classification process involves making a decision as to which pollution pattern among the subclasses is most close to the actual pollution sources. A discriminant function based on the feature vector, together with a suitable threshold, is used to that end.

$$d(\gamma_j) = W\gamma_{j_T} \tag{8.3-30}$$

$$= \frac{1}{4}\sum_{i=1}^{4} r_j^2(x_{m_i},f) \tag{8.3-31}$$

where

$$W = [\ \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\ ] \tag{8.3-32}$$

Simulation Results

For simulation purposes, "measurement data" at the four DO measurement locations were generated by solving Eqn.(8.3-1) and (8.3-2), using for F(x,t) which is identical to the subclass P_3 which contains the following numerical data :

$$F_{act}(x,t) = g_{act}(x)f_{act}(t)$$

$$= 2.0sin(\pi x/x_f)\left[1.0+cos(\omega t+\frac{\pi}{4})+0.5cos(2\omega t+\frac{\pi}{2})+0.25cos(3\omega t)\right] \tag{8.3-33}$$

with added Gaussian noise with zero mean and standard deviation $\sigma = 0.1$.

These data are presented in Table 8.3.1.

Next, these measurement data were transformed into frequency domain data $v_m(x_{m_i},f)$, and their power spectra were calculated using conventional signal processing techniques involving : tapering by window functions, Fourier transform by FFT, calibration for truncation, and smoothing and averaging. The power spectra are presented in Table 8.3.2. Note that there are only four frequency components with substantial amplitudes, while all other frequency components are negligible.

To test the classification procedure, solutions at the four measuring points based on each of the subclasses of input functions were calculated using the finite difference method. The resulting solutions $v_j(x_i,t)$ were transformed into the frequency domain using the FFT. Finally, feature vectors γ_j were calculated using Eqn.(8.3-28) and the discriminant functions are calculated according to Eqn.(8.3-31). These computational results are presented in Table 8.3.3. For each subclass P_i, their parameter values a_i, ϕ_i are optimized so that the summation of the discriminant functions, $\sum_{i=1}^{4} d_j(\gamma_i)$ have maximum values. This optimization problem was executed by the univariate searching method [Fox71]. The optimized parameters for each subclass are presented in Table.8.3.4. Since there is a relation between subclasses P_i such that

$$P_0 \subset P_1 \subset P_2 \ldots\ldots \subset P_i \subset \ldots\ldots \subset P_n \tag{8.3-34}$$

the discriminant functions d_i also have a relation such that

Table 8.3.1 Measurement Data of DO

Time	$Vm(x_1,t)$	$Vm(x_2,t)$	$Vm(x_3,t)$	$Vm(x_4,t)$
.00	5.00000	5.00000	5.00000	5.00000
.10	4.84034	5.03925	4.94436	5.17203
.20	4.94971	5.07066	4.98918	5.02436
.30	4.98301	4.86721	4.98444	4.99121
.40	4.88507	.5.02589	5.01173	4.89227
.50	5.13642	4.92114	4.76620	4.93046
.60	5.09122	5.06797	5.09955	4.81875
.70	5.15465	5.12273	4.94037	4.99008
.80	4.83841	5.01573	5.00404	4.98826
.90	4.85754	4.99780	4.86100	4.95581
1.00	4.91818	4.89655	5.09034	5.02747
1.10	5.02934	4.89245	5.05133	5.00187
1.20	5.22654	4.73038	4.91001	4.96809
1.30	4.80558	4.89559	5.06350	4.95901
1.40	4.87201	4.96074	4.85403	4.81339
1.50	4.91620	4.84033	5.00796	5.00879
1.60	4.73089	4.96816	4.78979	4.80799
1.70	4.75576	4.85934	4.80252	4.98128
1.80	4.85574	4.80373	4.76065	4.97491
1.90	5.05365	4.78268	4.76274	4.73398
2.00	4.86573	4.80754	4.93964	4.65795
2.10	4.91040	5.01691	4.71314	4.77771
2.20	4.95214	4.69957	4.70986	4.75716
2.30	4.89496	4.78581	4.72523	4.74967
2.40	4.85717	4.77458	4.83151	4.98344
2.50	5.01439	4.54018	4.79686	4.76699
2.60	4.98048	4.79532	4.56881	4.76574
2.70	5.03903	4.74109	4.70258	4.73356
2.80	4.77325	4.83066	4.75690	4.77897
2.90	4.77002	4.82345	4.57799	4.59953
3.00	4.78362	4.85556	4.70106	4.61689
3.10	4.79140	4.62491	4.47873	4.67507
3.20	4.88960	4.58952	4.57798	4.57004

Table 8.3.2 Power Spectra Density of Measurement Data

Frequ	$Bm(x_1,f)$	$Bm(x_2,f)$	$Bm(x_3,f)$	$Bm(x_4,f)$
.0000	.1326339e+02	.7588817e+02	.1472964e+03	.1486962e+03
.3125	.5055708e+00	.9136800e+00	.4716505e+00	.2167033e+00
.6250	.5873273e-01	.1474542e+00	.1253791e+00	.9020714e-01
.9375	.1028304e-01	.3515814e-01	.3322197e-01	.2793627e-01
1.2500	.6513150e-02	.2291196e-01	.2197842e-01	.1417530e-01
1.5625	.7095356e-02	.9694351e-02	.1665511e-01	.1383096e-01
1.8750	.3697958e-02	.9044702e-02	.1079138e-01	.6399042e-02
2.1875	.2653246e-02	.7316845e-02	.6225013e-02	.6276168e-02
2.5000	.3157692e-02	.5559013e-02	.8563926e-02	.7284626e-02
2.8125	.1500197e-02	.5623172e-02	.8022639e-02	.5594852e-02
3.1250	.2370585e-02	.3698822e-02	.5290414e-02	.3901769e-02
3.4375	.2037616e-02	.3713295e-02	.4285173e-02	.3009835e-02
3.7500	.2165662e-02	.4701524e-02	.4246833e-02	.2780752e-02
4.0625	.2672509e-02	.2476008e-02	.4372450e-02	.3788869e-02
4.3750	.2189758e-02	.4565815e-02	.4577229e-02	.2170503e-02
4.6875	.2389260e-02	.4328977e-02	.5067400e-02	.4327997e-02
5.0000	.1362975e-02	.4754798e-02	.6085012e-02	.2258275e-02

Table.8.3.3 Feature Vectors and Discriminant Function

subclass	f	$r^2(x_{m_1},f)$	$r^2(x_{m_2},f)$	$r^2(x_{m_3},f)$	$r^2(x_{m_4},f)$	$d(\gamma_j)$
P_0	.0	.99528	.99670	.99776	.99820	.99699
	.313	.11989	.23166	.40575	.68318	.36012
	.626	.50637	.69231	.85388	.89465	.73680
	.939	.58103	.83555	.93377	.92906	.81985
P_1	.0	.99959	.99989	.99992	.99992	.99983
	.313	.99831	.99909	.99907	.99699	.99836
	.626	.50797	.70495	.86809	.91360	.74865
	.939	.58295	.83306	.95028	.96416	.83261
P_2	.0	.99978	.99998	.99999	.99999	.99994
	.313	.99800	.99915	.99871	.99662	.99812
	.626	.99411	.99381	.99738	.98845	.99344
	.939	.58330	.83230	.95193	.96918	.83418
P_3	.0	.99978	.99996	.99998	.99999	.99993
	.313	.99789	.99912	.99862	.99644	.99802
	.626	.99390	.99393	.99729	.98840	.99338
	.939	.97686	.99338	.99091	.98955	.98767
P_4	.0	.99978	.99996	.99998	.99999	.99993
	.313	.99789	.99912	.99862	.99644	.99802
	.626	.99390	.99393	.99729	.98840	.99338
	.939	.97686	.99338	.99091	.98955	.98767

Table.8.3.4 The Optimal Parameters for Each Subclass

$$P_{act} \; : \; f_{act} = 1.0 + 1.0cos(\omega t+0.785398)$$

$$+ \; 0.5cos(2\omega t+1.5707963) + 0.25cos(3\omega t+0.0)$$

$$P_0 \; : \; f_0 = 1.0$$

$$P_1 \; : \; f_1 = 1.0 + 1.0300cos(\omega t+0.8654)$$

$$P_2 \; : \; f_2 = 1.0 + 1.0300cos(\omega t+0.8654) + 0.5033cos(2\omega t+1.5748)$$

$$P_3 \; : \; f_3 = 1.0 + 1.0300cos(\omega t+0.8654) + 0.5033cos(2\omega t+1.5748)$$

$$+ \; 0.2810cos(3\omega t+0.0335)$$

$$P_4 \; : \; f_4 = 1.0 + 1.0300cos(\omega t+0.8654) + 0.5033cos(2\omega t+1.5748)$$

$$+ \; 0.2810cos(3\omega t+0.0335) + 0.0060cos(4\omega t+0.0020)$$

$$d_1 \leq d_2 \leq d_3 \ldots \leq d_n \leq 1.0 \tag{8.3-35}$$

Therefore, as the index i is increased, the discriminant function d_i increases until the subclass P_3 (which is equivalent to the original time pattern $f_{act}(t)$) is reached. After reaching P_3, that is for P_n (for n>4), the discriminant function becomes almost constant and does not improve the value of the discriminant function as presented in Fig.8.3.2. This means that P_3 is sufficient to describe the original time function $f_{act}(t)$. Thus the time function of the original pollution source function can be identified. The time functions for each subclass are presented in Fig.8.3.1. It is found that the P_3 is almost identical to the original time function $f_{act}(t)$

Once the time function is determined, the space function g(x) of the pollution source is identified. As shown in Section 6.3, the space function can be identified by the least square method which is based on the iterative equations formulated in Eqn.(6.4-29) through (6.4-32). The computational results of g(x) for different noise levels are shown in Fig.8.3.2. As can be seen from Fig.8.3.2, the original space function can be identified clearly in a presence of high noise levels.

8.4 Example 3

In this section, distributed pollution sources in aquifer pollution systems are identified from measurement data. The aquifer under study is assumed to be located near the surface where pollutants are injected by some factories. The

Fig.8.3.1 The Time Functions for Each Subclass

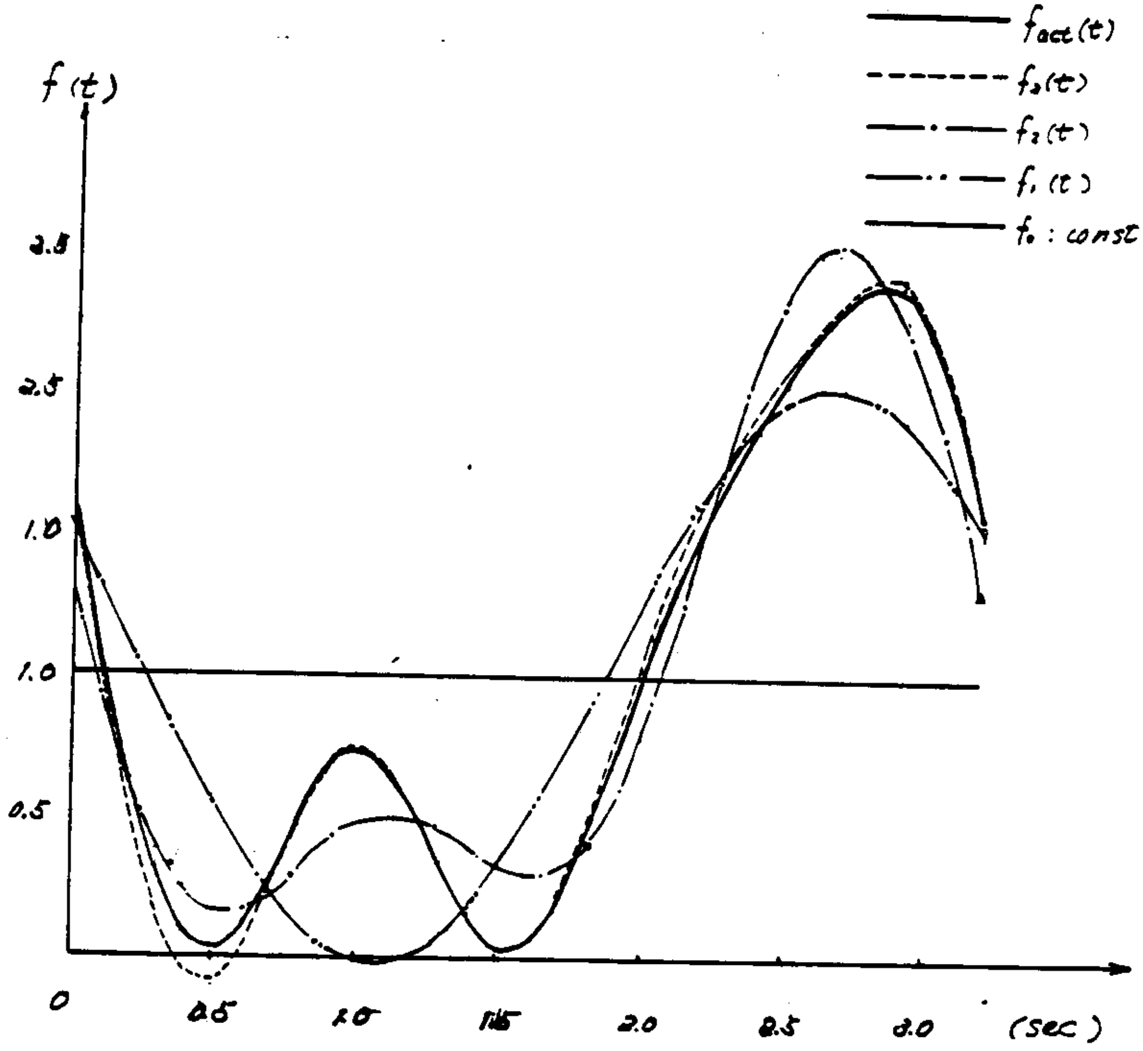
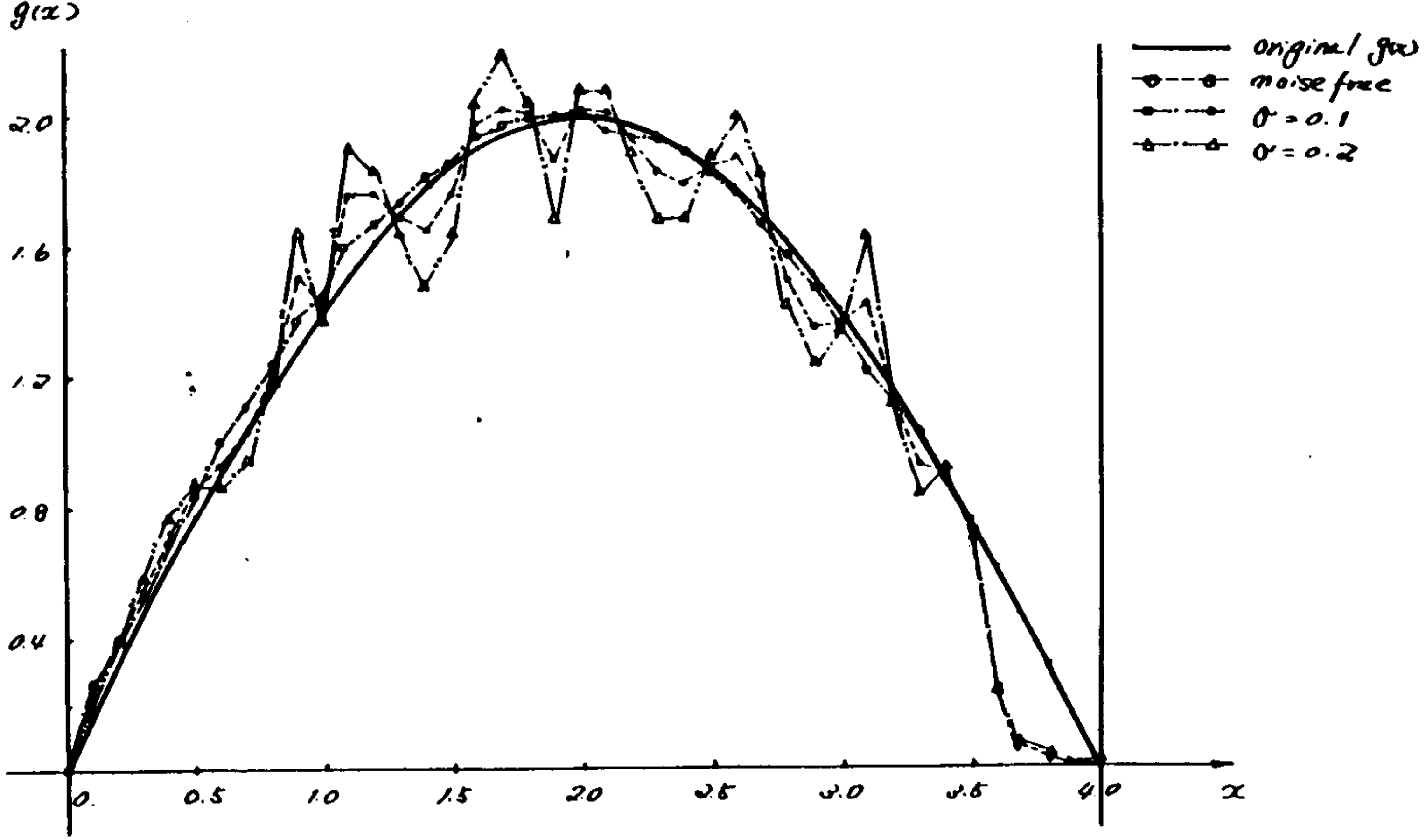


Fig.8.3.3 Ideentification of The Space Function

by The Least Square Method



	1	2	3	4	5	6	7	8	9	10
1	Q									
2	Q									
3	Q		M						M	
4	Q									
5	Q				M					
6	Q									
7	Q									
8	Q									
9	Q		M						M	
10	Q									

impermeable

M : Measurement Stations at

(1500,1500),(1500,4500),(2500,2500),(3500,1500),(3500,4500)

Q : Aquifer Pumping with rate = 500

S : Storage Coefficient = 0.015

n : Porosity = 0.3

$\delta x = \delta y = 500$

location and intensity of the pollution source to be identified is assumed to be unknown. The aquifer pollution system near the surface under ground is characterized by the following two dimensional equations :

The aquifer flow equation is

$$S\frac{\partial h}{\partial t} = \frac{\partial}{\partial x}\left(K_x(x,y)\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y(x,y)\frac{\partial h}{\partial y}\right) - Q \qquad (8.4\text{-}1)$$

where h(x,y,t) is the hydraulic head, $K_x(x,y),K_y(x,y)$ is the conductivity (assumed to be a function of x and y), S is the specific storage (assumed to be constant), and Q is volume flow (assumed to be constant). The aquifer quality system is

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D_x(x,y,t)\frac{\partial c}{\partial x}\right) + \left(D_y(x,y,t)\frac{\partial c}{\partial y}\right)$$

$$- \frac{\partial}{\partial x}\left(v_x(x,y,t)c\right) - \frac{\partial}{\partial y}\left(v_y(x,y,t)c\right) + \frac{cQ}{n} + F(x,y,t) \qquad (8.4\text{-}2)$$

where c(x,y,t) is the pollutant concentration, D(x,y,t) is the diffusion coefficient (assumed to be a function of space and time), n is porosity (assumed to be a constant), and F(x,y,t) is the pollution source to be identified.

The Darcy equations are

$$v_x(x,y,t) = \frac{K_x(x,y)}{b}\frac{\partial h}{\partial x} \qquad (8.4\text{-}3)$$

$$v_y(x,y,t) = \frac{K_y(x,y)}{b}\frac{\partial h}{\partial y} \qquad (8.4\text{-}4)$$

The diffusion coefficients are

$$D_x(x,y,t) = e_x v_x(x,y,t) \tag{8.4-5}$$

$$D_y(x,y,t) = e_y v_y(x,y,t) \tag{8.4-6}$$

where e_x,e_y are dispersion coefficients. Assume numerical values for these parameters as :

$$K_x(x,y) = K_y(x,y) = 31000.0 - 2.25(x+y) \tag{8.4-7}$$

$$S = 0.015 \tag{8.4-8}$$

$$e_x = e_y = 50.0 \tag{8.4-9}$$

$$Q = 500.0 \tag{8.4-10}$$

and initial condition

$$h(x,y,0) = 200.0 \quad x,y \in D \tag{8.4-11}$$

$$c(x,y,0) = 0.0 \quad x,y \in D \tag{8.4-12}$$

and boundary conditions

$$h(x_b,y_b,t) = 200.0 \quad x_b,y_b \in \partial D \tag{8.4-13}$$

$$c(x_b,y_b,t) = 0 \tag{8.4-14}$$

where the space region ,D, is

$$0 \le x \le 5000 \quad 0 \le y \le 5000 \tag{8.7-15}$$

and time region, T, is

$$0 \le t \le 2000.0 \quad t \in T \tag{8.4-16}$$

The measurement data are obtained at five stations, within the boundary of the region under consideration, and involve measurement noise as follows :

$$z(x_{m_i},y_{m_i},t_k) = c(x_{m_i},y_{m_i},t_k) + e_i(t_k) \;\text{ for } i=1,2,\ldots,5 \qquad (8.4\text{-}17)$$

where $e_i(t)$ for i=1,5 are measurement noise.

Objective

To identify the distributed pollution sources F(x,y,t) from measurement data. The entire identification procedure is indicated in Section 6.3.3. For numerical computations the following space and time step sizes are specified :

$$\Delta x = \Delta y = 500.0 \qquad (8.4\text{-}18)$$

$$\Delta\, t = 20.0 \qquad (8.4\text{-}19)$$

To solve the partial differential equations, the explicit finite difference approximation method is employed.

Pattern of Pollution Sources F(x,y,t)

The distributed pollution sources F(x,y,t) are functions of time and space and can be expressed generally as :

$$F(x,y,t) = g(x,y)f(t) \qquad (8.4\text{-}20)$$

By taking advantage of a priori knowledge of pollution sources, that is, the locations of factories or their daily or monthly or yearly operation patterns, etc., explicit subclasses of F(x,y,t) can specified as the nominated pollution patterns. For example, the following five subclasses might be generated from a priori information :

$$P_n :\; F_n(x,t) = g(x,y)\left\{(a_0 + \sum_{i=1}^{n} a_i\cos(i\omega t+\phi_i)\right\} \qquad (8.4\text{-}21)$$

Feature Vector Extraction

In order to extract the feature of the subclasses, coherence functions $r_j(x_{m_i},y_{m_i},f)$ are employed as the elements of the feature vectors γ_j.

$$r_j^2(x_{m_i},y_{m_i},f) = \frac{|\,\overline{C_j(x_{m_i},y_{m_i},f)}\,|^2}{Bm(x_{m_i},y_{m_i},f)B_j(x_{m_i},y_{m_i},f)} \qquad (8.2\text{-}22)$$

where

$\overline{Bm(x_{m_i},y_{m_i},f)}$: power spectra density function of measurement data $z(x_{m_i},y_{m_i},t)$

$\overline{B_j(x_{m_i},subm_i,f)}$: power spectra density function of the solution $\overline{c_j}(x_i,subm_i,t)$ based on the subclass $F_j(x,y,t)$

$\overline{C_j(x_{m_i},y_{m_i},f)}$: cross spectra density function between $c_j(x_{m_i},y_{m_i},t)$ and $z(x_{m_i},y_{m_i},t)$

i (=5) : the number of the measurement station

j (=n) : the number of the subclass of $F_j(x,y,t)$

The feature vectors can be easily expressed as:

$$\gamma_j = [\ r_j^2(x_{m_1},y_{m_1},f), r_j^2(x_{m_2},y_{m_2},f), r_j^2(x_{m_3},y_{m_3},f), r_j^2(x_{m_4},y_{m_4},f), r_j^2(x_{m_5},y_{m_5},f)\ ] \qquad (8.4\text{-}23)$$

Classification

The classification process involves making a decision as to which pollution pattern among the subclasses is most close to the actual pollution sources. A discriminant function based on the feature vector, together with a suitable threshold, is used to that end.

$$d(\gamma_j) = W\gamma_j^T \tag{8.2-24}$$

$$= \frac{1}{5}\sum_{i=1}^{5} r_j^2(x_{m_i}, y_{m_i}, f) \tag{8.2-25}$$

where

$$W = [\ \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\ ] \tag{8.2-26}$$

Simulation Results

For simulation purposes, "measurement data" at the five measurement locations were generated by solving Eqn.(8.4-1) through (8.4-2), using for F(x,y,t) a transient identical to subclass P_3 which is expressed by

$$F(x,y,t) = sin(\pi x/20000)sin(\pi y/20000)\sin^2(\pi t/320) \tag{8.4-27}$$

and added Gaussian noise with zero mean and standard deviation σ = 0.1. These data are presented in Table 8.4.1.

Next these measurement data were transformed into frequency domain data $z_m(x_{m_i}, y_{m_i}, f)$, and their power spectra were calculated using conventional signal processing techniques involving: tapering by window functions, Fourier transform by FFT, calibration for truncation smoothing and averaging, the power spectra are presented in Table 8.4.2.

To test the classification procedure, solutions at the five measuring points based on each of the five subclasses of input functions, were calculated using both the finite difference method. The resulting solutions $c_j(x_{m_i}, y_{m_i}, t)$ were transformed into the frequency domain using the FFT. Finally, feature vectors

Table 8.4.1 Measurement Data of Aquifer Pollutants

Time	$z(x_{m_1},y_{m_1},t)$	$z(x_{m_2},y_{m_2},t)$	$z(x_{m_3},y_{m_3},t)$	$z(x_{m_4},y_{m_4},t)$	$z(x_{m_5},y_{m_5},t)$
.00	.00000	.00000	.00000	.000000	.00000
.05	.83532	.34671	.25182	1.40188	.33206
.10	2.17257	.31907	.34695	2.34705	.30922
.15	3.03628	.31091	.20533	3.18419	.44665
.20	3.51773	.44354	.22868	3.25420	.33237
.25	3.84090	.35850	.37688	3.22482	.52990
.30	3.63805	.19886	.23141	2.96447	.36907
.35	3.17465	.21184	.08492	2.81640	.17540
.40	2.76972	.13095	.12889	2.59169	.30030
.45	2.53522	.11816	.26473	2.21006	.48308
.50	1.99113	.11913	.15115	1.77474	.17179
.55	2.07176	.13562	.04399	1.75372	.10996
.60	1.56507	.07747	.05055	1.58878	.22128
.65	1.24871	.17416	-.01047	1.15550	-.06352
.70	1.22627	.00268	.14336	.98959	.03925
.75	.94819	.14031	.18257	.83192	.00022
.80	.75158	-.00499	.02174	.89369	-.13747
.85	.80376	.24439	-.04560	.64851	.10799
.90	.67370	-.02732	-.02628	.70364	.06149
.95	.74942	-.01440	.00839	.73089	.15467
1.00	.99045	.17111	-.17559	.81367	.05526
1.05	.87604	.08960	-.10066	.70876	.22071
1.10	.70211	.04255	.01112	.50045	.16601
1.15	.61737	.05970	-.08420	.56063	-.03673
1.20	.19971	-.07478	.19456	.30698	-.06676
1.25	.04821	-.01135	-.09616	.04369	.05012
1.30	-.03401	.05233	-.04842	.02802	.06328
1.35	.08737	.11394	.06357	.09146	-.05867
1.40	.14843	.12858	.18138	.08011	.05253
1.45	.63277	-.00066	.07565	.56812	.12751
1.50	1.33006	.24193	.20567	1.35503	.07903
1.55	2.07693	.11339	.36493	2.14354	.23078
1.60	2.85543	.40324	.21776	2.63654	.40262

Table 8.4.2 Power Spectra Density of Measurement Data

Frequ	$Bm(x_{m_1},y_{m_1},f)$	$Bm(x_{m_2},y_{m_2},f)$	$Bm(x_{m_3},y_{m_3},f)$	$Bm(x_{m_4},y_{m_4},f)$	$Bm(x_{m_5},y_{m_5},f)$
.000	.163723e+01	.118769e-01	.104825e-01	.137971e+01	.191031e-01
.625	.403482e+00	.285676e-02	.368023e-02	.349520e+00	.404966e-02
1.250	.109900e+00	.146066e-02	.753928e-03	.946544e-01	.255083e-02
1.875	.367536e-01	.548892e-03	.379635e-03	.320423e-01	.523805e-03
2.500	.241095e-01	.645715e-03	.322972e-03	.184053e-01	.342726e-03
3.125	.141106e-01	.287364e-03	.267599e-03	.114986e-01	.220825e-03
3.750	.768899e-02	.484182e-03	.201049e-03	.556541e-02	.205033e-03
4.375	.285319e-02	.887761e-04	.298273e-03	.251022e-02	.496061e-03
5.000	.129248e-02	.728366e-04	.768498e-04	.150714e-02	.183690e-03
5.625	.117645e-02	.115902e-03	.242817e-03	.604663e-03	.101708e-03
6.250	.180117e-02	.192102e-03	.341892e-03	.153849e-02	.364662e-03
6.875	.391251e-02	.126545e-03	.543017e-04	.258026e-02	.216759e-03
7.500	.303287e-02	.187923e-03	.264130e-03	.273074e-02	.109342e-03
8.125	.266530e-02	.360196e-03	.147285e-03	.202787e-02	.353010e-03
8.750	.240209e-02	.149460e-03	.133113e-03	.152652e-02	.150661e-03
9.375	.999845e-03	.495459e-03	.139833e-03	.535298e-03	.298554e-03
10.00	.282463e-03	.845758e-04	.295471e-03	.401852e-03	.238025e-03

based on the coherence γ_j were calculated using Eqn.(8.4-21) and the discriminant functions were calculated according to Eqn.(8.4-22). These computational results are presented in Table 8.4.3. For each subclass P_i, their parameter values of a_i, ϕ_i are optimized so as the summation of the discriminant functions, $\sum_{i=1}^{5} d_j(\gamma_i)$ have maximum values. The optimized parameters for each subclass are presented in Table.8.4.4. As the number of i is increased, the discriminant function d_i increases until the subclass P_3 which is equivalent to the original time pattern $f_{act}(t)$. Beyond P_3, that is, for P_n (for $n \geq 4$) the discriminant function becomes almost constant. This means that P_3 is sufficient to describe the original time function $f_{act}(t)$. Thus the time function of the original pollution source function can be identified. The time functions for each subclass are presented in Fig.8.4.1. It is found that the P_3 is almost identical to the original time function $f_{act}(t)$.

Once the time function is determined, the space function g(x,y) of the pollution source is identified. As derived in Section 6.4, the space function can be identified by the least square method which is based on the iterative equations formulated in Eqn.(6.4-29) through (6.4-32). The computational results of g(x) for different noise levels are shown in Fig.8.4.2. As can be seen from Fig.8.4.2, the original space function can be identified clearly in a presence of high noise levels.

Table.8.4.3 Feature Vectors and Discriminant Function

subclass	f	$r^2(x_{m_1},y_{m_1},f)$	$r^2(x_{m_2},y_{m_2},f)$	$r^2(x_{m_3},y_{m_3},f)$	$r^2(x_{m_4},y_{m_4},f)$	$r^2(x_{m_5},y_{m_5},f)$	$d(\gamma_j)$
P_0	.000	.99967	.98568	.97911	.99966	.99068	.99096
	.625	.10717	.59597	.53530	.10500	.35847	.43038
	1.250	.08134	.58222	.15878	.07273	.21231	.22147
	1.875	.02177	.38330	.07285	.02627	.25317	.15147
	2.500	.82626	.30950	.26685	.69561	.07934	.43551
P_1	.000	.99984	.98560	.97913	.99980	.99071	.99101
	.625	.99990	.97280	.98268	.99979	.98635	.98831
	1.250	.07988	.64309	.17602	.07230	.21518	.23729
	1.875	.02109	.40479	.07421	.02597	.25189	.15559
	2.500	.82509	.31365	.26597	.69487	.07820	.43556
P_2	.000	.99988	.98557	.97914	.99984	.99072	.99103
	.625	.99987	.97282	.98267	.99983	.98631	.98831
	1.250	.99888	.96745	.86368	.99897	.95614	.95703
	1.875	.02058	.46501	.08280	.02575	.25515	.16986
	2.500	.82410	.32858	.26820	.69423	.07817	.43866
P_3	0.000	.99992	.98556	.97915	.99988	.99073	.99105
	.625	.99977	.97282	.98266	.99984	.98629	.98828
	1.250	.99862	.96747	.86380	.99909	.95623	.95704
	1.875	.99741	.81275	.84580	.98980	.86013	.90118
	2.500	.82427	.37106	.28527	.69428	.08393	.45178
P_4	.000	.99993	.98555	.97915	.99989	.99073	.99105
	.625	.99976	.97282	.98266	.99983	.98628	.98828
	1.250	.99858	.96747	.86381	.99909	.95624	.95704
	1.825	.99724	.81274	.84585	.99006	.86000	.90118
	2.500	.96980	.59563	.92623	.97981	.85909	.86611

Table.8.4.4 The Optimal Parameters for Each Subclass

$$P_{act} \; : \; f_{act} = 1.0 + 1.0cos(\omega t - 0.785398)$$
$$+ \; 0.5cos(2\omega t - 0.785398) + 0.20cos(3\omega t + 0.0)$$

$$P_0 \; : \; f_0 = 1.0$$

$$P_1 \; : \; f_1 = 1.0 + 0.7200cos(\omega t - 0.84482)$$

$$P_2 \; : \; f_2 = 1.0 + 0.7200cos(\omega t - 0.84482) + 0.370cos(2\omega t - 0.83940)$$

$$P_3 \; : \; f_3 = 1.0 + 0.7200cos(\omega t - 0.84482) + 0.370cos(2\omega t - 0.83940)$$
$$+ \; 0.2740cos(3\omega t + 0.0705)$$

$$P_4 \; : \; f_4 = 1.0 + 0.7200cos(\omega t - 0.84482) + 0.370cos(2\omega t - 0.83940)$$
$$+ \; 0.2740cos(3\omega t + 0.0705) + 0.0330cos(4\omega t + 0.140)$$

Fig.8.4.1 The Time Functions for Each Subclass

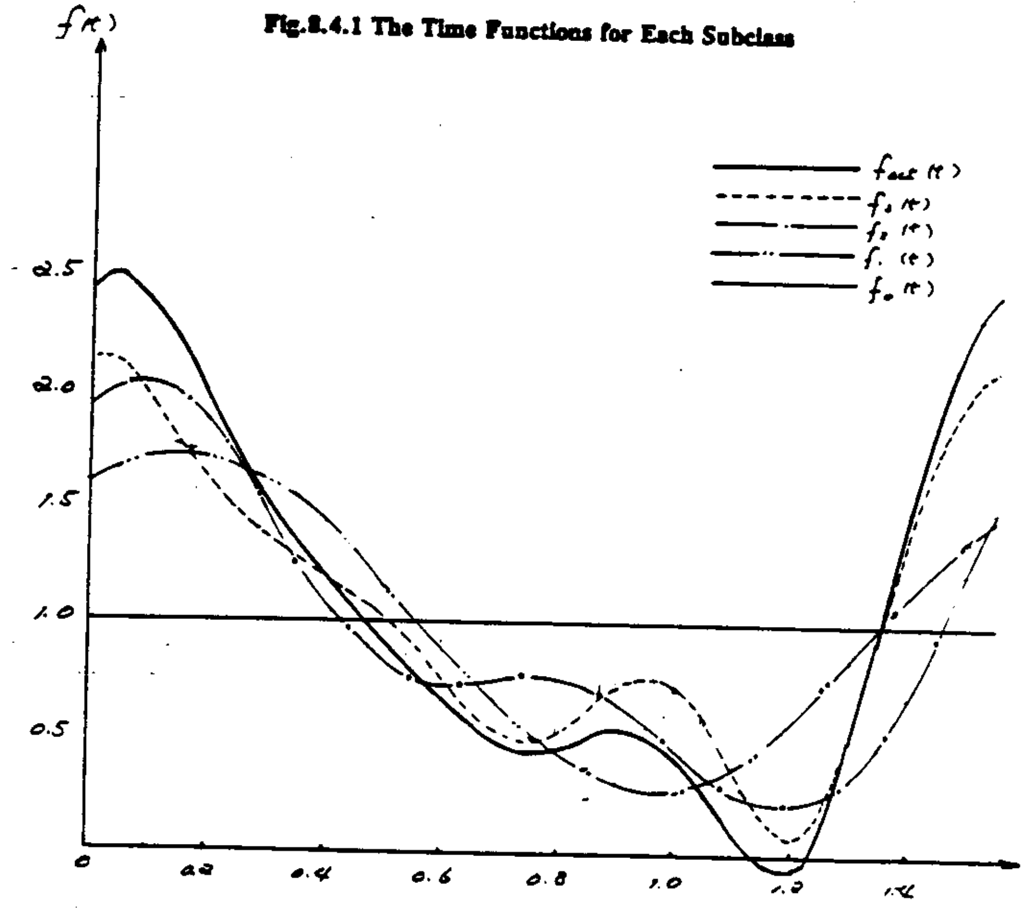


Fig.8.4.2 Identification of the Space Function

by the Least Square Method

The Original Space Function

	1	2	3	4	5	6	7	8	9	10
1	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
2	.000	.954	1.816	2.500	2.938	3.090	2.938	2.500	1.816	.954
3	.000	1.816	3.454	4.755	5.590	5.877	5.590	4.755	3.454	1.816
4	.000	2.500	4.755	6.545	7.694	8.090	7.694	6.545	4.755	2.500
5	.000	2.938	5.590	7.694	9.045	9.510	9.045	7.694	5.590	2.938
6	.000	3.090	5.877	8.090	9.510	10.000	9.510	8.090	5.877	3.0902
7	.000	2.938	5.590	7.694	9.045	9.510	9.045	7.694	5.590	2.9389
8	.000	2.500	4.755	6.545	7.694	8.090	7.694	6.545	4.755	2.5000
9	.000	1.816	3.454	4.755	5.590	5.877	5.590	4.755	3.454	1.8164
10	.000	.954	1.816	2.500	2.938	3.090	2.938	2.500	1.816	.954

The Calculated Space Function

	1	2	3	4	5	6	7	8	9	10
1	.046	1.016	1.957	2.577	2.850	2.828	2.582	2.167	1.641	1.076
2	.168	1.374	2.632	3.524	3.969	4.010	3.744	3.212	2.460	1.601
3	.275	1.858	3.596	4.914	5.634	5.752	5.507	4.885	3.832	2.500
4	.415	2.424	4.748	6.650	7.757	7.787	7.563	6.947	5.619	3.681
5	.479	2.756	5.361	7.999	9.819	9.144	8.811	8.261	7.030	4.476
6	.208	2.300	4.721	6.781	8.009	8.095	7.877	7.245	5.880	3.796
7	.092	1.811	3.789	5.449	6.442	6.651	6.289	5.388	4.085	2.578
8	.198	1.554	3.235	4.789	5.780	6.040	5.575	4.248	2.904	1.740
9	.437	1.482	2.884	4.608	5.742	6.265	6.653	3.953	2.393	1.345
10	.851	1.715	3.114	4.589	5.565	5.764	5.105	3.506	2.126	1.174

8.5 Example 4

In this section, point pollution sources in river pollution systems are identified from the BOD measurement data. The pollutant concentration u(x,t) of BOD is given by:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} - b(x)\frac{\partial u}{\partial x} - cu + F(x,t) \qquad (8.5\text{-}1)$$

BOD data are obtained by sampling the river water at four locations. Assume numerical values for the parameters:

$$a=0.01 \qquad (8.5\text{-}2)$$

$$b=0.25(\ sin(4\pi x/x_f)+1.0\ ) \qquad (8.5\text{-}3)$$

$$c=0.1 \qquad (8.5\text{-}4)$$

and initial condition

$$u(x,0)=0 \qquad x_0 \le x \le x_f \qquad (8.5\text{-}5)$$

and boundary conditions

$$u(x_0,t)=0 \qquad (8.5\text{-}6)$$

$$u(x_f,t)=0 \qquad (8.5\text{-}7)$$

where

$$0 \le x \le 2.0\ , \qquad 0 \le t \le 10.0$$

The measurement data of BOD are obtained at four stations within the region under consideration and involve measurement noise as follows :

$$z(x_{m1},t_k)=u(x_{m1},t_k)+e_1(t_k) \ \ at\ x_{m1}=0.4$$

(8.5-8)

$$z(x_{m2},t_k)=u(x_{m2},t_k)+e_2(t_k) \quad at\ x_{m2}=0.8 \tag{8.5-9}$$

$$z(x_{m3},t_k)=u(x_{m3},t_k)+e_3(t_k) \quad at\ x_{m3}=1.2 \tag{8.5-10}$$

$$z(x_{m4},t_k)=u(x_{m4},t_k)+e_4(t_k) \quad at\ x_{m4}=1.4 \tag{8.5-11}$$

where $e_i(t)$ for i=1,4 are measurement noise.

Objective

To identify the distributed pollution sources F(x,t) from measurement data as well as the mathematical model. For numerical computations the following space step size and time step size are specified:

$$\Delta x\ =\ 0.1\ or\ 0.2 \tag{8.5-12}$$

$$\Delta t\ =\ 0.1 \tag{8.5-13}$$

To solve the parabolic partial differential equation, the explicit finite difference method is employed. For simulation, the following pollution sources were assumed as actual sources to obtain the output data

$$case1 : F(x,t)\ =\ \delta(x-0.2)\sin^2(\pi t/3.2) \tag{8.5-14}$$

$$case2 : F(x,t)\ =\ \delta(x-0.2)\sin^2(\pi t/3.2)\ +\ 0.2\delta(x-1.2)(1.0+\sin(\pi t/3.2)) \tag{8.5-15}$$

Then random Gaussian noise signal with 0-mean and different standard deviations where added to the output data to generate the measurement data.

First of all, let us identify the point pollution sources locations. Suppose the nominated pollution locations, x_{l_i} are

$$x_{l_1}=0.2,\ \ x_{l_2}=0.8,\ \ x_{l_3}=1.2,\ \ x_{l_4}=1.4 \tag{8.5-16}$$

Therefore, the coefficient matrix B in Eqn.(6.4-5) for $\Delta x = 0.1$ is

$$B = (b_{i,j}) = \begin{cases} b_{1,2}=b_{2,8}=b_{3,12}=b_{4,14}= 1.0 \\ other\ wise\ b_{i,j}=0.0 \end{cases} \tag{8.5-17}$$

Since the measurement locations are

$$x_{m_1=0.4,}\ \ x_{m_2=0.8,}\ \ x_{m_3=1.2,}\ \ x_{m_4}=1.6, \tag{8.5-18}$$

the output matrix C in Eqn.(6.4-7) can be expressed as

$$C = (c_{i,j}) = \begin{cases} c_{1,4}=c_{2,8}=c_{3,12}=c_{4,16}= 1.0 \\ other\ wisec_{i,j}= 0.0 \end{cases} \tag{8.5-19}$$

On the other hand, the measurement data are accumulated from t_0 to t_f as $\sum_{k=0}^{100} Z_k$.

By using these matrices A, B, C and $\sum_{k=0}^{100} Z_k$, the feature vectors α_i can be calculated as follows :

$$(\alpha_i) = [G^T G]^{-1} G^T \sum_{k=0}^{100} Z_k \tag{8.5-20}$$

where

$$G = C[I-A]^{-1}B \tag{8.5-20}$$

The simulation results are shown in Table.8.5.2 to 8.5.4. It is apparent from the simulation results that the feature vector α_1 based on the locations x_{l_1} has a larger value, which is close to the theoretical value. The other feature values α_2 to α_4 are all negative small values which are also close to the theoretical value, zero. The differences between the calculated values and theoretical values are due to

Table.8.5.2 Point Source Identification from th BOD Measurement Data

Case 1 : Single Pollution Source at x_{act} = 0.2

The number of discrete point N=20 (Δx= 0.1)

Nominated source locations
x_{l_1} =0.2 (l_1 = 2)
x_{l_2} =0.8 (l_2 = 8)
x_{l_3} =1.2 (l_3 = 12)
x_{l_4} =1.6 (l_4 = 16)

Computational Results

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	67.5080	64.0000
$\alpha(x_{l2})$		-0.77414	0
$\alpha(x_{l3})$		-0.81959	0
$\alpha(x_{l4})$		-0.57833	0
$\alpha(x_{l1})$	0.2	67.1670	64.0000
$\alpha(x_{l2})$		1.6623	0
$\alpha(x_{l3})$		-3.5419	0
$\alpha(x_{l4})$		1.4293	0
$\alpha(x_{l1})$	0.5	66.6570	64.0000
$\alpha(x_{l2})$		5.3171	0
$\alpha(x_{l3})$		-7.6255	0
$\alpha(x_{l4})$		2.7058	0

Table.8.5.3 Point Source Identification from th BOD Measurement Data

Case 2 : Single Pollution Source at $x_{act} = 0.2$

The number of discrete point N=10 ($\Delta x = 0.2$)

Nominated source locations
$x_{l_1} = 0.2$ ($l_1 = 2$)
$x_{l_2} = 0.8$ ($l_2 = 8$)
$x_{l_3} = 1.2$ ($l_3 = 12$)
$x_{l_4} = 1.6$ ($l_4 = 16$)

Computational Results 2

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	67.3140	64.0000
$\alpha(x_{l2})$		-0.40614	0
$\alpha(x_{l3})$		-0.64727	0
$\alpha(x_{l4})$		-0.41042	0
$\alpha(x_{l1})$	0.2	66.8940	64.0000
$\alpha(x_{l2})$		2.4077	0
$\alpha(x_{l3})$		-3.7839	0
$\alpha(x_{l4})$		0.90757	0
$\alpha(x_{l1})$	0.5	66.2630	64.0000
$\alpha(x_{l2})$		6.6283	0
$\alpha(x_{l3})$		-8.4888	0
$\alpha(x_{l4})$		2.3305	0

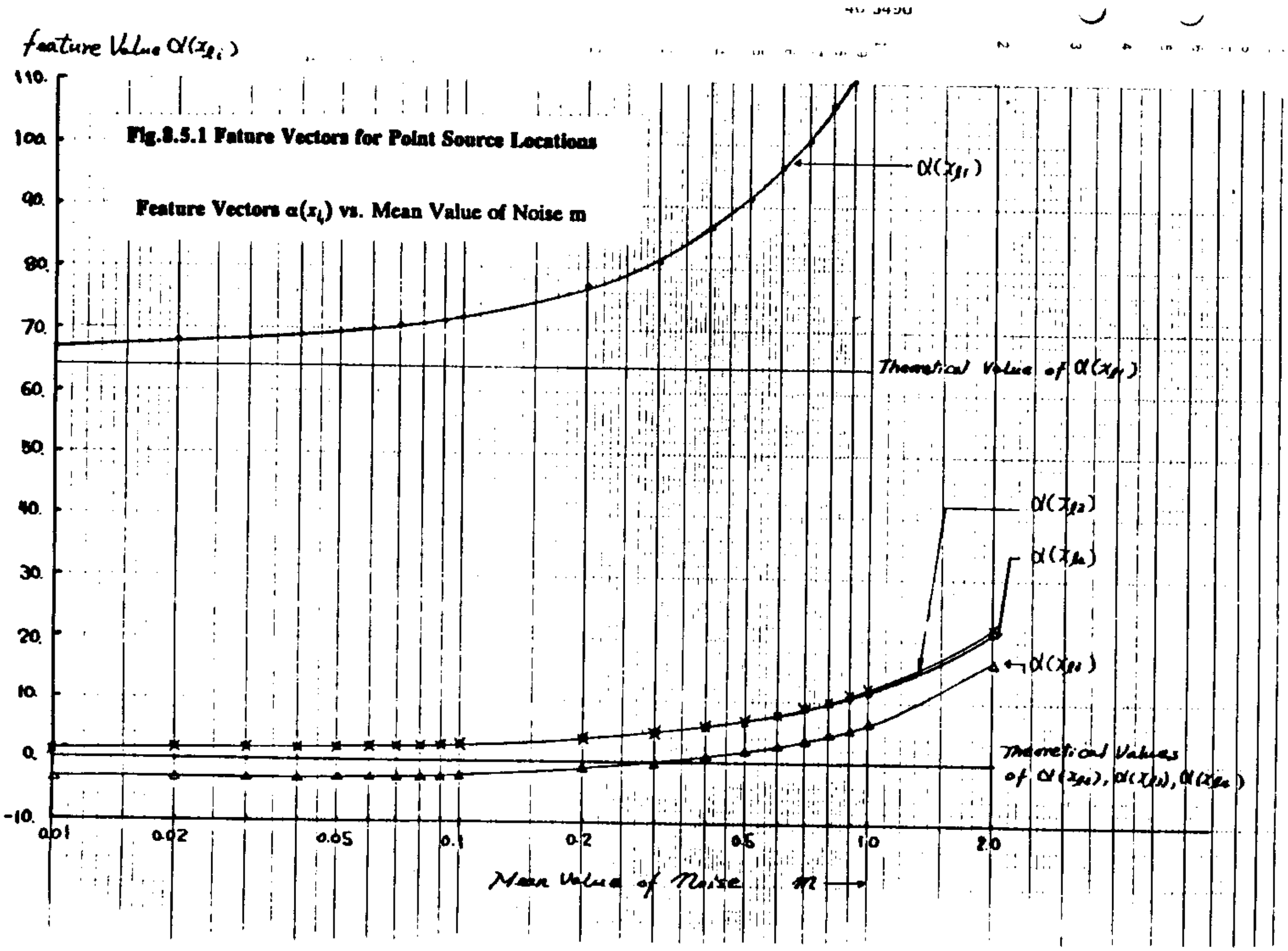


Fig.8.5.1 Fature Vectors for Point Source Locations

Feature Vectors $\alpha(x_i)$ vs. Mean Value of Noise m the following calculation errors :

(1) error due to calculation of inverse matrix $[I-A]^{-1}$

(2) error due to calculation of inverse matrix $[G^TG]^{-1}$

As measurement data include increasing levels of random noise, the feature α_i become somewhat worse than in the noise free case because the summation of noise components do not have zero-mean so that the measurement data have a noise bias. Fig.8.5.1 presents the changes of the feature vectors for each pollution location when the noise bias on the measurement increases. Therefore, this bias must be considered in the discriminant function as a threshold, ϵ. Table.8.5.2 presents the case in which the step size is half of the previous case i.e. $\Delta x = 0.2$. The result is almost same as for the case of $\Delta x = 0.1$. In another words, the accuracy of the feature vector does not improve because the calculations of inverse matrix $[I - A]^{-1}$ introducing error as the dimension of the matrix A is increasing. Table.8.5.3 shows the second case in which two point pollution sources exist at x_{l_1} and x_{l_3}. Obviously, the feature vector can extract the actual pollution source locations even though there are the multiple point sources. It appears that the pattern recognition approach can be used to identify the point source locations even in the presence of high levels of measurement noise.

Once the pollution source locations are determined, their time pattern or time history must be identified. By the dynamic programming, the time pattern ($F(k)=[\,f_1(k),f_2(k)\,]$) can be calculated based on the iterative equations from Eqn.(6.4-29) to (6.4-32). The computational results of $f_1(k),f_2(k)$ are shown in

Table.8.5.4 Point Source Identification from th BOD Measurement Data

Case 1 : Multiple Pollution Sources at x_{acl} = 0.2, 1.2

The number of discrete point N=20 (Δx= 0.1)

Nominated source locations
x_{l_1} =0.2 (l_1 = 2)
x_{l_2} =0.8 (l_2 = 8)
x_{l_3} =1.2 (l_3 = 12)
x_{l_4} =1.6 (l_4 = 16)

Computational Results 3

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	67.5080	64.0000
$\alpha(x_{l2})$		-0.77415	0
$\alpha(x_{l3})$		27.1150	25.0000
$\alpha(x_{l4})$		-0.81807	0
$\alpha(x_{l1})$	0.2	67.1670	64.0000
$\alpha(x_{l2})$		1.6623	0
$\alpha(x_{l3})$		27.9660	25.0000
$\alpha(x_{l4})$		-3.5404	0
$\alpha(x_{l1})$	0.5	66.6570	64.0000
$\alpha(x_{l2})$		5.3171	0
$\alpha(x_{l3})$		29.2420	25.0000
$\alpha(x_{l4})$		-7.6239	0

Fig.8.5.4 and Fig.8.5.5 for different noise levels. When there is no measurement noise, both time functions can be completely determined without error by substituting the weighting matrix R=0. As the measurement noise increases, the identification of both time functions become progressively worse, even though the weighting matrix R is optimized. This is because the time functions $f_1(k)$ and $f_2(k)$ are input function for the system equation, and even large changes in the input do not result large changes in the output. Similarly, even small changes in the output must be the result of large changes in the input. As a result the dynamic programming method for identification of time function is very sensitive to measurement noise. Therefore if the measurement noise level is appreciable, the dynamic programming method is not accurate. However, rough time functions can be identified. Thus multiple pollution sources in river pollution system can be identified by first identifying the pollution source locations by pattern recognition, and then by identifying their time functions by the dynamic programming method.

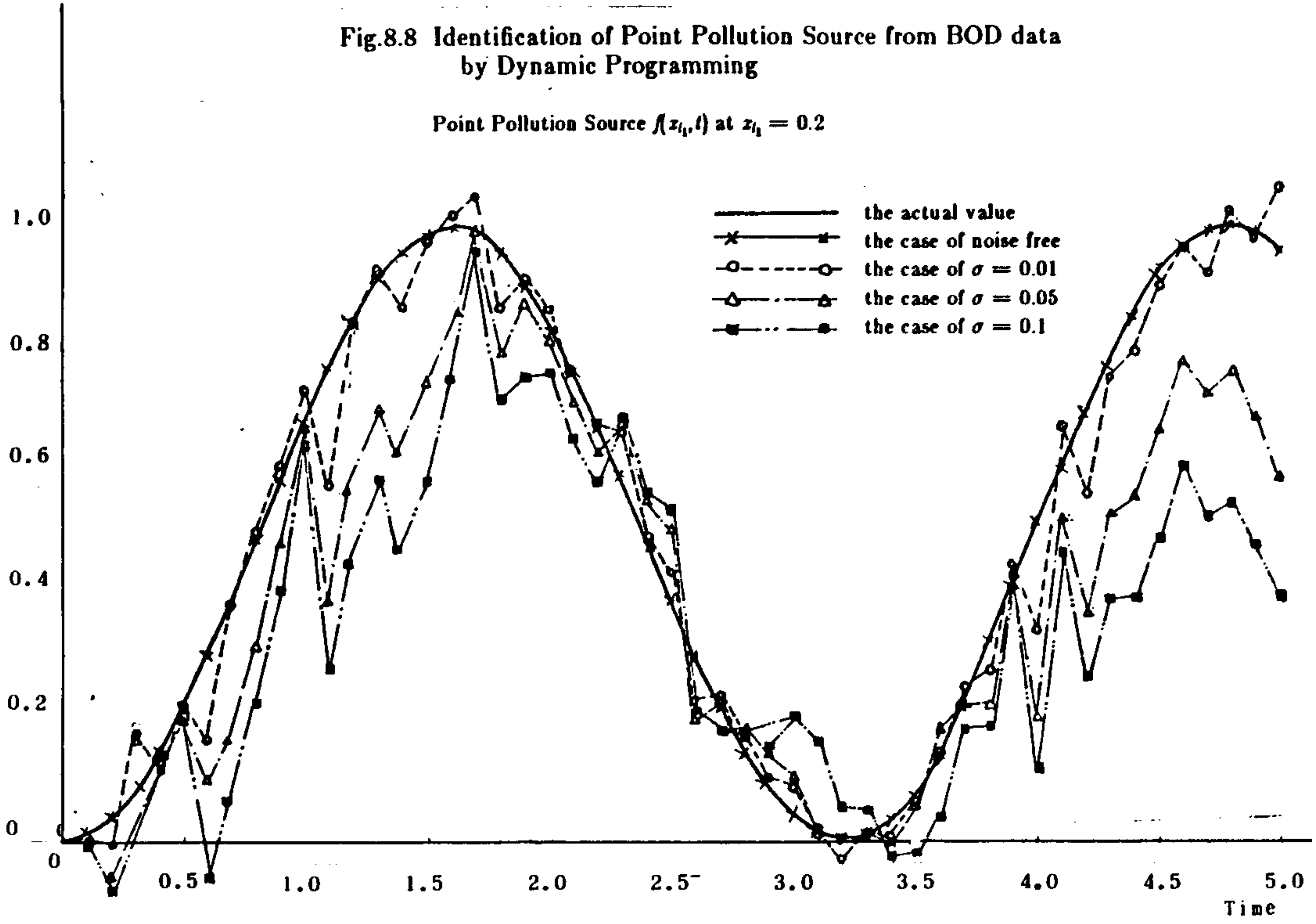
8.6 Example 5

In this section, point pollution sources in river pollution system are identified from DO measurement data. Besides Eqn.(8.5-1), the DO equation is needed to formulate identification procedure as follows :

$$\frac{\partial v}{\partial t} = a\frac{\partial^2 v}{\partial x^2} - b(x)\frac{\partial v}{\partial x} - K_{r(v_s} - v) + K_d u \qquad (8.6\text{-}1)$$

Assume numerical values for the parameters:

Fig.8.8 Identification of Point Pollution Source from BOD data by Dynamic Programming



Point Pollution Source $f(z_{l_1},t)$ at $z_{l_1} = 0.2$

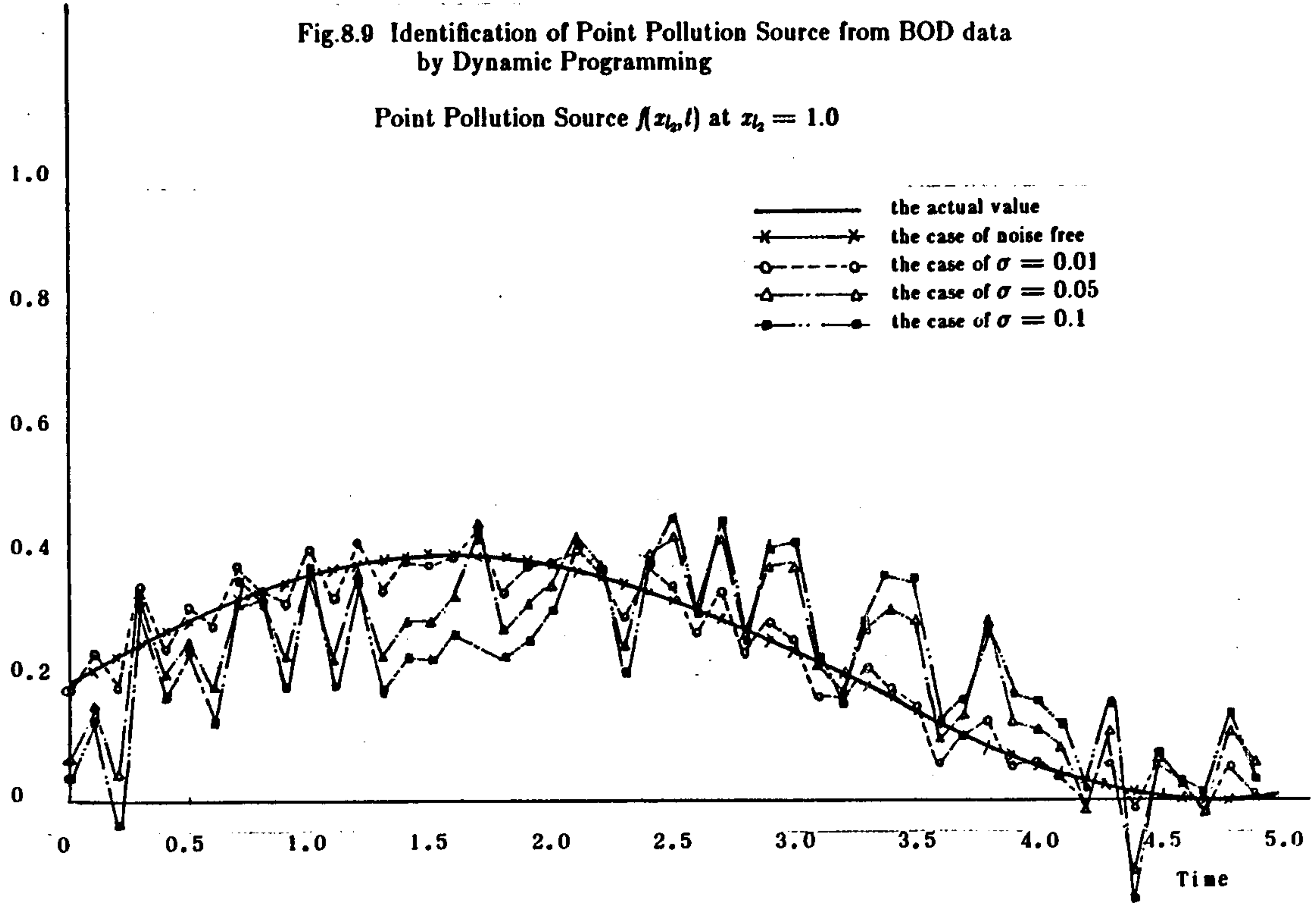


Fig.8.9 Identification of Point Pollution Source from BOD data by Dynamic Programming

Point Pollution Source $f(x_{l_2}, l)$ at $x_{l_2} = 1.0$

$$a=0.01 \tag{8.6-2}$$

$$b=0.25(\, sin(4\pi x/x_f)+1.0\,) \tag{8.6-3}$$

$$c=0.1 \tag{8.6-4}$$

$$K_r=0.1 \tag{8.6-5}$$

$$K_d=0.5 \tag{8.6-6}$$

$$V_s=5.0 \tag{8.6-7}$$

and initial condition

$$v(x,0)=V_s \qquad x_0 \leq x \leq x_f \tag{8.6-8}$$

and boundary conditions

$$v(x_0,t)=V_s \tag{8.6-9}$$

$$u(x_f,t)=V_s \tag{8.6-10}$$

The other conditions are assumed to be the same as in Section 8.5. The measurement data of BOD are obtained at four stations within the region under considerations and involve measurement noise as follows :

$$z(x_{m_1},t_k)=v(x_{m_1},t_k)+e_1(t_k) \quad at\ x_{m_1}=0.4 \tag{8.6-11}$$

$$z(x_{m_2},t_k)=v(x_{m_2},t_k)+e_2(t_k) \quad at\ x_{m_2}=0.8 \tag{8.6-12}$$

$$z(x_{m_3},t_k)=v(x_{m_3},t_k)+e_3(t_k) \quad at\ x_{m_3}=1.2 \tag{8.6-13}$$

$$z(x_{m_4},t_k)=v(x_{m_4},t_k)+e_4(t_k) \quad at\ x_{m_4}=1.4 \tag{8.6-14}$$

where $e_i(t)$ for i=1,4 are measurement noise.

Objective

To identify the distributed pollution sources F(x,t) from the DO measurement data as well as the mathematical model.

First of all, let us identify the point pollution source locations. The transition matrix A' can be calculated from Eqn.(6.3-38), (6.3-39) and (6.4-32). Suppose the nominated pollution locations, x_{l_i} to be :

$$x_{l_1}=0.2,\ \ x_{l_2}=0.8,\ \ x_{l_3}=1.0,\ \ x_{l_4}=1.4 \tag{8.6-15}$$

Therefore, the coefficient matrix B' in Eqn.(6.4-33) for $\Delta x=$ 0.1 is same as the matrix B in the previous section.

$$B' = (b_{i,j}) = \begin{cases} b_{1,2}=b_{2,8}=b_{3,10}=b_{4,14}=1.0 \\ other\ wise\ b_{i,j}=0.0 \end{cases} \tag{8.6-16}$$

Since the measurement locations are

$$x_{m_1}=0.4,\ \ x_{m_2}=0.8,\ \ x_{m_3}=1.2\ \ x_{m_4}=1.4. \tag{8.6-17}$$

Therefore, the output matrix C' in Eqn.(6.4-34) can be expressed as

$$C' = (c_{i,j}) = \begin{cases} c_{1,23}=c_{2,27}=c_{3,29}=c_{4,35}=1.0 \\ other\ wise\ c_{i,j}=0.0 \end{cases} \tag{8.6-18}$$

Also, the measurement data are accumulated from t_0 to t_f as $\sum_{k=0}^{100} W_k$. By using these matrices A',B',C' and $\sum_{k=0}^{100} W_k$, the feature vectors α_i can be calculated as follows :

$$(\alpha_i) = [G'^T G']^{-1} G'^T \sum_{k=0}^{100} W_k \qquad (8.6\text{-}19)$$

where

$$G' = C'[I-A']^{-1}B' \qquad (8.6\text{-}20)$$

The simulation results are shown in Tables 8.6.2. to 8.6.4. The simulation results show that the feature vector α_1 based on the locations x_{l_1} can be used to extract the actual point source location, but accuracy is worse than in the case of BOD. This is due to the fact that the size of the matrix $A', B' C'$ becomes twice as large as in the case of BOD, so that the error in calculating the inverse matrix is increased. Therefore, the identification procedure using DO measurements requires more accurate calculations. The determination of the time functions by dynamic programming is shown in Fig.8.6.3 and Fig.8.6.4. In this case, the identification is also very sensitive to measurement noise as is true in the case of BOD. Therefore, the measurement data should be observed with minimum noise. As a result, identification of point pollution sources from DO measurement data is more difficult than from the BOD.

8.7 Example 6

In this section, point pollution sources in aquifer pollution systems are identified from the measurement data. The aquifer under study is assumed to exist near the ground surface where the pollutants are injected from factories. The locations and intensities of the pollution sources are assumed to be unknown. The pollution system is characterized by quantity and quality models. The

Table.8.6.2 Point Source Identification from the DO Measurement Data

Case 2 : Single Pollution Source at $x_{act} = 0.2$

The number of discrete point N=20 (Δx= 0.1)

Nominated source locations
$x_{l_1} = 0.2$ ($l_1 = 2$)
$x_{l_2} = 0.8$ ($l_2 = 8$)
$x_{l_3} = 1.2$ ($l_3 = 12$)
$x_{l_4} = 1.6$ ($l_4 = 16$)

Computational Results 1

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	61.1090	64.0000
$\alpha(x_{l2})$		0.94530	0
$\alpha(x_{l3})$		-4.0085	0
$\alpha(x_{l4})$		-.27263	0
$\alpha(x_{l1})$	0.2	59.2410	64.0000
$\alpha(x_{l2})$		1.8768	0
$\alpha(x_{l3})$		-5.0628	0
$\alpha(x_{l4})$		-6.9707	0
$\alpha(x_{l1})$	0.5	56.4380	64.0000
$\alpha(x_{l2})$		3.2741	0
$\alpha(x_{l3})$		-1.8670	0
$\alpha(x_{l4})$		-2.1516	0

Table.8.6.3 Point Source Identification from the DO Measurement Data

Case 2 : Multiple Pollution Sources at x_{act} = 0.2 ,1.2

The number of discrete point N=20 (Δx= 0.1)

Nominated source locations
x_{l_1} =0.2 (l_1 = 2)
x_{l_2} =0.8 (l_2 = 8)
x_{l_3} =1.2 (l_3 = 12)
x_{l_4} =1.6 (l_4 = 16)

Computational Results 2

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	61.20300	64.0000
$\alpha(x_{l2})$		6.3393	0
$\alpha(x_{l3})$		34.7370	25.0000
$\alpha(x_{l4})$		0.17145	0
$\alpha(x_{l1})$	0.2	59.3350	64.0000
$\alpha(x_{l2})$		-2.7190	0
$\alpha(x_{l3})$		44.0520	25.0000
$\alpha(x_{l4})$		-0.79824	0
$\alpha(x_{l1})$	0.5	56.5320	64.0000
$\alpha(x_{l2})$		-16.339	0
$\alpha(x_{l3})$		58.0280	25.0000
$\alpha(x_{l4})$		-2.2352	0

Fig.8.10 Identification of Point Pollution Source from DO Data by Dynamic Programming

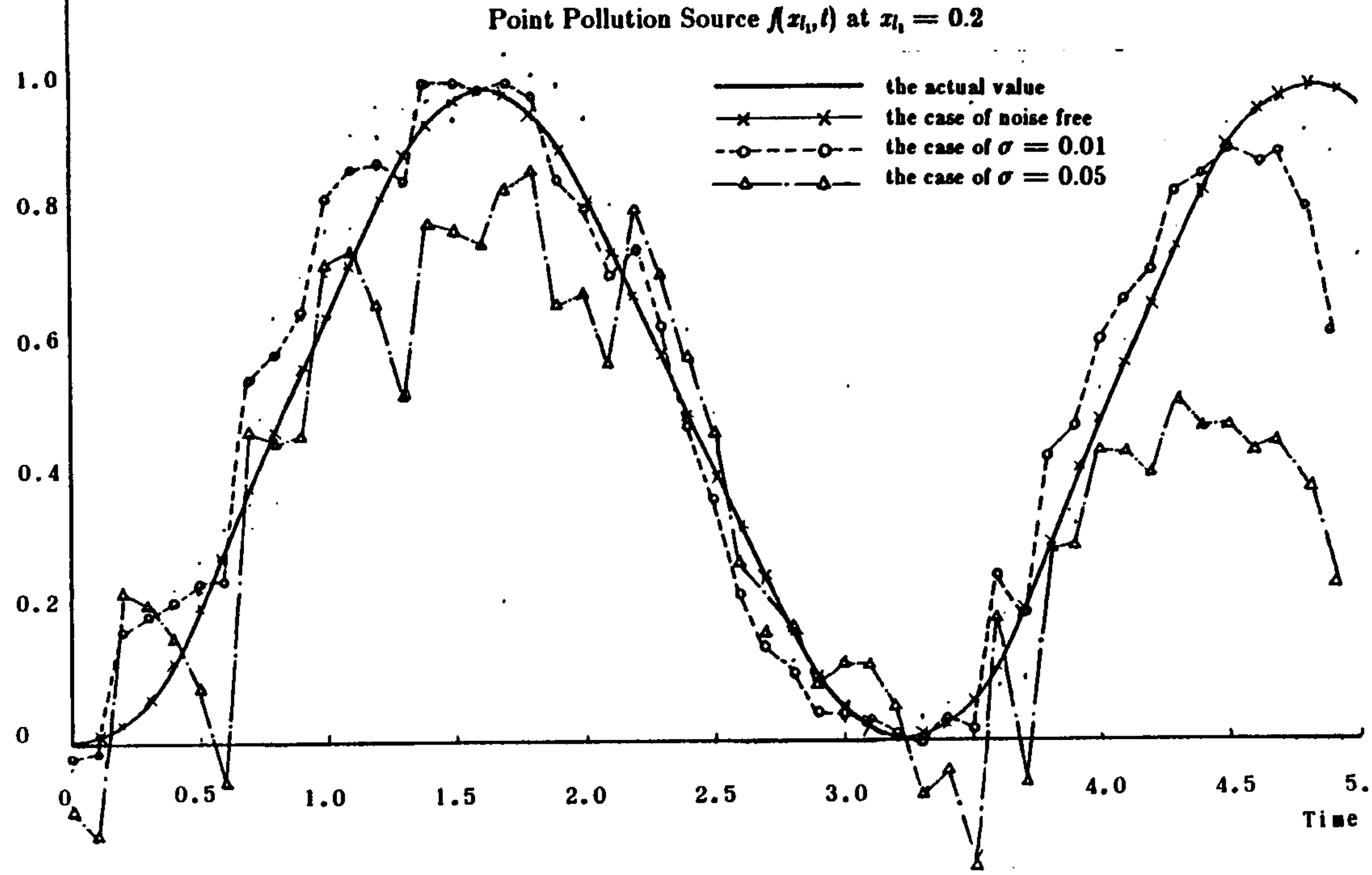
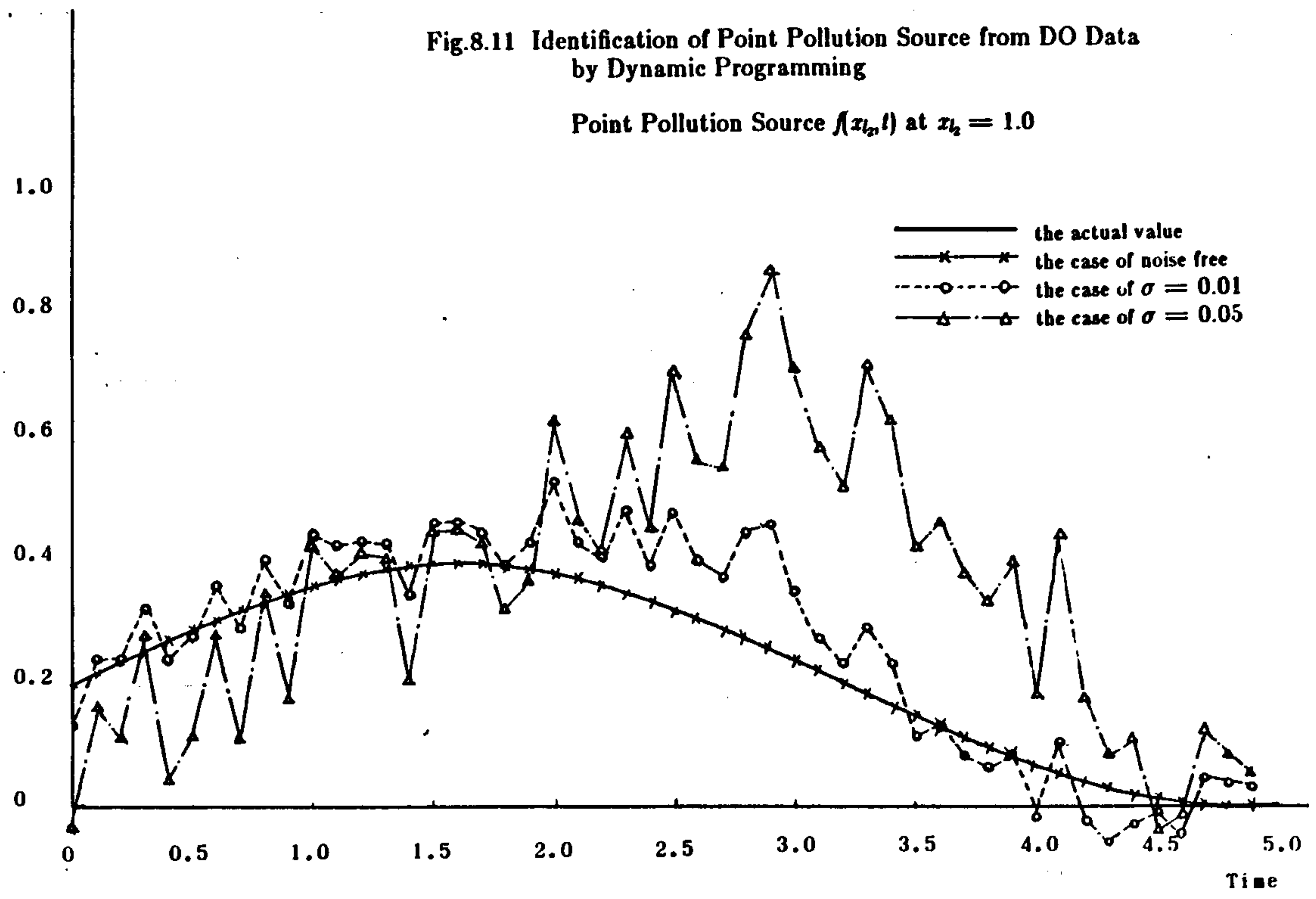


Fig.8.11 Identification of Point Pollution Source from DO Data by Dynamic Programming

Point Pollution Source $f(x_{l_z}, t)$ at $x_{l_z} = 1.0$



aquifer flow equation is

$$\frac{\partial}{\partial x}\left(K(x)\frac{\partial h}{\partial x}\right) = S\frac{\partial h}{\partial t} + Q \tag{8.7-1}$$

where h(x,t) is the hydraulic head, K(x) is the conductivity (assumed to be a function of space), S is the specific storage (assumed to be constant), Q is volume flow (assumed to be constant). The aquifer quality system is expressed by

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}\left(D(x,t)\frac{\partial c}{\partial x}\right) - \frac{\partial}{\partial x}\left(v(x,t)c\right) + \frac{cQ}{n} + F(x,t) \tag{8.7-2}$$

where c(x,t) is material concentration, D(x,t) is the diffusion coefficient (assumed to be a function of space and time), n is porosity (assumed to be a constant), and F(x,t) is the pollution source to be identified. The Darcy equation is

$$v(x,t) = \frac{K(x)}{n}\frac{\partial h}{\partial x} \tag{8.7-3}$$

Assume the numerical values for these parameters

$$K(x) = 31000.0 - 2.25x \tag{8.7-4}$$

$$S = 0.015 \tag{8.7-5}$$

$$Q = 500.0 \tag{8.7-6}$$

and initial condition

$$h(x,0) = 200.0 - \frac{300}{0.25}\log_e\left(\frac{31000}{31000-2.25x}\right)$$

$$(8.7\text{-}7)$$

$$c(x,0) = 0.0 \tag{8.7-8}$$

and boundary conditions

$$h(0,t) = 200.0 \tag{8.7-9}$$

$$\frac{\partial h(x_f,t)}{\partial x} = 0 \tag{8.7-10}$$

$$c(0,t) = 0.0 \tag{8.7-11}$$

$$\frac{\partial c(x_f,t)}{\partial x} = 0 \tag{8.7-12}$$

where the region is

$$0 \le x \le 10000.0 \quad 0 \le x \le 2000.0 \tag{8.7-13}$$

The measurement data are obtained at four stations, within the boundary of the region under consideration, and involve measurement noise as follows :

$$z(x_{m1},t_k) = c(x_{m1},t_k) + e_1(t_k) \;\; at \; x_{m1}=1000.0 \tag{8.7-14}$$

$$z(x_{m2},t_k) = c(x_{m2},t_k) + e_2(t_k) \;\; at \; x_{m2}=3000.0 \tag{8.7-15}$$

$$z(x_{m3},t_k) = c(x_{m3},t_k) + e_3(t_k) \;\; at \; x_{m3}=5000.0 \tag{8.7-16}$$

$$z(x_{m4},t_k) = c(x_{m4},t_k) + e_4(t_k) \;\; at \; x_{m4}=7000.0 \tag{8.7-17}$$

where $e_i(t)$ for i=1,4 are measurement noise.

Objective

To identify the distributed pollution sources F(x,t) from measurement data. The entire identification procedure is indicated as in Section 6.4.3. For numerical computations the following space step size and time step size are specified:

$$\Delta x = 500.0 \tag{8.7-18}$$

$$\Delta t = 20.0 \tag{8.7-19}$$

To solve these partial differential equations, the explicit finite difference approximation method method is employed. For simulation, the following pollution sources were assumed as actual sources to obtain the output data

$$F(x,t) = \delta(x-2000.0)\sin^2(\pi t/320) \tag{8.7-20}$$

Then random gaussian noise signals with 0-mean and different standard deviation where added to the output data to generate the "measurement data". Let us first identify the point pollution source locations. Suppose the nominated pollution locations, x_{l_i} are :

$$x_{l_1}=2000,\ x_{l_2}=4000,\ x_{l_3}=6000,\ x_{l_4}=8000 \tag{8.7-21}$$

Therefore, the coefficient matrix B in Eqn.(6.4-5) for $\Delta x=500$ is

$$B = (b_{i,j}) = \begin{cases} b_{1,4} = b_{2,8} = b_{3,12} = b_{4,16} = 1.0 \\ otherwise\ b_{i,j} = 0.0 \end{cases} \tag{8.7-22}$$

Since the measurement locations are

$$x_{m_1}=1000,\ x_{m_2}=3000,\ x_{m_3}=5000\ \ x_{m_4}=7000. \tag{8.7-23}$$

Therefore, the output matrix C in Eqn.(6.4-7) can be expressed as

$$C = (c_{i,j}) = \begin{cases} c_{1,2} = c_{2,6} = c_{3,10} = c_{4,14} = 1.0 \\ other\ wisec_{i,j} = 0.0 \end{cases} \tag{8.7-23}$$

The measurement data are accumulated from t_0 to t_f as $\sum_{k=0}^{100} Z_k$. By using the matrices A,B,C and $\sum_{k=0}^{100} Z_k$, the feature vectors α_i can be calculated as follows :

$$(\alpha_i) = [G^T G]^{-1} G^T \sum_{k=0}^{100} Z_k \tag{8.7-24}$$

where

$$G = C[I - A]^{-1} B \tag{8.7-25}$$

The simulation results are shown in Table.8.7.2. and Table.8.7.3. The results show the feature vector based on the location x_{i_2} can extract the actual point source location, increasing levels of measurement noise, but the accuracy is worse than in the case of river pollution system. This is due to the fact that the transition matrix A is time variant so that an averaged transition matrix, $\bar{A}$ was employed instead of A to derive the feature vector α_i. Therefore, the accuracy of the entire identification depends on the aquifer flow dynamics. The determination of the time functions by dynamic programming is shown in Fig.8.7.1. The time function of the pollution sources can be identified in the noise free case, even though the pollution system equation is time variant, but the solution is very sensitive to measurement noise.

Table.8.7.2 Point Source Identification of Aquifer Pollution System

Case 1 : Single Pollution Source at x_{act} = 4000.0

The number of discrete point N=20 (Δx= 500.0)

Nominated source locations
x_{l_1} =2000.0 (l_1 = 2)
x_{l_2} =4000.0 (l_2 = 8)
x_{l_3} =6000.0 (l_3 = 12)
x_{l_4} =8000.0 (l_4 = 16)

Computational Results 1

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	-0.00474	0
$\alpha(x_{l2})$		11.18500	9.6577
$\alpha(x_{l3})$		0.24307	0
$\alpha(x_{l4})$		-0.10745	0
$\alpha(x_{l1})$	0.2	0.77603	0
$\alpha(x_{l2})$		10.27800	9.6577
$\alpha(x_{l3})$		0.60630	0
$\alpha(x_{l4})$		-0.16558	0
$\alpha(x_{l1})$	0.5	1.94721	0
$\alpha(x_{l2})$		8.91730	9.6577
$\alpha(x_{l3})$		1.15120	0
$\alpha(x_{l4})$		-0.25276	0

Table.8.7.3 Point Source Identification of Aquifer Pollution System

Case 1 : Multiple Pollution Sources at x_{act} = 2000.0, 6000.0

The number of discrete point N=20 (Δx= 500.0)

Nominated source locations
x_{l_1} =2000.0 (l_1 = 2)
x_{l_2} =4000.0 (l_2 = 8)
x_{l_3} =6000.0 (l_3 = 12)
x_{l_4} =8000.0 (l_4 = 16)

Computational Results

feature vector	noise level	calculated value	theoretical vaule
$\alpha(x_{l1})$	0	10.16600	9.6557
$\alpha(x_{l2})$		0.22029	0
$\alpha(x_{l3})$		12.17000	11.9390
$\alpha(x_{l4})$		1.79810	0
$\alpha(x_{l1})$	0.2	10.94700	9.6577
$\alpha(x_{l2})$		-0.68678	0
$\alpha(x_{l3})$		12.53300	11.9390
$\alpha(x_{l4})$		-1.74000	0
$\alpha(x_{l1})$	0.5	12.11800	9.6557
$\alpha(x_{l2})$		-2.04740	0.0
$\alpha(x_{l3})$		13.07800	11.9390
$\alpha(x_{l4})$		1.65280	0

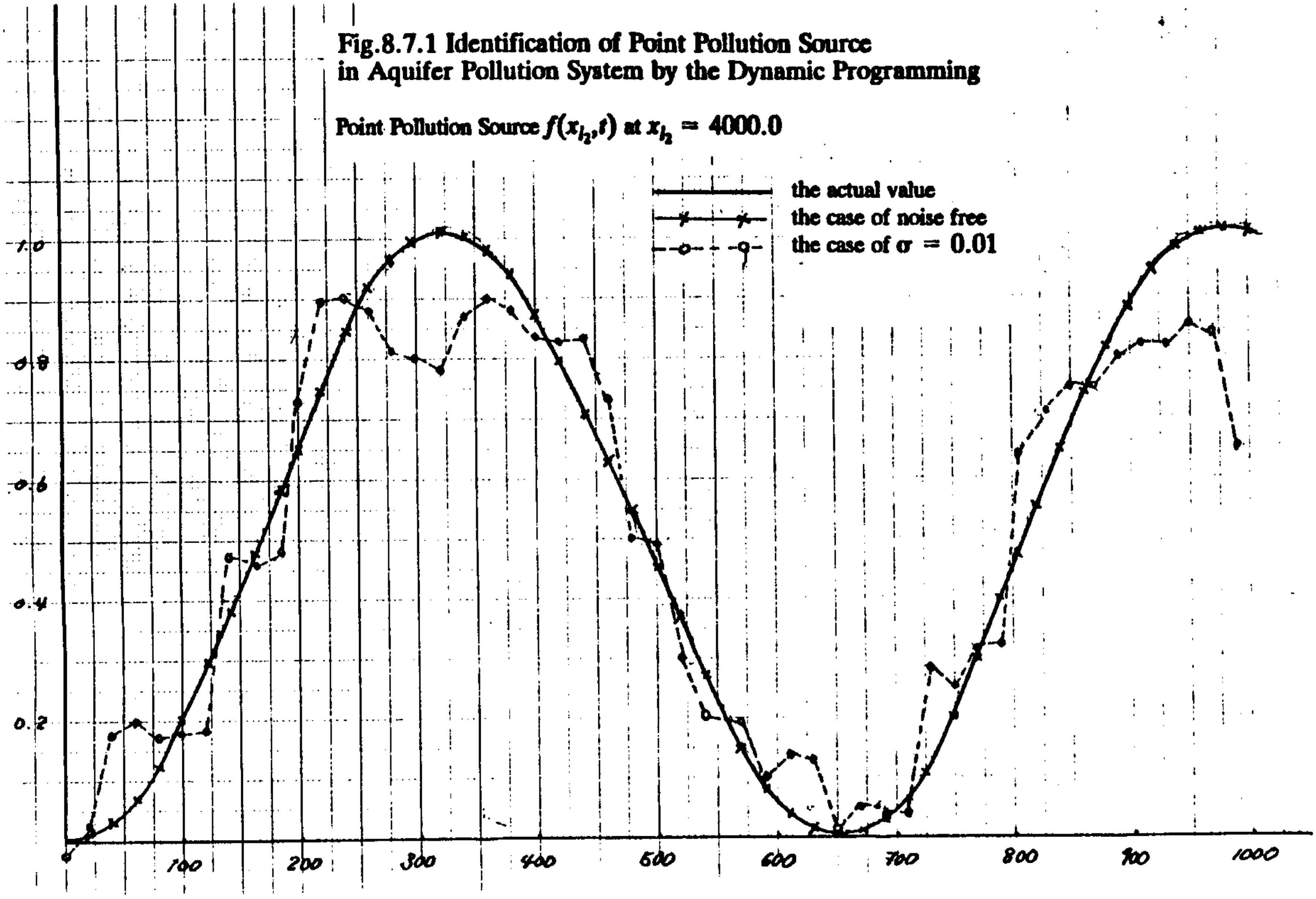


Fig.8.7.1 Identification of Point Pollution Source in Aquifer Pollution System by the Dynamic Programming

Point Pollution Source $f(x_{l_2},t)$ at x_{l_2} = 4000.0

CHAPTER 9

Conclusions

9.1 Scope and Objectives

The research described in this dissertation is directed to the development of a new technique for the identification of input functions in distributed parameter systems and more specifically of pollution sources in water pollution systems. Two major challenging problem areas, river pollution systems and aquifer pollution systems are discussed as examples. In general, there are major difficulties involved in the identification of distributed pollution sources in environmental systems. The followings are the principal sources of difficulty:

(1) Distributed pollution sources to be identified generally vary in space and time.

(2) The general nature of pollution sources are at best only approximately known, therefore, they must be inferred using physical insight.

(3) Since environmental pollution systems are generally modeled by partial differential equations (PDE's), a large amount of computations are involved. Therefore, the selection of computing hardware facilities as well as algorithms and software must be carefully considered.

(4) From the mathematical point of view, pollution source identification

problems in environmental systems are equivalent to the determination of input functions in distributed parameter systems, and are generally non-well posed inverse problems.

(5) The system observations involve measurements of certain system variables at a relatively small number of measuring stations and relatively infrequently in time, i.e. the observational data are generally very sparse.

(6) The observations are subject to substantial measurement error.

In order to cope with the above difficulties, the mathematical models of both river pollution systems and aquifer pollution systems and their pollution source identification problems are first precisely formulated in Chapter 2. Among these formulations, the pollution sources to be identified are classified into the point pollution source type and the distributed pollution source type using separable functions of space and time variables.

In Chapter 3, an investigation of the possibilities of the conventional identification methods for the identification problems formulated in Chapter 2 is made. Several typical approaches including the regularization method, the cross-validation, the least square method, the Kalman filtering and the dynamic programming, are employed for the identification of distributed pollution sources in rivers which are characterized by the one-dimensional partial differential equation. For each approach, the difficulties and restrictions on the execution of the entire identification procedure are analyzed.

In order to complement the conventional identification methods, rather than to replace them, the pattern recognition approach, involving pattern generation, feature extraction, classification and evaluation ~~introduced~~ introduced in Chapter 4. In the pattern generation step, which entails the partitioning of the general pollution source into subsets, the time pattern and the pollution source locations can be regarded as the generated subclass pollution pattern. As feature vectors, which enhance the similarities of elements of the same class while enhancing the differences of the elements of classes, the coherence functions and the normalized correlation functions are employed. In the classification steps, a linear discriminant function which consists of an equal weight vector is employed for simplicity. Performance evaluation is carried out with the aid of simulations.

The entire identification process for river pollution systems with both BOD and DO index and aquifer pollution systems, which are classified into the distributed pollution sources and point pollution sources, is formulated. in Chapter 6. In the case of the distributed pollution sources, first, the time pattern of the pollution source is identified by pattern recognition. Then the space function is identified by conventional identification, specifically, the least square method. In the case of point pollution sources, the pollution source locations are identified by the pattern recognition approach. The time functions are then identified by dynamic programming.

In order to improve the computational speed of the entire identification procedure, the Monte Carlo method is used as an alternatives of numerical method for the solution of the partial differential equations, and the peripheral array processor system is introduced as a new implementation tool of the entire identification procedure.

In Chapter 5, the computational algorithms of the Monte Carlo method for the one-dimensional river pollution systems with both BOD and DO and for two-dimensional aquifer pollution systems are formulated. The several numerical examples are employed to compare with the computational speed and accuracy of solutions by the finite difference equation and by the Monte Carlo method.

In Chapter 7, the implementations of the entire identification procedure using the peripheral array processor, which is connected to the general purpose computer, are precisely formulated. Performance evaluations of the solution of the partial differential equations, the calculation of the feature vectors, the calculation of the conventional identification methods of the identification process, are performed using the VAX-11/750 and the Mini-Map array processor.

In chapter 8, in order to demonstrate the verification of the entire identification procedure formulated in Chapter 9, several numerical examples including one-dimensional river pollution systems and two-dimensional aquifer pollution systems are analyzed with the aid of simulations. Evaluations of the identification procedure for both distributed pollution sources identification and point pollution source identification are made by varying the noise level of the measurement data.

9.2 Principal Conclusions

The main purpose of the research described in this dissertation is to demonstrate how pollution sources can be identified using a methodology based on the pattern recognition approach which was introduced by Karplus [Karp 72], and how the entire identification procedure can be executed efficiently aid of array processors. There are already several studies in which the pattern recognition techniques have been found to be very useful in the mathematical modeling aquifer quantity and quality systems for which sufficient information is not available to implement conventional identification methods. As another application area of the pattern recognition, the problem of identifying the pollution sources in water pollution systems is investigated in our research.

The introduction of a separable function of space and time for the pollution source, not only has a physical justification, but also makes it possible to classify identification problems into distributed pollution source and point source identification. This is convenient for the entire identification process to be implemented precisely because the time function and the space function can be determined by combining the pattern recognition approach and conventional methods. However, in the case in which the pollution sources are moving or for multiple distributed source types, separability cannot be directly applied. Therefore more general structures should be assumed for the pollution sources

Using Fourier series to express the time function has a big advantage in that the subclasses of the pollution time pattern can be automatically generated, so that the pattern recognition process automatically approaches the actual time pattern progressively as the number of Fourier series terms is increased. However, in the case that the time function of the pollution source behaves in a random manner (i.e. there is no periodic time pattern) the pattern recognition approach is relatively ineffective.

Coherence functions and normalized correlation functions as feature vectors have significant attraction in that their feature extraction abilities are inherent even in the presence of high measurement noise. Furthermore, their computations are highly computer intensive so that FFT and signal processing techniques can be used. It should be noted that relatively large number of measurement are required to obtain statistical data for both the coherence function in the presence of measuring noise.

In point pollution source identification, the locations of the pollution sources can be extracted roughly from a priori nominated pollution sources by the pattern recognition approach even when the measurement data are corrupted by high noise level (i.e., large noise variance) with zero mean. The calculated feature vectors are significant because their values represent directly the total amount of pollution discharged from the nominated pollution locations during a measuring period, no matter how the time functions of these sources behave.

The conventional identification methods are still useful and can be employed after the determination of either the time function or the space function by the pattern recognition method.

The computer algorithm for the entire pattern recognition approach including the solution of the partial differential equations and the tuning of the parameters, a_i and ϕ_i of the generated subclass patterns, are highly computation intensive. Array processor systems are particularly attractive in that the computation time can be improved with one order of magnitude over the host machine as the number of the space steps is increased. Furthermore, the identification process can be programmed and executed easily by using the scientific subroutines which are supplied by the manufacture.

The Monte Carlo method for the solution of partial differential equations is useful to obtain the rough solutions at several specific locations with relatively high speed.

The entire identification process formulated in this research can be applied to the other problems, such as, lake pollution systems, bay pollution systems,or air pollution systems as long as their mathematical models are well formulated and measurement data are available.

9.3 Specific Accomplishments

The accomplishments of the research described in this dissertation are summarized as follows :

(1) An investigation of the mathematical modeling of river pollution systems and aquifer pollution systems and the identification of pollution sources from measurement data, has been conducted.

(2) It has been shown that the conventional identification methods have weakness and difficulties in coping with identification of pollution sources from the measurement data.

(3) The pattern recognition approach was introduced to overcome some of the difficulties of conventional identification methods.

(4) Various methods for employing feature vectors and pattern recognition to extract original pollution source pattern were shown. The correlation function and the coherence function are particularly useful as feature vectors because of their invariance in the presence of additive measurement noise.

(5) It has been shown that feature extraction from power spectra functions or coherence functions has the advantages that the time function of the original pollution source pattern can be used directly to obtain the frequencies which are partitioned into subclasses.

(6) It has been shown that the Monte Carlo method is suitable for solving the partial differential equations, because solutions for only a few points (corresponding to the measuring stations) are required.

(7) The entire identification procedure has been demonstrated by using a variety of combinations of the pattern recognition and of conventional identification methods.

(8) Performance evaluations of the entire identification process have been made for both the simulated river pollution systems and the simulated aquifer pollution systems.

(9) It has been shown that array processors can carry out the entire identification procedure at high speed and at relatively modest cost when combined with conventional computers.

9.4 Suggestions for Future Research

The main purpose of the research discussed in this dissertation was to develop a methodology and implementations to overcome the difficulties with conventional methods summarized in Chapter 1. According to the results of the implementations and performance evaluations of the proposed identification procedure, this goal has been achieved. In order to improve the proposed identification procedure, the following topics should be investigated.

9.4.1 High-Order Finite Difference Approximation

In order to approximate the original partial differential equations, we employed the second central difference to approximate for second order partial derivative terms. As a result, the transition matrix, A, for the state vector equations is tridiagonal. In order to obtain a higher accuracy in the feature calculation and conventional identification and in order to reduce the truncation error, higher-order finite difference approximation may be desirable particularly for the case in which there are multiple-point pollution sources and the distances between them are short, or the case in which there are multiple measuring stations which are not located homogeneously and some of the stations are close to each other. In these cases, point pollution source identification is difficult to carry out because the calculation of inverse matrix, $[I - A]^{-1}$ is computationally diverging. Therefore, a high-order approximation is desirable. By employing high-order approximations, the following advantages can be expected :

(1) Numerical solutions of the mathematical equations will be more accurate,

(2) The accuracy of the conventional identification can be improved; as a result, more accurate source identification can be possible.

On the other hand,

(1) the calculation time increases when high-order finite approximations are employed.

9.4.2 Identifications with unknown field parameters

In our research, all the field parameters were assumed to be known. However, in many practical situations, pollution source identification must be executed with unknown field parameters. This makes the problem more difficult because :

(1) the input pollution sources are unknown.

(2) The mathematical model is partly unknown.

(3) The measurement data available are corrupted by measurement noise.

Therefore, there is no deterministic information available. For this case, the field parameters must first be identified by injecting tracers from particular upstream location. Then the pollution source can be identified from the measurement data.

9.4.3 Pollution Source Identification with Unknown Initial Conditions and Boundary Conditions

In the example included in Chapter 8, pollution source identification was demonstrated with known initial and boundary conditions. In some practical situations, these conditions might not be known so that the pollution source identification must be implemented with unknown initial and boundary conditions. Perhaps the initial state can be represented approximately by measurement data near the boundary area. Another possible way is to identify both the initial and the boundary conditions by regarding both conditions as pollution sourccs.

9.4.4 Optimal Measurement Locations for Identification

In our research, the number of measurement stations was assumed to be fixed in advance. From the identification view point it is desirable to locate as many of the measurement stations as possible near the pollution sources so as to attain a greater accuracy in the entire identification procedure. However, from the economic point of view, measuring stations could be very costly so that the number of the measuring stations should be minimized. Therefore, the locations of measuring stations should be optimized so as to obtain optimally accurate identifications. It is thus necessary to evaluate the locations of the measurement stations. One of possibility for evaluation is to consider a sensitivity function such as :

$$S(x_{m_i}) = \frac{\partial}{\partial F}\int_0^t (u_m (x_{m_i},t) - u(x_{m_i},t))^2 dt \qquad (9.4\text{-}1)$$

so that the optimal measuring stations can be found by maximizing the sensitivity function.

9.4.5 Syntactic Pattern Recognition Approaches

The pattern recognition method which was introduced in this research is a statistical approach. Alternatively, syntactic pattern recognition approaches are possible. This is done by constructing the feature vector from the difference between the two succeeding measurement data. This method has already been employed for the prediction of the aquifer quantity in the annual volume problem [UmUR78] [UmUn80].

APPENDIX A

The Least Square Method

Problem statement

Given the systm equation with initial condition U_0

$$U_{k+1} = AU_k + Gf_k \tag{A-1}$$

the observation equation

$$Y_k = CU_k \tag{A-2}$$

and actual measurement data Z_k such that

$$Z_k = Y_k + W_k \tag{A-3}$$

find the optimal estimate value $\hat{G}$ to minimize the error function J such that

$$J = \sum_{k=1}^{n} (Z_k - CU_k)^T (Z_k - CU_k) \tag{A-3-1}$$

At first, the system equation (A-1) is transformed into the equation which includes Y_k instesd of U_k. When k=1,

$$Y_1 = CU_1 = C(AU_0 + Gf_0) \tag{A-4}$$

When k=2,

$$Y_2 = CU_2 = C(AU_1 + Gf_1) = CA^2 + CAGf_0 + CGf_1 \tag{A-5}$$

. When k=k-1,

$$Y_k = CU_k = C(AU_{k-1}Gf_{k-1})$$

$$= CA_kU_0+CA^{k-1}Gf_0+A^{k-2}Gf_1+ \cdots +CAGf_{k-2}+CGf_{k-1} \tag{A-7}$$

Therefore,

$$Y_k = (CA^{k-1}f_0+CA^{k-2}f_1+ \cdots +CAf_{k-2}+Cf_{k-1})G$$

$$= X_kG \tag{A-8}$$

where

$$X_k = CA^{k-1}f_0+CAk-2+ \cdots +CAf_{k-2}+Cf_{k-1} \tag{A-9}$$

Using the eqn (A-9), the error function J can be written as

$$J = \sum_{k=1}^{n}(Z_k-Y_k)^T(Z_k-Y_k) \tag{A-10}$$

$$= \sum_{k=1}^{n}(Z_k-X_kG)^T(Z_z-X_kG) \tag{A-11}$$

In order to find the optimal estimate value of $\hat{G}(n)$, that minimize the error function J,

$$\frac{\partial J}{\partial G} = 0 \quad at\ \hat{G}(n) \tag{A-12}$$

must satisfied. From this condition, we have

$$\frac{\partial J}{\partial G} = \sum_{k=1}^{n}\{-2X_k^TZ_k+2X_k^TX_k\hat{G}(n)\} = 0 \tag{A-13}$$

That is,

$$(\sum_{k=1}^{n} X_k^T X_k)\hat{G}(n) = \sum_{k=1}^{n} (X_k^T Z_k) \tag{A-14}$$

Moreover, we define that

$$P(n) = [\sum_{k=1}^{n} X_k^T X_k]^{-1} \tag{A-15}$$

and

$$Q(n) = \sum_{k=1}^{n} X_k^T Z_k \tag{A-16}$$

Therefore, the optimal estimate value $\hat{G}(n)$ can be expressed as

$$\hat{G}(n) = P(n)Q(n) \tag{A-17}$$

At the same procedure, for k=n+1

$$\hat{G}(n+1) = P(n+1)Q(n+1) \tag{A-18}$$

Then we consider the P(n+1). By definition of the (A-15),

$$P^{-1}(n+1) = \sum_{k=1}^{n+1} X_k^T X_k \tag{A-19}$$

$$= \sum_{k=1}^{n} X_k^T X_k + X_{n+1}^T X_{n+1} \tag{A-20}$$

$$= P^{-1}(n) + X^T(n+1)X(n+1) \tag{A-21}$$

By inversing the (A-21), we can obtain

$$P(n+1) = P(n) - P(n)X^T(n+1)[X(n+1)P(n)X^T(n+1)+I]^{-1}X(n+1)P(n) \tag{A-22}$$

or

$$= P(n)\ P(n)X^T(n+1)K(n)X(n+1)P(n) \tag{A-23}$$

Where we define

$$K(n) = [X(n+1)P(n)X^T(n+1)+I]^{-1} \tag{A-24}$$

and I is identity matrix.

Next, we consider the $\hat{G}(n+1)$. By definition of the (A-18),

$$\hat{G}(n+1) = P(n+1)Q(n+1) \tag{A-25}$$

$$= P(n+1)\{Q(n)+X^T(n+1)Z(n+1)\} \tag{A-26}$$

By substituting the (A-23) into the (A-26), we can obtain

$$\hat{G}(n+1) = (P(n)$$

$$\hat{G}(n+1) = \{P(n)-P(n)X^T(n+1)K(n)X(n+1)P(n)\}\{Q(n)+X^T(n+1)Z(n+1\} \tag{A-27}$$

$$= P(n)Q(n)+P(n)X^T(n+1)Z(n+1)$$

$$-P(n)X^T(n+1)K(n)X(n+1)P(n)(Q(n)+X^T(n+1)Z(n+1)) \tag{A-28}$$

$$= \hat{G}(n)+P(n)X^T(n+1)K^{-1}(n)Z(n+1) \tag{A-29}$$

$$-P(n)X^TK(n)x(n+1)P(n)(Q(n)+X^T(n+1)Z(n+1))$$

$$\hat{G}(n)+P(n)X^T(n+1)K(n)(X(n+1)P(n)X^T(n+1)+I)Z(n+1)$$

$$-P(n)X^T(n+1)K(n)X(n+1)P(n)(Q(n)+X^T(n+1)Z(n+1)) \tag{A-30}$$

$$= \hat{G}(n)+P(n)X^T(n+1)K(n+1)Z(n+1) \tag{A-31}$$

$$-P(n)X^T(n+1)K(n)X(n+1)P(n)Q(n)$$

$$= \hat{G}(n)+P(n)X^T(n+1)K(n)(Z(\hat{n}+1)-X(n+1)G(n)) \tag{A-32}$$

As a consequence, we could derive the Least-Square algorithm to identify the otimal estimate of $\hat{G}(n+1)$ aas follows ;

$$\hat{G}(n+1) = G(n)+P(n)X^T(n+1)K(n)(Z(\hat{n}+1)-X(n+1)G(N)) \tag{A-33}$$

$$p(n+1) = P(n)P(n)X^TX(n+1)K(n)X(n+1)P(n) \tag{A-34}$$

$$K(n) = [X(n+1)P(n)X^T(n+1)+I]^{-1} \tag{A-35}$$

Those equation can be calcuated b giving th initial estimate values $\hat{G}(0)$ and P(0).

APPENDIX B

The Dynamic Programming

Let us be the model equation

$$U_{k+1} = A_k U_k + B_k F_k, \tag{B-1}$$

the output equation

$$Y_k = C_k U_k \tag{B-2}$$

the observation equation

$$Z_k = Y_k + n_k, \tag{B-3}$$

then find the optimal input function F^* to minimize the cost function such that

$$J = \frac{1}{2}(Z_k - CU_k)^T H(Z_k - CU_k)$$

$$+ \frac{1}{2}\sum_{k=0}^{N-1}[(Z_k - CU_k)^T Q(Z_k - CU_k) + F_k^T R F_k] \tag{B-4}$$

Let us assume J_{N-k} be the cost function needed to go from the $N-k^{th}$ stage to the N^{th} stage as

$$J_{N-k,N} = \frac{1}{2}(Z_{N-k} - CU_{N-k})^T Q(Z_{N-k} - CU_{N-k}) + J_{N-k+1,N} \tag{B-5}$$

with

$$J_{N,N} = \frac{1}{2}(Z_N - CU_N)H(Z_N - CU_N) \tag{B-6}$$

At first, we consider the case for k=1. From Eqn. (B-5)

$$J_{N-1,N} = \frac{1}{2}(Z_{N-1}-CU_{N-1})_T Q(Z_{N-1}-CU_{N-1})$$

$$+ \frac{1}{2}F^T_{N-1}RF_{N-1} + \frac{1}{2}(Z_N-CU_N)H(Z_N-CU_N) \tag{B-7}$$

The minimum cost, $J^*_{N-1,N}$ is a function of U_{N-1} and F_{N-1} ,

$$J^*_{N-1,N} = \min\{J_{N-1,N}(U_{N-1},F_{N-1})\} \tag{B-8}$$

Now U_N is related to F_{N-1} by Eqn. (B-1) so that

$$J^*_{N-1,N} = \min\{\frac{1}{2}(Z_{N-1}-CU_{N-1})^T Q(Z_{N-1}-CU_{N-1})$$

$$+ \frac{1}{2}F^T_{N-1}RF_{N-1}$$

$$+ \frac{1}{2}[(Z_N-CAU_{N-1}-CBF_{N-1})^T H(Z_N-CAU_{N-1}-CBF_{N-1})]\} \tag{B-9}$$

In order to minimize the $J_{N-1,N}$ with respect to the U_{N-1} , The next condition must be satisfied.

$$\frac{\partial J_{N-1,N}}{\partial F_{N-1}} = 0 \tag{B-10}$$

Evaluating the indicated partial derivertive gives

$$F^*_{N-1} = [R + (CB)^T HCB]^{-1}(CB)^T H[Z_N-CAU_{N-1}] \tag{B-11}$$

$$= K_{N-1}E_{N-1}$$

Where

$$K_{N-1} = [R + (CB)^T HCB]^{-1}(CB)^T H \tag{B-12}$$

$$E_{N-1} = Z_N - CAU_{N-1} \tag{B-13}$$

For convinience, we define

$$H = P_0 \tag{B-14}$$

Substituting Eqn. (B-12) into Eqn. (B-9), the minimum cost can be expressed as

$$\begin{aligned} J^*_{N-1,N} &= \frac{1}{2}(Z_{N-1}-CU_{N-1})^T Q(Z_{N-1}-CU_{N-1}) \\ &+ \frac{1}{2}F^T_{N-1}RF_{N-1} \\ &+ \frac{1}{2}[(Z_N-CAU_{N-1}-CBF_{N-1})^T P_0(Z_N-CAU_{N-1}-CBF_{N-1}) \end{aligned} \tag{B-15}$$

$$\begin{aligned} &= \frac{1}{2}(Z_{N-1}-CU_{N-1})^T Q(Z_{N-1}-CU_{N-1}) \\ &+ \frac{1}{2}(K_{N-1}E_{N-1})^T R(K_{N-1}E_{N-1}) \\ &+ \frac{1}{2}[(E_{N-1}-CBK_{N-1}E_{N-1})^T P_0(K_{N-1}-CBK_{N-1}E_{N-1})] \end{aligned} \tag{B-16}$$

$$\begin{aligned} &= \frac{1}{2}(Z_{N-1}-CU_{N-1})^T Q(Z_{N-1}-CU_{N-1}) \\ &+ \frac{1}{2}E^T_{N-1}(K^T_{N-1}RK_{N-1})E_{N-1} \\ &+ \frac{1}{2}[\{(I-CBK_{N-1})E_{N-1}\}^T P_0\{(I-CBK_{N-1})E_{N-1}\}] \end{aligned} \tag{B-17}$$

$$\begin{aligned}&= \frac{1}{2}(Z_{N-1}-CU_{N-1})^T Q(Z_{N-1}-CU_{N-1})\\ &+ \frac{1}{2}E^T_{N-1}(K^T_{N-1}RK_{N-1})E_{N-1}\\ &+ \frac{1}{2}E^T_{N-1}[(I-CBK_{N-1})^T P_0(I-CBK_{N-1})]E_{N-1}\end{aligned} \tag{B-18}$$

$$\begin{aligned}&= \frac{1}{2}(Z_{N-1}-CU_{N-1})^T Q(Z_{N-1}-CU_{N-1})\\ &+ \frac{1}{2}E^T_{N-1}[K^T_{N-1}RK_{N-1}+(I-CBK_{N-1})^T P_0(I-CBK_{N-1})]E_{N-1}\end{aligned} \tag{B-19}$$

Where the second term of Eqn. (B-19)

$$K^T_{N-1}RK_{N-1}+(I-CBK_{N-1})^T P_0(I-CBK_{N-1}) \tag{B-20}$$

$$= k^T_{N-1}RK_{N-1}+[P_0-(CBK_{N-1})^T P_0](I-CBK_{N-1}) \tag{B-21}$$

$$\begin{aligned}&= K^T_{N-1}RK_{N-1} + P_0 - P_0CBK_{N-1}\\ &- (CBK_{N-1})^T P_0+(CBK_{N-1})^T P_0(CBK_{N-1})\end{aligned} \tag{B-22}$$

$$\begin{aligned}&= P_0 - P_0CBK_{N-1} - (CBK_{N-1})^T P_0\\ &+ K^T_N(R + B^T C^T P_0CB)K_{N-1}\end{aligned} \tag{B-23}$$

$$\begin{aligned}&= P_0 - P_0CBK_{N-1} -(CBK_{N-1})^T P_0\\ &+ K^T_{N-1}(R+B^T C^T P_0CB)(R+B^T C^T P_0CB)^{-1}B^T C^T P_0\end{aligned} \tag{B-24}$$

$$= P_0(I-CBK_{N-1}) \tag{B-25}$$

$$= P_0(I - CB[R + B^T C^T P_0 CB]^{-1} B^T C^T P_0)$$

(B-26)

We define Eqn. (B-25) as P_1

$$P_1 = P_0 - P_0 CB[R + B^T C^T P_0 CB]^{-1} B^T C^T P_0$$

(B-27)

Therefore, we can get the next equation

$$J_{N-1,N} = \frac{1}{2}(Z_{N-1} - CU_{N-1})^T Q (Z_{N-1} - CU_{N-1})$$

$$+ \frac{1}{2} E_{N-1}^T P_0 E_{N-1}$$

(B-28)

Finally, we can get the next three iterative equations :

$$F_{N-1}^* = K_{N-1}[Z_N - CAU_{N-1}]$$

(B-29)

$$P_1 = P_0 - P_0 CB[R + B^T C^T P_0 CB]^{-1} B^T C^T P_0$$

(B-30)

$$K_{N-1} = [R + B^T C^T P_0 CB]^{-1} B^T C^T P_0$$

(B-31)

For k=2, by the same procedure, we can get the those three equations :

$$F_{N-2}^* = K_{N-2}[Z_{N-1} - CAU_{N-2}]$$

(B-32)

$$P_2 = P_1 - P_1 CB[R + B^T C^T P_1 CB]^{-1} B^T C^T P_1$$

(B-33)

As a summury of the procedure, we can finnay get the next general equations :

$$F_{N-k}^* = K_{N-k}[Z_{N-k+1} - CAU_{N-k}]$$

(B-34)

$$P_k = P_{k-1} - P_{k-1} CB[R + B^T C^T P_{k-1} CB]^{-1} B^T C^T P_{k-1}$$

(B-35)

$$K_k = [R + B^T C^T P_{N-k} CB]^{-T} B^T C^T P_{N-k}$$

(B-36)

$$P_0 = H$$

(B-37)

APPENDIX C

Let consider state vector equation with the BOD index,u_1 whose space function is epxpress by G as it nodal vector value

$$u_1(n+1) = Au_1(n) + Gy(n) \tag{C-1}$$

The output equation at measuring station x_{m_i} is

$$u_1(x_{m_i},n) = C_i^T u_1(n) \tag{C-2}$$

For convinience, the initial condition is assumed to be 0. Applying z-transform for Eqn.(C-1) and (C-2), the next two equations are obtained

$$zU_1(z) = AU_1(z) + Gy(z) \tag{C-3}$$

$$U_1(x_{m_i},z) = C_i^T U_1(x_{m_i},z) \tag{C-4}$$

Substituting Eqn.(C-3) into (C-4), the following equation is obtained :

$$U_1(x_{m_i},z) = C_i^T T(z)Gy(z) \tag{C-5}$$

where we define

$$T(z) = (zI - A)^{-1} \tag{C-6}$$

Since $z = e^{-j2\pi f\Delta T}$, Eqn.(C-5) can be expressed with Fourier series such as

$$U_1(x_{m_i},f) = C_i^T T(f)Gy(f) \tag{C-7}$$

On the other hand, let consider the state vector equation with the BOD, u_2 whose space function is R (= the unit vector) as its nodal expression :

$$u_2(n+1) = Au_2(n) + Ry(n) \tag{C-8}$$

The output equation is

$$u_2(x_{m_i},n) = C_i^T u_2(n) \tag{C-9}$$

By applying the same procedure from Eqn.(C-3) to (C-7), the following equation is obtained :

$$U_2(x_{m_i},f) = C_i^T T(f) Ry(f) \tag{C-10}$$

Next, we consider the coherence function $r_1(x_{m_i},f)$ between the output $u_1(x_{m_i},t)$ and the measurement data $u_m(x_{m_i},t)$.

$$r_1(x_{m_i},f) = \frac{|\,C_{1m}\,|^2}{B_m(x_{m_i},f)B_1(x_{m_i},f)} \tag{C-11}$$

$$= \frac{|\,C_{m_iT}(f)GC_{1f}\,|^2}{B_m|c_i^T T(f)Gy(f)|^2} \tag{C-12}$$

$$= \frac{|\,C_{ym}\,|^2}{B_m(x_{m_i},f)|\,y(f)\,|^2} \tag{C-13}$$

On the other hand, the coherence function $r_2(x_{m_i},f)$ between the output $u_2(x_{m_i},t)$ and the mearement data $u_m(x_{m_i},t)$.

$$r_2(x_{m_i},f) = \frac{|\,C_{2m}\,|^2}{B_m(x_{m_i},f)B_2(x_{m_i},f)} \tag{C-14}$$

$$= \frac{|\,C_{m_iT}(f)RC_{2f}\,|^2}{B_m|\,C_i^{T(f)}Ry(f)\,|^2} \tag{C-15}$$

$$= \frac{| C_{ym} |^2}{B_m(x_{m_i},f) | y(f) |^2} \tag{C-16}$$

Therefore,

$$r_1(x_{m_i},f) = r_2(x_{m_i},f) \tag{C-17}$$

Therefore, the coherence function gives the same values for the different space function expressed by nodal values. As a result, the space function g(x) can be replaced by simply 1.

Appendix D

```
C
C
      PROGRAM HBOD
C
C     THIS IS A MAIN PROGRAM FOR HOST MACOMPUTER
C     TO CALCULATE THE ONE DIMENSIONAL PARTIAL
C     DIFFERENTIAL EQUATION IN THE RIVER POLLUTION
C     SYSTEM
C
      REAL VAL1,VAL2,VAL3,TO,TIME,OTIME,T
      REAL PI,DX,DT,A,C,W
      REAL P1,P2,P3
      REAL X,B
      REAL BOD,F(499),FF(499),FS
      REAL OUT1(101),OUT2(101),OUT3(101),OUT4(101)
C
      INTEGER*2 PLUN,ILUN,OLUN
      INTEGER*2 MMPHYS(2)
      INTEGER*2 N,KTIME,I,J
C
      COMMON /WINDOW/ VAL2(499),VAL3(499),FS(499),BOD(501),
     *                VAL1,N,KTIME
C
      DATA PLUN,ILUN,OLUN /3,5,7/
      DATA MMPHYS /4096,0/
C
      OPEN (UNIT=PLUN,NAME='MBOD.QL',
     *      TYPE='OLD',READONLY)
C
      OPENMM(0,PLUN,MMPHYS,IERR)
      IF(IERR.NE.0) STOP
C
      PI=3.1415926
      N=100
      DX=0.02*PI
      DT=0.001
      KTIME=100
      A=0.01
      C=0.1
    C
      P1=A*DT/DX/DX
      CALL RPOKE(VAL1,P1)
      TYPE 40,N
   40 FORMAT(1H , ' N = ',I5)
C
      DO 5 I=1,N-1
      X=(FLOAT(I)-0.5)*DX
      B=0.25*(1.0+SIN(X))
      P2=P1+B*DT/DX
      P3=1.0-(P1+P2)-C*DT
      F(I)=DT*(SIN(X)+1.0)
      CALL RPOKE(VAL2(I),P2)
      CALL RPOKE(VAL3(I),P3)
    5 CONTINUE
C
      DO 10 I=1,N+1
      CALL RPOKE(BOD(I),0.0)
   10 CONTINUE
C
      TO=0.0
C
C
      DO 20 I=1,N-1
      FF(I)=F(I)
      CALL RPOKE(FS(I),FF(I))
   20 CONTINUE
C
      TO=SECNDS(TO)
   99 CALL STRTMM(0,IERR)
      IF(IERR.NE.0) STOP
C
C
C
      CALL WAITMM(IERR)
      IF(IERR.NE.0) STOP
C
      TIME=SECNDS(TO)
C
C
      TYPE 300,TIME
  300 FORMAT(1H ,///,' COMPUTATION TIME = ',
     *                E12.6, ' (SEC)')
C
      CALL CLOSMM(IERR)
C     IF(IERR.NE.0) STOP
C
      STOP
      END
C
C
```

```
C
C
C
      PROGRAM MBOD
C
C     THIS IS A MCL PROGRAM FOR MINI-MAP TO CALCULATE
C     ONE DIMENSIONAL PARTIAL DIFFERENTIAL EQUATION
C     IN THE RIVER POLLUTION SYSTEM
C
      REAL VAL1,VAL2,VAL3,BOD,FS
      REAL P2,P3,BBOD
      REAL O1,O2,O3,O4
C
      INTEGER N,KTIME,K
C
      COMMON /ACOM/ VAL2(499),VAL3(499),FS(499),BOD(501),
     *              VAL1,N,KTIME
      COMMON /BCOM/ P2(499),P3(499),BBOD(501),
     *              O1(100),O2(100),O3(100),O4(100)
C
C
   99 K=1
  100 CALL VMUL(P2(1:N-1),1.0,BOD(1:N-1),0.0,VAL2(1:N-1),0.0)
      CALL VMUL(P3(1:N-1),1.0,BOD(2:N  ),0.0,VAL3(1:N-1),0.0)
C
      CALL VSMA2(BBOD(2:N),1.0,P2(1:N-1),1.0,P3(1:N-1),0.0)
      CALL VSMA2(BBOD(2:N),VAL1,BOD(3:N+1),1.0,FS(1:N-1),
     *           1.0,BBOD(2:N),0.0)
C
      CALL VMOV(BOD(2:N),BBOD(2:N))
C
      K=K+1
      IF(K.LT.KTIME+1) GO TO 100
C
      PAUSE
      GO TO 99
C
      STOP
      END
C
C
```

<u>Appendix E</u>

```
C
C
      PROGRAM HDO
C
C     THIS IS MAIN PROGRAM FOR HOST COMPUTER
C     TO CALCULATE THE DO INDEX OF
C     ONE-DIMENSIONAL PARTIAL DIFFERENTIAL EQUATION
C     IN THE RIVER POLLUTION SYSTEM WITH DO INDEX
C
      REAL VAL1,VAL2,VAL3,VAL4,VAL5,TO,TIME,OTIME,T
      REAL PI,DX,DT,A,C,W,CR,CD
      REAL P1,P2,P3,P4,P5
      REAL X,B
      REAL BOD,DO,F(1001),FF,FS
      REAL OUT1(101),OUT2(101),OUT3(101),OUT4(101)
C
      INTEGER*2 PLUN,ILUN,OLUN
      INTEGER*2 MMPHYS
      INTEGER*2 N,KTIME,I,J
C
C
      COMMON /WINDOW/ VAL2(501),VAL3(501),FF(501),
     *                VAL1,VAL4,VAL5,N,KTIME
C
C
C
      DATA PLUN,ILUN,OLUN / 3,5,7 /
      DATA MMPHYS / 4096 /
C
      OPEN (UNIT=PLUN,NAME='MDO.QL',
     *      TYPE='OLD',READONLY)
C
      CALL OPENMM(O,PLUN,MMPHYS,IERR)
      IF(IERR.NE.O) STOP
C
      TYPE *,' -----------------------------------------'
      TYPE *,' ---                                  ----'
      TYPE *,' --- TYPE IN THE NUMBER OF DATA       ----'
      TYPE *,' ---                                  ----'
      TYPE *,' -----------------------------------------'
      ACCEPT *, KTIME
C
      TYPE *,' -----------------------------------------'
      TYPE *,' ---                                ------'
      TYPE *,' --- THE NUMBER OF STEP             ------'
      TYPE *,' ---                                ------'
      TYPE *,' -----------------------------------------'
      ACCEPT *, N
C
      PI=3.1415926
      DX=O.02*PI
      DT=0.001
      A=0.5
      C=0.1
      CR=C
      CD=0.1
      VS=0.5
C
      P1=A*DT/DX/DX
      CALL RPOKE(VAL1,P1)
      TYPE 40,N
   40 FORMAT(1H , ' N = ',I5)
C
      DO 5 I=1,N-1
      X=(FLOAT(I)-0.5)*DX
      B=0.25*(1.0+SIN(X))
      P2=P1+B*DT/DX
      P3=1.0-(P1+P2)-C*DT
      F(I)=DT*(SIN(X)+1.0)
      CALL RPOKE(VAL2(I),P2)
      CALL RPOKE(VAL3(I),P3)
    5 CONTINUE
C
      P4=DT*CR*VS
      P5=-DT*CD
      CALL RPOKE(VAL4,P4)
      CALL RPOKE(VAL5,P5)
C
      MMPHYS=12288
C
      CALL SETMMR(MMPHYS,IEER)
      IF(IERR.NE.0) STOP 'FAIL SET'
C
      DO 10 I=1,N+1
      CALL RPOKE(FF(I),F(I))
      CALL RPOKE(VAL2(I),0.0)
      CALL RPOKE(VAL3(I),5.0)
   10 CONTINUE
C
C
      TO=0.0
C
C
C
C
      TO SECNDS(TO)
   99 CALL STRTMM(0,IERR)
      IF(IERR.NE.0) STOP
C
C
C
      CALL WAITMM(IERR)
      IF(IERR.NE.0) STOP
```

```
C
      TIME=SECNDS(T0)
C
C
C
  300 FORMAT(1H ,///, ' COMPUTATION TIME = ',E12.6,
     *                ' (SEC)',///)
C
C
      TYPE 300 TIME
      CALL CLOSMM(IERR)
      IF(IERR.NE.0) STOP
C
      STOP
      END
C
```

```
C
C
      PROGRAM MDO
C
C     THIS PROGRAM IS FOR MCL PROGRAM FOR MINI-MAP TO
C     CALCULATE THE DO INDEX IN THE ONE DIMENSIONAL
C     PARTIAL DIFFERENTIAL
C     EQUATION IN THE RIVER POLLUTION SYSTEM
C
      REAL VAL1,VAL2,VAL3,VAL4,VAL5
      REAL P2,P3,P4,P5,BOD,BBOD,DO,DDO,FS,F
      REAL O1,O2,O3,O4
      INTEGER N,KTIME,K
C
      COMMON / ACOM / VAL2(501),VAL3(501),F(501),
     *                VAL1,VAL4,VAL5,N,KTIME
      COMMON / BCOM / BOD(501),DO(501),FS(501)
      COMMON / CCOM / P2(501),P3(501),P4(501),
     *                P5(501),BBOD(501),DDO(501)
     *                ,O1(100),O2(100),O3(100),O4(100)
C
C
   99 K=1
  100 CALL VMUL(P2(1:N-1),1.0,BOD(1:N-1),
     *          0.0,VAL2(1:N-1),0.0)
      CALL VMUL(P3(1:N-1),1.0,BOD(2:N  ),
     *          0.0,VAL3(1:N-1),0.0)
      CALL VMUL(P4(1:N-1),1.0,DO  (1:N-1),
     *          0.0,VAL2(1:N-1),0.0)
      CALL VMUL(P5(1:N-1),1.0,DO  (2:N  ),
     *          0.0,VAL2(1:N-1),0.0)
C
      CALL VSMA2(BBOD(2:N),1.0,P2(1:N-1),
     *          1.0,P3(1:N-1),0.0)
      CALL VSMA3(BBOD(2:N),VAL1,BBOD(3:N+1),
     *          1.0,FS(1:N-1),1.0,BBOD(2:N,0.0)
      CALL VSMA2(DDO  (2:N),1.0,P4(1:N-1),
     *          1.0,P5(1:N-1),0.0)
      CALL VSMA3(DDO  (2:N),VAL1,DDO(3:N+1),
     *          VAL5,BBOD(2:N),VAL4,DDO(2:N),0.0)
C
      CALL VMOV(BOD(2:N),BBOD(2:N))
      CALL VMOV(DO  (2:N),DDO  (2:N))
C
      K=K+1
      IF(K.LT.KTIME+1) GO TO 100
C
      PAUSE
      GO TO 99
C
      STOP
      END
```

C
C
C

Appendix E

```
C
C
C
C
C
      PROGRAM MGRA
C
C
      INTEGER*2 IPRINT,KTIME,M,N,K,I,J,L
      INTEGER*2 ILUN,PLUN,OLUN,MMPHYS
C
      REAL PI,ALPHA,DT,DX,A1,A2,B,S,FRATE,RECHG,P
      REAL AMP,K1,K2,KR1,KR2,ST1,ST3
      REAL R1(10,10),R2(10,10),R3(10,10),R4(10,10)
      REAL RO(10,10),RR1(10,10),RR3(10,10)
      REAL T1,T2,T3,T4,TO
      REAL W,Q,WR(10,10),QR(10,10),H,C
      REAL VX,VY,D
      REAL TEMP1,TEMP2

      REAL TO,TIME,TTIME,CTIME,TE1,TER1,TE2,TER2,HIGH
      REAL TER,DDCHX,DDT,PP,HHIGH
C
      COMMON/ /WINDOW/ DT,DX,P,GIHIGH,K1,K2,TE2,
     *T1(10,10),T2(10,10),T3(10,10),T4(10,10),
     *TO(10,10),ST1(10,10)ST3(10,10),W(10,10),Q(10,10),
     *VX(10,10),XVY(10,10),TEMP1(10,10),TEP2(10,10),
     *D(10,10),C(12,12),H(12,12),N,K,KTIMEC
C
      DATE PLUN,ILUN,OLUN /3,5,7/
      DATE MMPHYS /4096/
C
C
C
      OPEN (UNIT=PLUN,NAME='MCRAD.PL',
     *       TYPE='OLD',READONLY)
C
      CALL OPENMM(O,PLUNMMPHYS,IERR)
      IF(IERR.NE.O) STOP 'MM NOT AVAILABLE'
C
      PI=3.1415926
      DDX=500.0
      DDT=0.02
      PP=0.3
      AMP=100.0
      S=0.015
      HHIGH=100.0
      A1=2.25
      A2=1.0
      B=31000.0
      FRATE=200000.0
      RECHY=80000.0
C
      KTIME=20
      M=10
      N=10
C
      TYPE 5000,N,N
      TYPE 5001,PI
      TYPE 5002,DDX
      TYPE 5003,DDT
      TYPE 5004,PP
      TYPE 5005,HHIGH
      TYPE 5006,S
      TYPE 5007,AMP
      TYPE 5008,FRATE
      TYPE 5009,RECHG
      TYPE 5010,KTIME
C
 5000 FORMAT(1H ,,'THE NUMBER OF GRID POINT=',I3,'X',I3,/)
 5001 FORMAT(1H ,'PI =',F10.7,/)
 5002 FORMAT(1H ,'GRID SIZE = ', F10.5,' ( FT ) ',/)
 5003 FORMAT(1H ,'TIME STEP sIZE = ',F10.5,' ( DAY )',/)
 5004 FORMAT(1H ,'POROCITY = ',F10.5,/)
 5005 FORMAT(1H ,'LAYER OF AQUIFER = ',F10.2,' ( FT )',/)
 5006 FORMAT(1H ,'STORAGE COEFFICIENT = ',F10.5,/)
 5007 FORMAT(1H ,'POLLUTION SOURCE=',F10.5,'( PPM/DAY)',/)
 5008 FORMAT(1H ,'POMPING RATE=',F10,0,' ( FT**2/DAY )',/)
 5009 FORMAT(1H ,'RECHArGE RATE=',F10.0,'( FT**2/DAY )',/)
 5010 FORMAT(1H ,'THE NUMBER OF DATE =',I5,///)
C
      TER1=DDT/DDX/DDX/S
      TER2=PI*4.0/KTIME
C
      CALL RPOKE(DT,DDT)
      CALL RPOKE(DX,DDX)
      CALL RPOKE(P,PP)
      CALL RPOKE(HIGH,HHIGH)
C
      DO 10 I=1,N
      DO 11 J=1,N
      R1(I,J)=B-(A1*I+A2*(J+0.5))*DDX
      R2(I,J)=B-(A1*I+A2*(J-0.5))*DDX
      R3(I,J)=B-(A1*(I+0.5)+A2*J)*DDX
      R4(I,J)=B-(A1*(I-0.5)+A2*J)*DDX
      CALL RPOKE(ST1(I,J),R1(I,J))
      CALL RPOKE(ST3(I,J),R3(I,J))
      R1(I,J)=TER1*R1(I,J)
      R2(I,J)=TER1*R2(I,J)
      R3(I,J)=TER1*R3(I,J)
      R4(I,J)=TER1*R4(I,J)
      RO(I,J)=1.0-R1(I,J)-R2(I,J)-R3(I,J)-R4(I,J)
C
```

```
      CALL RPOKE(T2(I,J),R2(I,J))
      CALL RPOKE(T1(I,J),R1(I,J))
      CALL RPOKE(T3(I,J),R3(I,J))
      CALL RPOKE(T4(I,J),R4(I,J))
      CALL RPOKE(TO(I,J),RO(I,J))
   11 CONTINUE
   10 CONTINUE
C
      TYPE 1000
 1000 FORMAT(1H ,'TRANSMISSIVITU ( FT**2/DAY )')
C
      DO 14 I=1,N
      TYPE 2000, (RPEeK(ST1(I,L)),L=1,M
 2000 FORMAT(1H ,10F7.0)
   14 CONTINUE
C
      DO 20 I=1,N
      DO 21 J=1,N
      W(I,J)=0.0
      Q(I,J)=AMP*SIN(PI*(I-0.5)/N)*SIN(PI*(J-0.5)/N)
   21 CONTINUE
   20 CONTINUE
C
      W(3,5)=-FRATE
      W(3,9)=-FRATE
      W(7,9)=-FRATE
      W(9,5)=-FRATE
C
      DO 22 I=1,10
      W(I,1)=RECHG
   22 CONTINUE
C
      DO 30 I=1,N
      DO 31 J=1,N
      WR(I,J)=TER1*W(I,J)
      QR(I,J)=DDT*Q(I,J)
      CALL RPOKE(W(I,J),WR(I,J))
      CALL RPOKE(Q(I,J),QR(I,J))
   31 CONTINUE
   30 CONTINUE
C
      DO 40 I=1,N+2
      DO 41 J=1,N+2
      CALL RPOKE(H(I,J),200.0)
      CALL RPOKE(C(I,J),0.0)
   41 CONTINUE
   40 CONTINUE
C
      TYPE 4000
 4000 FORMAT(1H ,'INITIAL CONDITION OF AQUIFER ( PPM )')
      TYPE 8000, 0.0
 8000 FORMAT(1H ,' TIME = ',F10.5)
```

```
      DO 71 I=2,N+1
      TYPE 4001, (RPEEK(C(I,L)),L=2,M+1)
   71 CONTINUE
C
C
      K=1
      TO=0.0
  100 TER=TER2*FLOAT(K)
      CALL RPOKE(TE2,TER)
C
C     START MINI-MAP
      TO=SECNDS(TO)
      CALL STRTMM(0,IERR)
      IF(IERR.NE.0) STOP 'FAILED START'
C
      CALL WAITMM(IERR)
      IF(IERR.NE.0) STOP 'FAILED WAITMM'
C
      OTIME=SECNDS(TO)
      TO=OTIME
      TIME=DDT*K
      TYPE 8000,TIME
      DO 60 I=2,N+1
C     TYPE 4001, (RPEEK(H(I,L)),L=2,M+1)
      TYPE 4001, (RPEEK(C(I,L)),L2=,M+1)
   60 CONTINUE
C
      K=K+1
      IF(K.LT.KTIME+1) GO TO 100
C
C
C
      TYPE 4002,OTIME
 4001 FORMAT(1H ,10F7.2)
 4002 FORMAT(1H ,'COMPUTATION TIME=',F10.5,' ( SEC )')
      CALL CLOSMM(IIERR)
      IF(IERR.NE.0) STOP ' CLOSE FAILED'
C
      STOP
      END
      PROGRAM  MGRAD
C
      INTEGER I,K,KTIME,N
      REAL DX,DT,K1,K2,P,TE2,HIGH,HI
      REAL T1,T2,T3,T4,TO
      REAL ST1,ST3
      REAL W,Q
      REAL H,HH
      REAL TTT1,TTT2,TTT3,TTT4,TTTO
      REAL TEMP1,TEMP2
      REAL VX,VY,VAX,VAY,D
      REAL DD1,DD2.DD3,DD4,DDO
      REAL C,CC
      REAL Z
C
      COMMON /ACOM/ DT,DX,P,HIGH,K1,K2,TE2,
     *T1(10,10),T2(10,1G),T3(10,10),T4(10,10),
     *TO(10,10),ST1(10,10),ST3(10,10),W(10,10),Q(10,10),
     *VX(10,10),VY(10,10),TEMP1(10,10),TEMP2(10,10),
     *D(10,10),C(12,12),H(12,12),N,K,KTIME
      COMMON /BCOM/ TTT1(10,10),TTT2(10,10),TTT3(10,10),
     *TTT4(10,10),TTTO(10,10),
     *VAX(10,10),VAY(10,10)
      COMMON /CCON/ DO1(10,10),DD2(10,10),DD3(10,10),
     *DD4(10,10),DDO(10,10),HH(12,12),CC(12,12),HI,Z,K1,I
C
      Z=-1.0
      HI=1.0/F/DX/HIGH
      K1=DT/DX
      K2=K1/DX
C
C **** RESET DATA  ****
      I=1
   98 CALL VMOV(HH(I,1:N+2),H(I,1:N+2))
      CALL VMOV(CC(I,1:N+2),C(I,1:N+2))
      I=I+1
      IF(I.LT>N+3) GO TO 98
C
C*** CALCULATION OF AQUIFER HEAD QUANTITY ( FT/DAY ) ***
C
C     I=1
C 100 CALL VSMA1(TT1(I,1;N),TE1,T1(I,1:N),O.)
C     CALL VSMA1(TT2(I,1:N),TE1,T2(I,1:N),O.)
C     CALL VSMA1(TT3(I,1:N),TE1,T3(I,1:N),O.)
C     CALL VSMA1(TT4(I,1:N),TE1,T4(I,1:N),O.)
C     CALL VSMA1(ST1(I,1:N),TE3,T1(I,1:N),O.)
C     CALL VSMA1(ST3(I,1:N),TE3,T3(I,1:N),O.)
C     CALL VSMA3(TTO(I,1:N),-1.,TT1(I,1:N),-1.,
C    *TT2(I,1:N),-1.,TT3(I,1:N),O.)
C     CALL VSMA2(TTO(I,1:N),-1.,TT4(I,1:N),1.,TTO(I,1:N),O.)
C
C     CALL VSMA1(W(I,1:N),TE1,W(I,1:N),O.)
C     CALL VSMA1(Q(I,1:N),DT,Q(I,1:N),O.)
C     I=I+1
C     IF(I.LT.N+1) GO TO 100
C
   99 I=1
  101 CALL VMUL(TTT1(I,1:N),1.,H(I+1,3:N+2),O.,T1(I,1:N),O.)
      CALL VMUL(TTT2(I,1:N),1.,H(I+1,1:N  ),O.,T2(I,1:N),O.)
      CALL VMUL(TTT3(I,1:N),1.,H(I+2,2:N+1),O.,T3(I,1:N),O.)
      CALL VMUL(TTT4(I,1:N),1.,(I    2:N+1),O.,T4(I,1:N),O.)
      CALL VMUL(TTTO(I,1:N),1.,H(I+1,2:N+1),O.,TO(I,1:N),O.)
C
```

```
       CALL VSMA3(HH(I+1,2:N+1),1.,TTT1(I,1:N),1.,TTT2(I,1:N),
      *1.O,TTT3(I,1:N),O.)
       CALL VSMA3(HH(I+1,2:N+1),1.,TTT4(I,1:N),1.,TTTO(I,1:N),
      *1.O,HH(I+1,2:N+1),O.)
       CALL VSMA2(HH(I+1,2:N+1),1.,W(I,1:N),1.,
      *HH(I+1,2:N+1),O.)
C
       I=I+1
       IF(I.LT.N+1) GO TO 101
C
       I=1
  102  CALL VMOV(H(I+1,2:N+1),HH(I+1,2:N+1))
       I=I+1
       IF(I.LT.N+1) GO TO 102
C
       CALL VMOV(H(1:N+2,1),HH(1:N+2,2))
       CALL VMOV(H(1:N+2,N+2),HH(1:N+2,N+1))
       CALL VMOV(H(1,1:N+2),HH(2,1:N+2))
       CALL VMOV(H(N+2,1:N+2),HH(N+1,1:N+2))
C
C
C    **** CALCULATION OF AQUIFER QUALITY ( PPM/DAY )
C
C
       I=1
  103  CALL VSMA2(TEMP1(I,1:N),1.O,
      *H(I+1,3:N+2),Z,H(I+1,2:N+1),O.)
       CALL VMUL(VX(I,1:N),HI,ST1(I,1:N),O.,
      *TEMP1(I,1:N),O.)
       CALL VSMA2(TEMP2(I,1:N),1.O,
      *H(I+2,2:N+1),Z,H(I+1,2:N+1),O.)
       CALL VMUL(VY(I,1:N),HI,ST3(I,1:N),O.,
      *TEMP2(I,1:N),O.)
C
C      CALL VAB(VAX(I,1:N),1.,VX(I,1:N),O.)
C      CALL VAB(VAY(I,1:N),1.,VY(I,1:N),O.)
C      CALL VTHR(VX(I,1:N),O.O1,VAX(I,1:N))
C      CALL VTHR(VY(I,1:N),O.O1,VAY(I,1:N))
C
       CALL VMG(D(I,1:N),VX(I,1:N),VY(I,1:N))
C
       CALL VSMA2(DD1(I,1:N),K2,D(I,1:N),K1,VX(I,1:N),O.)
       CALL VSMA2(DD2(I,1:N),K2,D(I,1:N),K1,VX(I,1:N),O.)
       CALL VSMA2(DD3(I,1:N),K2,D(I,1:N),K1,VY(I,1:N),O.)
       CALL VSMA2(DD4(I<1:N),K2,D(I,1:N),K1,VY(I,1:N),O.)
       CALL VSMA3(DDO(I,1:N),Z,DD1(I,1:N),
      *Z,DD2(I,1:N),Z,DD3(I,1:N),1.)
       CALL VSMA2(DDO)I,1:N),Z,DD4(I,1:N),1.,DDO(I,1:N),O.)
C
       CALL VML(DD1(I,1:N),C(I+1,3:N+2))
       CALL VML(DD2(I,1:N),C(I+1,1:N   ))
       CALL VML(DD3(I,1:N),C(I+2,2:N+1))
      CALL VML(DD4(I,1:N),C(I   ,2:N+1))
      CALL VML(DDO(I,1:N),C(I+1,2:N+1))
C
C
      CALL VSMA3(CC(I+1,2:N+1),1.0,DD1(I,1:N),1.0,DD2(I,1:N),
     *1.0,DD3(I,1:N),0.0)
      CALL VSMA3(CC(I+1,2:N+1),1.0,DD4(I,1:N),1.0,DDO(I,1:N),
     *1.0,CC(I+1,2:N+1),0.0)
      CALL VSMA2(CC(I+1,2:N+1),TE2,Q(I<1:N),1.0,
     *CC(I+1,2:N+1),0.0)
C
      I=I+1
      IF(I.LT.N+1) GO TO 103
C
      I=1
  110 CALL VMOV(C(I+1,2:N+1),CC(I+1,2:N+1))
      I=I+1
      IF(I.LT.N+1) GO TO 110
      CALL VMOV(C(1:N+2,1),CC(1:N+2,2))
      CALL VMOV(C(1:N+2,N+2),CC(1:N+2,N+1))
      CALL VMOV(C(1,1:N+2),CC(2,1:N+2))
      CALL VMOV(C(N+2,1:N+2),CC(N+1,1:N+2))
C
C
      PAUSE
      GO TO 99
C
      STOP
      END
```

Appendix G

```
C
C
C     THIS IS MAIN PROGRAM FOR HOST COMPUTER
C     TO CALCULATE FEATURE VECTORS IN RIVER
C     POLLUTION SYSTEMS
C
      REAL T,Y(4,2048),X(4,2048)
      REAL U
      REAL TIME
      REAL UN,UL
C
      INTEGER*2 PLUN,ILUN,OLUN
      INTEGER*2 MMPHYS(2),IERR,N,L,I,NTIME,K,NP,MEASU
C
      COMMON /WINDOW/ U(2048)
C
      DATA PLUN,ILUN,OLUN /3,5,7/
      DATA MMPHYS /4096,0/
C
      OPEN (UNIT=PLUN,NAME='FEAT.QL',
     *      TYPE='OLD',READONLY)
C
      CALL OPENMM(O,PLUN,MMPHYS,IERR)
      IF(IERR.NE.O) STOP
C
      TYPE *,' -----------------------------------'
      TYPE *,' ---                             ---'
      TYPE *,' ---   TYPE IN NUMBER OF DATA    ---'
      TYPE *,' ---                             ---'
      TYPE *,' -----------------------------------'
      ACCEPT *,N
      TYPE *,' -----------------------------------'
      TYPE *,' ---                             ---'
      TYPE *,' ---   TYPE IN THE NUMBER        ---'
      TYPE *,' ---   OF ITERATIONS             ---'
      TYPE *,' -----------------------------------'
      ACCEPT *,NTIME
      TYPE *,' -----------------------------------'
      TYPE *,' ---
      TYPE *,' ---   TYPE IN THE NUMBER OF THE ---'
      TYPE *,' ---   MEASUREMENT STATIONS      ---'
      TYPE *,' ---                             ---'
      TYPE *,' -----------------------------------'
      ACCEPT *, L
      NP=N/L
C
      DO 5 I=1,N
      READ (23,1000) T, (Y(K,I),K=1,MEASU)
      READ (24,1000) T, (X(K,I),k=1,MEASU)
 1000 FORMAT (1H ,E12.6)
    5 CONTINUE
```

```
C
      UN=N
      UL=L
      CALL RPOKE  (U(1),UN)
      CALL RPOKE  (U(2),UL)
C
      J=1
      TO=0.0
C
   99 MMPHYS(1)=12288
      MMPHYS(2)=0
      CALL SETMMR(MMPHYS,IERR)
      IF(IERR.NE.0) STOP  'MAPPIN FAIL'
C
      DO 10 I=1,N
      CALL RPOKE  (U(I),Y(1,I))
      CALL RPOKE  (U(I+1024),X(1,I))
   10 CONTINUE
C
C
      DO 24 K=2,MEASU
      TIME=SECNDS(TO)
C
      CALL STRTMM(0,IEER)
      IF(IEER,NE.0) STOP  'MIN-MAP COULD NOT STARTED'
C
      DO 25 I=1,N
      CALL RPOKE  (U(I),Y(K,I))
      CALL RPOKE  (U(I+1024),X(K,I))
   25 CONTINUE
C
      CALL WAITMM(IERR)
      IF(IERR.NE.0) STOP 'MINI-MAP HUNG DURING PROCESSING'
C
      TO=SECNDS(TIME)
   24 CONTINUE
C
      J=J+1
      IF(J.LT.NTIME+1) GO TO 99
      TO=TO/NTIME
C
      MMPHYS(1)=4096
C
      TYPE *, 'COMPUTATION FINISH '
      DO 30 K=1,NP
      TYPE 300,K,RPEEK(U(K+200)),RPEEK(U(K+400)),
     *          RPEEK(U(K+600)),RPEEK(U(K+800))
  300 FORMAT(1H ,I5,4E15.5)
   30 CONTINUE
C
      TYPE 200, TO
  200 FORMAT(1H ,/, 'COMPUTATION TIME=',E12.6,'(SEC)')
```

```
C
      CALL CLOSMM(IERR)
      IF(IERR.NE.0) STOP 'CLOSE FAIL'
C
      STOP
      END
C
C
```

```
C
C
C
      PROGRAM FEAT
C
C     THIS PROGRAM IS A MCL PROGRAM FOR MINI-MAP
C     TO CALCULATE THE FEATURE VECTORS
C     IN THE RIVER POLLUTION SYSTEMS
C
      REAL D,U,CW,V
      REAL A,B,UY,UX
      REAL Y,X,PY,PX,PXY
      REAL AVPY,AVPX,AVPXYR,AVPXYI,AVC
C
      INTEGER N,NPOW,N1,N2,N3,L,J,M
      INTEGER K,L2,I
C
      COMMON /COM/ D(2048)
      COMMON /ACOM/ U(2048)
      COMMON /A1COM/ UY(2048)
      COMMON /A2COM/ UX(2048)
      COMMON /BCOM/ CW(2048)
      COMMON /CCOM/ V(2048)
      COMMON /DCOM/ Y(2048)
      COMMON /ECOM/ X(2048)
      COMMON /FCOM/ PY(2048)
      COMMON /GCOM/ PX(2048)
      COMMON /HCOM/ PXY(2048)
      COMMON /ICOM/ AVPY(2048)
      COMMON /JCOM/ AVPX(2048)
      COMMON /KCOM/ AVPXYR(2048)
      COMMON /LCOM/ AVC(2048)
      COMMON /MCOM/ A,B,NPOW,N1,N2,N3,J,K,L2,I,M
      COMMON /NCOM/ AVPXYI(2048)
C
      CALL SFX2(N,D(1))
      CALL SFX2(L,D(2))
C
      A=1.0/N
      N1=N/2
      N2=N+N1
      N3=N*2
      B=1.0/L
      L2=L*2
      I=N/L
C
      CALL SSIN2(A,1.0,A)
      CALL VCOSR(CW(1:N3:2),0.0,1.0,A)
      CALL VMOV(CW(2:N1:2),CW(N2+1:N3:2))
      CALL VMOV(CW(N1+2:N3:2),CW(1:N2:2))
C
    1 CALL VMOV(UY(1:N3:2),U(1:N:1))
```

```
      CALL VMOV(UX(1:N3:2),U(1025:N+1024:1))
C
      CALL FCFOO(Y(1:N3:2),0,UY(1:N3:2),
     *V(1:N3:2),CW(1:N3:2))
      CALL FCFOO(X(1:N3:2),0,UX(1:N3:2),
     *V(1:N3:2),CW(1:N3:2))
C
C
C
      CALL VPOW(PY(1:N:1),0.0,Y(1:N3:2),0.0)
      CALL VMSQ(PY(1:N:1),Y(1:N3-1:2),Y(2:N3:2))
      CALL VMSQ(PX(1:N:1),X(1:N3-1:2),X(2:N3:2))
      CALL VPOW(PX(1:N:1),0.0,X(1:N3:2),0.0)
C
      CALL CCVML(PXY(1:N3:2),0.0,Y(1:N3:2),0.0,X(1:N3:2))
C
      K=1
  100 J=(K-1)*L
      M=J*2
      CALL RSUM(AVPY(K),PY(J+1:J+L:1),B)
      CALL RSUM(AVPX(K),PX(J+1:J+L:1),B)
      CALL RSUM(AVPXYR(K),PXY(M+1:M+L2-1:2),B)
      CALL RSUM(AVPXYI(K),PXY(M+2:M+L2:2),B)
      K=K+1
      IF(K.LT.I+1) GO TO 100
C
      CALL VMSQ(AVC(1:I:1),AVPXYR(1:I:1),AVPXYI(1:I:1))
      CALL VDVX(AVC(1:I:1),AVPY(1:I:1))
      CALL VDVX(AVC(1:I:1),AVPX(1:I:1))
      CALL VMOV(D(201:I+200:1),AVC(1:I:1))
C
   88 PAUSE
      GO TO 1
C
      END
C
C
```

REFERENCES

[ApKa82] Apenyo, A. and Karplus, W. J., "A Domein Segmentation Approach to Modeling Distributed Parameter Systems," IEEE, Transactions on Systems, Man and Cybernetics, Vol. SMC-12, pp299-307, May/June 1982

[Karp80] Karplus, W. J., "The Rule of Pattern Recognition Methods in Mathematical Modeling of Distributed Parameter Systems," Computer Science Department Quarterly, University of California, Los Angeles, Vol. 8, No. 2, pp115-122,Spring Quarterly 1980

[Simu75] Simundich, T. M., "Systems Characterization: A Pattern Recognition Approach," Ph.D. Dissertation, Computer Science Department,University of California, Los Angeles, 1975

[Apen79] Apenyo, K., "Segmentation of the domain of distributed parameter systems," Ph.D. dissertation, University of California, Los Angeles, 1979

[SiKa75] Simundich, T. M. and Karplus, W. J., "Feature Extraction Methods for the Modeling of Distributed Parameter Systems," Proceeding of the Summer Computer Simulation Conference, pp85-89 1975

[Ande79] Anderson, M.P., "Using Models to Simulate the Movement of Contaminants through Ground Water Flow Systems," Critical Reviews in Environmental Control, Vol. 9, Issue 2, pp97-156, November 1979

[MeFa80] Mercer, J. W. and Faust, C. R., "Ground-Water Modeling: Mathematical Model," Ground Water, Vol. 18, No.3, pp212-228 May/June 1980

[Rina79] Rinaldi, S., "Modeling and Control of River Quality," McGraw-Hill, N.Y. 1979

[Khan79] Khan, A. I. "Ground Water Simulation Using A Dimensional Flow Model," Water Resources Bulletin, Vol. 15, No. 6, December, pp.1618-1627, 1979

[Khan80] Khan, A. I. "A Lumpted Approach to The Problem in Ground Water HyDrology," Water Resources Bulletin, Vol.16, No. 5, October pp.866-873

[Pick77] Pikens, J. F.,Merritt, W. F. and Cherry, J. A., "Field Determination of the Physical Contaminant Transport Parameters in a Sandy Aquifer, in Proc. IAEA Advisory Grounp Meeting, Cracow, Poland, 1977

[YoBe74] Young, P. and beck, B., "The Modeling and Control of Water Quality in a River System," Autmatica, Vol.10, pp455-468, 1974

[StPh25] Streeter, H. W. and Phelps, E. B., "A Study of the Pollution and Natural Purification of the Ohio River," Public Hralth Bulletin, Vol. 3, No.146, United States Public Health Service, Reprinted by U.S. Department of Health, Education and Welfare, 1958

[O'con67] O'connor, D. J., "The Temporal and Distribution of Dissolved Oxygen in Streams," Water Resources Research, Vol. 3, No. 1, pp65-79, 1967

[O'cDi70] O'connor, D. T. and Ditoro, D. M., "Photosynthesis and Oxygen Balance in Streams," J. Sanit. Eng. Div. Amer. Soc. Civil Eng., Vol. 96(SA2), pp547-571, 1970

[Dobb64] Dobbins, W. E., "BOD and Oxygen Relationships in Streams," J. Sani. Eng. Div., Proc. ASCE, Vol. 90, pp53-78, 1964

[BaBr79] Baumert, H.Braun, E. G., "Modeiling and Computation of Water Quality Problems in River Network," Proc. IFIP'79, pp482-491, 1979

[Nibo82] Nihoul, J. C., "The effect of Fresh Water Discharges from Artificial Reservoirs on The Water Quality in Polluted River," Mathematics and Computers in Simulation XXIV, pp204-207, 1982

[KoPh76] Konivo, A. J. and Phillips, G., "Optimal Estimation of DO, BOD, and Stream Parameters Using a Dynamic Discrete Time Model," Water Resources Research, Vol.12, No. 4, pp705-711, August, 1976

[BoGr78] Bowles, D. S. and Grenny, W. J., "Steady State River Modeling by Sequential Extended Kalman Filters," Water Resources Research, Vol. 14, No.1, pp84-96, February, 1978

[MeBB79] Merbis, M., Boer, B. and Bagchi, A., "Parameter Estimation in a Model for the Oxygen Content of Surfacewater in a River," IFAC symposium'79, pp795-799, 1979

[McHu72] McMichael, F. C., "Stochastic Modeling of Temperature and Flow

[Geis75] Geisser, S., "The Predictive Sample Reuse Method with Applications," Journal of the American Statistical Association, 70, pp.320-328, 1975

[Ston74] Stone, M., "Cross-Validatory Choise and Assessment of Statistical Predictions," Journal of the American Statistical Society Series, B-36, pp.111-147, 1974

[Wahb75] Wahb "A Completely Automatic French Curve : Fitting Spline Cross-Varidation," Communication in Statistics, 4, pp.1-17, 1975

[Meis72] Meisel, W. S., "Computer-Oriented Approaches to Pattern Recognition," Academic Press, N.Y., 1972

[Kana68] Kanal, L. N., "Pattern Recognition," Thompson Book Co., Washington, D.C. 1968

[GoTh78] Gonzales, R. C. and Thomason, M. G., "Syntactic Pattern Recognition, A Introduction," Addison_Wesley Publishing Co., Massachusetts, 1978

[Fu76] Fu, K. D., "Digital Pattern Recognition," Spring-Verlag, N.Y.,1976

[ToGo74] Tou, J. T. and Gonzales, R. C., "Pattern Recognition Principles," Addison-Wesley Publishing Co.,Massachusetts, 1974

[Fu68] Fu, K. D., "Sequential Methods in Pattern Recognition and Machine Learning," Academic Press, N.Y., 1968

[Fu74] Fu, K. D. "Syntactic Methods in Pattern Recognition," Academic Press, N.Y., 1974

[BrPi73] Bredehoeft, J. D. and Pinder, G. F., "Mass Transport in Flowing Gound Water", Water Resour. Res., 9, 194, 1973.

[Koni77] Konibo, L. F., "Modeling Chloride Movement in The Alluvial Aquifer at The Rocky Mountain Arenal, Colorado, U.S. Geological Sarvey Water Supply Paper 2044, 1977.

[KOBr74] Konibow, L. F. and Gredehoeft, J. D., "Modeling Flow and Chemical Quality Changes in an Irrigated Stream-Aquifer System", Water Resour. Res., 10, 546, 1974

[Robs78] Robson, S.G., "Application of Digital Profile Modeling Illustrated by Application as Barstow, Ca", U.S. Geological Survey Water Research Investigation Report 66, 1974, pp.46-73.

[PiCh78] Pickens, J. F., Çherry, J. A., Grisak, G. E., Merritt, W. F., and Risto, B. A., "Amulti-Level Device for Ground-Water Sampling and Piezometric Modeling", Ground Water, 16, 322, 1978

[IkMS74] Ikeda, S., Miyamoto, S.and Sawaragi, Y., "Identification Method in Environmental Systems and Its Application to Water Pollution", Int. J. Sys. Sci, 5-8, 1974, pp.707-723.

[IkMS77] Ikeda, S., Miyamoto, S. and Sawaragi, Y., " Reguralization Method for Identification of Distributed Systems," IFAC Sympo., I5-1, 1977, pp.53-63.

[IkMS78] Ikeda, S., Miyamoto, S. and Sawaragi, Y., " Identification of Distributed Systems and the Theory of the Regularization," Jourl. Math. Analy. and Appl., 63, 1978, pp.77-95.

[Tikh63] Tikhonov, A. N., " Solution of Incorrectly Formulated Problems and The Regularization Method", Dokl. Akad. Nauk. SSSR., 151, 1963, pp1035-1038.

[UnUR78] Umed.S. P., Unny,T. E. and Ragageh, R.K., "A Feature Prediction Model in Synthetic Hydrology Based on Concepts of Pattern Recognition." Water Resources Research Vol. 14, No.2, 1978, pp.335-344.

[UmUn80] Umed S. P. and Unny, T. E., "Extension and Application of Feature Prediction Model for Synthesis of Hydrologic Records." Water Resources Research, Vol.16, No.1,1980, pp.77-96.

[Karp72] Karplus, W. J., "Systems Identification and Simulation - A Pattern Recognition Approach", Proceedings of the Fall Joint Computer Conference, 1972, pp.385-392.

[NeFa80] Nercer, J. W. and Faust, C.R., " Ground - Water Modeling : Mathematical Models," Ground Water, Vol. 18, No. 3, May, 1980, pp.212- 227.

[Pere72] Perez, A. I. and at al., " A Water Quality Model for Conjunctive Surface - Ground Water System : An Overview," Water Resources Bulletin, Vol.8, No.5, Oct. 1972, pp.900-908.

[DuHa73] Duda, R. O. and Hart, P. E. " Pattern Classification and Scene Analysis," John Wiley & Son, Inc., 1973.

[SaHo74] Saridis, G. H. and Hofstadter, R.E., " A pattern Recognition Approaches to the Characterization of Non Linear Systems," IEEE, Trans. on Sys. Man and Cyber., Vol. SMC-4, No.4, July 1974, pp.362-370.

[OtEn78] Otnes, R. K. and Enochson, L., " Applied Time Series Analysis," John- Wiley and Sons, Inc., 1978.

[BePi71] Bendat, J.S. and Piersol, A. G., " Random Data Analysis And Measurement Procedures," John Wiley and Sons, Inc., 1971.

[GeSa76] Gentleman, H.M. and Sanae, G., " Fast Fourier Transforms for Fun and Profit," Proc. Fall Joint Computer Conference, AFIPS, Vol. 28, 1976, pp.563-578.

[Erin81] Erilinger, D.R., " Time Series : Data Analysis and Theory," expanded edition , Holden-Day, Inc., San Francisco, 1981.

[Karp77] Karplus, W. J., " Peripheral Processors for High-speed Simulation," Simulation, Nov. 1977, pp.143-153.

[Karp81] Karplus, W. J.," Simulation Using Array Processors : Software issues," IEEE, Computers, 1981

[Karp80] Karplus, W. J., " Array Processors for Simulation," UCLA CSD, April, 1980.

[VeKa81] Vemuri, V. and Karplus, W. J., " Digital Computer Treatment of Partial Differential Equations," Prentice-Hall, Inc., 1981.

[Zien71] Zienkiewiz, O. C., " The finite Element Method in Engineering Science," McGraw-Hill Book Co., New York 1971

[DeAb72] Desai, C. S. and Abel, J. F., " Introduction to the Finite Element Method : A Numerical Method for Engineering Analysis," Van Nortrand Reinhold Co., New York, 1972.

[Klah60] Klahr, C. N., " A Monte Carlo Method for the Solution of Elliptic Partial Differential Equations in Mathematical Methods for Digital computers," A. Ralston and H.S. Wilf, eds.,John Wiley & Son. Inc., New York, 1960, pp.157-164.

[King51] King, G. W., " Monte Carlo Method for Solving Diffusion Problems," Industrial and Engineering Chemistry, vol.43, No.11, 1951, pp.2475-2478.

[Wason51] Wason, W., " Random Walk and the Eigenvalues of Elliptic Difference Equations," Journal of Research of the National Bureau of Standards, Vol.46, No.1, Jan. 1951, pp.65-73

[Lall76] Lalles, R., " Method of Resolution for Selected Boundary Problems,"

[Shre66] Shreider, Y. A., " The Monte Carlo Method," Peagamon Press Ltd., 1966.

[Meye54] Meyer, H. A., " Symposium on Monte Carlo Methods," John Wiley & Sons. Inc., New York, 1954.

[Eykh73] Eykhoff, P., "Identification and System Parameter Estimation," Proceedings of the Symposium," The Hague, Netherlands, North-Holland, Amsterdam, 1973

[Beke70] Bekey, G. A. "System Identification -An Identification and Survey," Simulation, Vol. 15, pp.151-161, 1970

[Levi69] Levi, M. D. "Feature Extraction : A survey," Proceedings of IEEE, Vol. 57, No.8, pp.1391-1407, August, 1969

[Thig74] Thiga, R., S., A. and Gough, N., E. "Artificial Intelligence Appliwd to the Discrimination of the Order of Multivariable Linear Systems," International Journal of Control, Vol. 20, No.6, pp.961-969. 1974

[GoFe74] Gonzalez, R. C., and Fry, D. N. and Kryter, R. C. "Results in the Application of Pattern Recognition Methods to Nuclear Core Component Surveillance," IEEE Trans Nucl. Sci., Vol. 21, No. 1, February 1974