THEORY AND APPLICATIONS OF PREDICTIVE STOCHASTIC COMPLEXITY

Jimmy Baikovicius

B.Sc. University of Minnesota 1986

M.Sc. University of Minnesota 1987

Department of Electrical Engineering
McGill University, Montréal

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

December 1992

ISBN 0-315-87643-3

Abstract

The objective of this thesis is to solve model order selection, and change-point detection problems in real time using a form of predictive stochastic complexity. A consistent method for finding the best model order for certain kinds of ARMA processes is presented. An interesting fact is that the estimator "badness", obtained when using fixed gain, increases the "badness" of overparametrization. Model order selection simulations involving AR processes illustrate this fact very clearly. A successful model order selection simulation for ARMA processes is presented.

A change-point detection method for certain kinds of ARMA processes is obtained for time variant change-points. Also, the abrupt jump parameter case, and change-point detection using undermodeling are considered. A novel result is that undermodeling could in many cases improve the performance of the change-point detection scheme. Some results of the analysis of the change-point detection scheme are obtained and extensive simulations show that the approach exhibits surprisingly good detection capabilities.

Lastly, we prove that the original form of the adaptive controller for linear time invariant systems, as obtained in [Ger90a], can be computed in a much less expensive manner. Simulations for an ARX system exemplify the stability and tracking capability of the adaptive controller. Moreover, the effect of dithering on closed loop performance is illustrated.

Résumé

Cette thèse présente une méthode pour le choix de l' ordre du modèle et la détection de point de changement en temps réel. On utilise une forme de compléxité stochastique prédictive. Une méthode consistante pour trouver le meilleur ordre du modèle pour certains processus ARMA est élaborée. Il est particulièrement intéressant de constater que le "mauvais-rendement" de l'estimateur à gain fixe augmente le "mauvais-rendement" de la sur-paramétrization. Ce fait est clairement illustré par des simulations de choix d'ordre du modèle, impliquant des processus AR. Une simulation réussie du choix d'ordre du modèle pour processus ARMA est aussi présentée.

Une méthode de détection de point de changement pour certains processus ARMA, pour points de changement temporellement variants, est obtenue. Le cas paramétrique d'un saut brusque et la détection de point de changement, utilisant un modèle de degré inférieur, sont également considérés. Quelques résultats partiels de l'analyse de l'algorithme de détection de point de changement sont obtenus et les simulations montrent que l'approche bénéficie de trés bonnes capacités de détection.

Finalement, nous montrons que la forme originale du contrôleur adaptatif pour les systèmes linéaires en temps invariant, comme ceux obtenus dans [Ger90a], peuvent être calculés de manière beaucoup plus efficace. Des simulations pour un système ARX montrent la stabilité et la capacité de suivi du contrôleur adaptatif. De plus l'effet de perturbation sur les performances du système en boucle fermée est examinée.

Acknowledgements

No person is an island, and even though we might sometimes feel as though we are, no graduate student is an island either.

With deep appreciation I would like to thank the following people for their direct and indirect contributions to the work of this thesis.

I begin by thanking Professor László Gerencsér for his superb technical supervision while on staff at McGill University as a Visiting Professor. Professor Gerencsér was certainly an inspiration and a driving force behind the work of this thesis. I would also like to thank Professor Peter E. Caines for his timely financial support as well and for his hospitality.

I would like to thank the McRCIM group for the excellent facility and work environment and friendship they provided. Some of the members of the group who I particularly wish to thank are Ayman Bedair, Kaouthar Benameur, Jan Binder Jean-Marc Gaubert, Yasmine Ghallab, Lee Iverson, Michael Langer, Lin Lin, Robert Lucyshyn, Ameen Maluf, Chafye Nemri, Carlos Perez, Mark Readman, Nemo Semret, Pierre Tremblay, and Finn Wredenhagen.

I would like to thank Jennifer Quinn, Mindle Levitt and Margarita Sanchez, who time and again turned crisis into a solution with above the call of duty help for which I am indebted. Examples include receiving and responding to SOS FAX's from distant

places and "opening doors" to tranquil retreats. Thanks.

I would like to thank Gius, Cleopatra, and Yiorgos Bayada who were warmhearted and loving while putting up with me when working on my thesis day and night in their home in Cyprus during this summer.

I would like to thank my grandparents, parents, my sister and my brother for their unconditional love, understanding and support.

Finally, there are three special people to whom I feel specially thankful. Walter Diconca whose presence and spicy sense of humor kept me company all along this long journey. Allan Solomon, my brother, whose support and love in the crucial last moments of this work gave me the strength to complete it in due time. And to Tonia Bayada who had to live with stochastic complexity every single day and always responded with deep and genuine love.

Claim of Originality

The following original contributions were made:

- Successful implementation of the stochastic complexity based model order selection results for ARMA systems.
- The use of predictive stochastic complexity to solve change-point detection problems: Work includes design, partial theoretical analysis, and extensive computer experimentation.
- The capability of the change-point detection scheme to detect slowly time varying change-points.
- Empirical investigation of the issue of undermodeling in change-point detection problems showing that undermodeling could in fact improve the performace of change-point detection methods.
- The reduction of the computational complexity of a Ljung scheme based adaptive controller: Work includes theoretical proof as well as computer simulations of the controller.

Contents

Abstract ii

Résumé iii

Acknowledgements iv

Claim of Originality vi

1 Introduction 1

1.1 Thesis Outline 2

2 Preliminary Material 6

2.1 The Classical Maximum Likelihood Method 7

2.2 A Major Limitation of the Classical Maximum Likelihood Method . . 12

2.3 Information and Coding Theory 15

2.4 L-Mixing Processes 21

3 Prediction Error Method for ARMA Processes 31

3.1 PEM: Off-Line Case 32

3.1.1 Off-Line PEM with Fixed Gain 35

3.2 PEM: On-Line Case 36

3.2.1 On-Line PEM with Fixed Gain 40

3.3 On-Line and Off-Line PEM's Link 41
3.4 A Simulation of the Recursive PEM 41

4 Model Order Selection 49
4.1 Introduction 49
4.2 Stochastic Complexity 58
4.2.1 Predictive Stochastic Complexity 67
4.3 Model Order Selection for ARMA Models using Predictive Stochastic Complexity 73
4.3.1 Technical Conditions and Main Theorems 74
4.3.2 Selecting the Best ARMA(p, q) Model 76
4.3.3 AR Model Order Selection Simulations 80
4.3.4 ARMA Model Order Selection Simulation 85
4.3.5 Simulation Example of Parameter Versus Model Order Uncertainty 91

5 Change-Point Detection 92
5.1 Introduction 92
5.2 First Elementary Detection Methods 95
5.3 Change-Point Detection for Signals 97
5.3.1 The Off-Line Mathematical Formulation 97
5.3.2 Change-Point Detection and Model Complexity 99
5.3.3 Change-Point Detection with Known Models 100
5.3.4 On-Line Versus Off-Line Procedures 101
5.3.5 Bayesian Versus Non-Bayesian Formulations 103
5.3.6 Change-Point Detection and Optimality 103
5.3.7 Change-Point Detection with Unknown Model Parameters . . 108
5.3.8 Change-Point Detection in Very Complex Situations 111

5.4 Failure Detection for Dynamical Systems 112
5.4.1 Modeling of Failures for Dynamical Systems 114
5.4.2 The Parity Space Approach 116
5.4.3 The Kalman Filter Based Approach 118
5.4.4 The Detection Filter Approach 121
5.5 Predictive Stochastic Complexity Applied to Change-Point Detection for ARMA Systems 123
5.5.1 The Mathematical Model 124
5.5.2 The Encoding Procedure 126
5.5.3 Change-Point Detection as Model Selection 127
5.5.4 Analysis of the Change-Point Detection Method 130
5.5.5 Change-Point Detection and Undermodeling 135
5.6 Change-Point Detection Simulations 138
5.6.1 Slowly Time Variant Change-Point Simulation 139
5.6.2 Performance of the Change-Point Detection Algorithm with Respect to the Fixed Gain λ 143
5.6.3 Jump Change-Point Detection Simulation 150
5.6.4 Change-Point Detection and Undermodeling 153
5.6.5 The Detector $d(N)$ Versus a "naive" Detector Based on Monitoring Parameter Estimates 157

6 Adaptive Control of an LSS **160**
6.1 Closed Loop Identifiability from Open Loop Identifiability 162
6.2 Closed Loop Identification via Ljung's Scheme 166
6.3 An Application: Adaptive Control of an ARX System 170
6.3.1 Simulation of the Adaptive Control of an ARX System 172
6.3.2 The Effect of the Dither 176

7 Conclusion 180

Chapter 1

Introduction

A crucial consideration when modeling phenomena from observed data, is the relationship and tradeoff between model complexity and model fit. For instance, a complex model might be able to match the data with high precision. However, if the description of the model itself (in some well specified sense) turns out to be as lengthy as that of the data, then no overall reduction in the complexity of the original string of data would be achieved. If this type of procedure is employed, it will certainly defeat the modeling objective of striving to maximally reduce the complexity of events by extracting their regular features using compact mathematical descriptions.

The stochastic complexity modeling theory (c.f., [Ris89]) takes both model complexity and performance into account in a natural way. At present, this theory provides one of the most respected methods for solving model selection problems. Moreover, it has by now evolved from its original framework, and is being used as a very important tool in change-point detection (c.f., [BG90], [GB91], [BG92a], and [BG92b]) and adaptive control problems (c.f., [Ger91c]).

The *stochastic complexity* of a set of data is the shortest code length that can be achieved when encoding the data with models in a given model class. This theoretical information measure is computed off-line making it impractical for a large class of real

problems. A major innovation of the theory has been the introduction of the concept of predictive stochastic complexity in [Ris86] which is a real time approximation of the stochastic complexity which depends on an estimation algorithm. The aim of this thesis is to implement, test and refine stochastic complexity based model selection and change-point detection methods in real time using a form of predictive stochastic complexity. These methods make extensive use of the theoretical results developed by Gerencsér and Rissanen (c.f., e.g., [Ris89], [GR91], [Ger91c]) in the area of stochastic complexity.

1.1 Thesis Outline

We shall shortly recapitulate the salient features of this dissertation whose body is divided into five chapters. Chapter 4, on model order selection, Chapter 5, on change-point detection, and Chapter 5 on the adaptive control of a linear stochastic system can be followed independently of each other.

In Chapter 2, we shall present some preliminary material which we feel is essential for an understanding of the stochastic complexity based method for model order selection and change-point detection, to be developed in this dissertation. These topics are the classical maximum likelihood method, the theory of information and coding, and a class of weakly dependent processes known as L-mixing processes (c.f., [Ger89c]). These types of mixing processes are found to play a central role in the asymptotic theory of a broad class of estimator processes.

Since the model selection, change–point detection, and adaptive control methods to be presented in this thesis rely heavily on the prediction-error method for the identification of parameters in stochastic systems (c.f., e.g., [Lju87]), Chapter 3 will present this estimation algorithm for the particular case of ARMA processes. Both its off-line and on-line versions will be introduced, as well as its variant, obtained when

using forgetting of past data with exponential rate. A simulation will be included to illustrate this identification procedure.

Chapter 4 will be initiated by a short discussion around some of the foundations of the modeling problem. The purpose is to place into perspective the stochastic complexity modeling theory, elaborated by Rissanen (c.f., [Ris89]). This theory serves as inspiration and guidance to much of the material to be presented in this dissertation. The stochastic complexity theory will be presented up to some extent, leading us to the notion of stochastic complexity for a set of data. An on-line approximation of it called predictive stochastic complexity (c.f., [Ris86]) will be described in some detail. It will be shown that predictive stochastic complexity is a mathematically well understood criterion which can be used to solve model selection problems in real time. (This on-line feature distinguishes the modeling methods derived using predictive stochastic complexity from other modeling method such as AIC or BIC which are inherently off-line.) Specifically, we will introduce a method for finding the best model order for a set of data among ARMA models of different order. We shall show by computer experiments that this modeling scheme is consistent for certain types of ARMA models; thus validating the theoretical claim found in [GR91]. Moreover, using fixed gain in the prediction-error estimation procedure, the sensitivity of the criterion to overmodeling increases qualitatively. This fact will be demonstrated by simulations which compute the best AR model for a given set of data. A model order selection simulation involving ARMA models will also be presented. Furthermore, we shall study the effect of parameter uncertainty versus model order uncertainty via simulation.

In Chapter 5, we will be concerned with a general outline of the change-point detection problem, and a number of frequently used change-point detection techniques (c.f., e.g., [Bas88]). We will begin by reviewing some of the well-known work done in the last 20 years in the area of change-point detection. One goal is to underline the

general problem formulation, and highlight some of its most essential features and difficulties. The survey should also serve as a self-contained introduction to the topic aiding readers unfamiliar with the area in understanding the change-point detection method in general, and our change-point detection scheme which is inspired by and based on the stochastic complexity theory. The stochastic complexity based change-point detection method to be developed in this thesis is intended for use with ARMA systems under the assumption that they have a slow and non-decaying drift after the change-point occurs. A salient feature is that the resulting change-point detection algorithm will be finally expressed in terms of fairly simple recursive equations. The abrupt jump parameter case and change-point detection with undermodeling will be also investigated. A novel result is that undermodeling—which is not treated in the literature of change-point detection—could in many cases improve the performance of the change-point detection scheme. Some partial results on the analysis of the scheme are obtained, showing that these methods are amenable to theoretical analysis. Moreover, simulations will show that the approaches exhibit surprisingly good detection capabilities. The simulations will include the issue of robustness of the change-point detection method with respect to the fixed gain of the prediction error algorithm, the improvement of performance when using undermodeling, and the comparison of the method with a "naive" procedure based on unprocessed parameter estimates.

In Chapter 6, the adaptive control problem for finite dimensional time invariant linear stochastic systems, as introduced in [Ger90a], will be described. One of the main computationally expensive features when implementing the adaptive controller of [Ger90a] is its dependence on the directional derivatives of the adaptive feedback transfer function gain. In this thesis, we will prove that in fact there is no such dependence, reducing considerably the computational complexity of the algorithm. We will also extensively illustrate this adaptive control methodology for an ARX system, showing the stability and tracking capability of the adaptive controller. Moreover,

the effect on the closed loop performance caused by the dither process, which is embedded in the controller to guarantee identifiability of the closed loop system, will be studied via simulations.

The final chapter, Chapter 7, will end with some concluding remarks about the contributions of this thesis along with an indication of possible areas of further research.

Chapter 2

Preliminary Material

In this Chapter we shall present some basic topics which we feel are essential for an understanding of the stochastic complexity based method for model order selection and change-point detection, which will be developed in subsequent sections. Thus in Section 2.1 the classical maximum likelihood method will be presented, while in Section 2.2 one of its main limitations—its inability to deal with models of different complexity—will be described. This limitation is overcome by the stochastic complexity theory in Chapter 4. Since one of the pillars of this theory is the theory of information and coding, we shall shortly introduce it in Section 2.3. Lastly in Section 2.4 a class of weakly dependent stochastic processes known as L-mixing processes, which play a central role in the asymptotic theory of a broad class of estimator processes, will be presented.

Some of these topics, such as the maximum likelihood method and the theory of information and coding, are well known in the scientific community. Nevertheless, the short introductions which follow will familiarize the reader with the notation of the thesis and help facilitate the understanding of the present dissertation.

2.1 The Classical Maximum Likelihood Method

The maximum likelihood (ML) method is unquestionably the most widespread estimation technique in both the theoretical and practical realms of the statistical discipline (c.f., [III81]). Since a vast literature is definitely available, the purpose here, beyond that of a brief account of the method itself, is to serve as an introduction to some of the main ideas of the modeling strategies that are extensively employed in subsequent sections.

Let the sequence of real numbers $y_1, \ldots, y_N$ represent a set of observed data obtained from a certain experiment. Often, sequences like $y_1, \ldots, y_N$ will be shortly denoted y^N. Now, based on physical considerations—or any other pertinent factor—the following *modeling* assumption is usually made:

Let $y_1, \ldots, y_N$ be a sequence of independent and identically distributed (i.i.d.) real-valued random variables each having density $f(\cdot, \theta_0)$, where $\theta_0 \in D \subset \mathbb{R}^k$, D an open domain, and $\mathbb{R}^k$ the k-dimensional Euclidean space. Then y^N is *assumed* to be a realization of the random variables y^N.

What the above assumption says is that y^N is an outcome of the joint density $\prod_{n=1}^{N} f(\cdot, \theta_0)$, and that this density is the "true" density governing the set of data $y_1, \ldots, y_N$. Hence $\prod_{n=1}^{N} f(\cdot, \theta_0)$ is the "true" model defining all the constraints experienced by the data. (Note that the above modeling set-up falls into a parametric type of modeling and the specification of θ_0 completely describes the "true" density.) The role of any identification method is thus to provide a good estimate, in some well specified sense, of the unknown "true" parameter θ_0. By doing so one obtains an approximation of the "true" density.

Although this type of modeling assumption characterize by the notion of "true"

models is of common practice in classical statistical inference, it is the source of one of its major weaknesses. This will mainly be addressed and discussed in Chapter 4.

The idea behind the ML method of estimation is to approximate the "true" density by the density that makes the data string $\mathbf{y}^N$ the most probable among the class of joint densities $\{\prod_{n=1}^N f(\cdot,\theta), \theta \in D\}$. This view is rooted in the belief that, at least for large data samples, the data must be a "good" representative of its assumed but unknown generating "true" density. More precisely, that the mass of the "true" density is predominately concentrated at the given observed data $\mathbf{y}^N$.

For a data set $\mathbf{y}^N$, define the likelihood function as

$$\prod_{n=1}^{N} f(\mathbf{y}_n,\theta) : D_0 \to \mathbb{R}_+, \tag{2.1}$$

where $D_0 \subset D$ is compact. This function expresses the likelihood of getting the data set $\mathbf{y}^N$ if the true density were specified by θ. Now define the estimate

$$\widehat{\theta}_N = \arg\max_{\theta \in D_0} \prod_{n=1}^{N} f(\mathbf{y}_n,\theta).$$

$\widehat{\theta}_N$ is called the ML estimator (MLE) of θ^*. The best approximation of the "true" density in the ML sense is then given by $f(\cdot,\widehat{\theta}_N)$.

Note that the likelihood function, defined by (2.1), is a deterministic function of its argument θ. Nevertheless, it is of common practice to consider the likelihood function to also be a function of all of the possible data sequences allowed by the assumed joint distribution. Thus one generally writes $\prod_{n=1}^N f(y_n,\theta)$, which is a random variable for any fixed $\theta \in D_0$. Under this framework, one could analyze various properties of the method itself independently of the actual data $\mathbf{y}^N$. In the sequel, only when we would like to stress the fact that y^N actually represents just a tentative model for the observed data, shall the notation $\mathbf{y}^N$ be alternatively used.

The analysis of the ML estimator is based on what we shall call the "up-bottom" approach, an approach which has been shown to be a powerful tool for both the

analysis and development of identification methods (c.f., [Ger89c]). It is named "up-bottom" since it is based on first finding a non-computable but analytically tractable solution to the identification problem and then approximating this solution by a computable one. More precisely, the approach consists of finding an asymptotic cost function with the property of having an absolute minimum at the "true" parameter $\mathring{\theta}$. Then one "goes down" and finds a good computable approximation of that asymptotic cost function. This is usually done by resorting to some kind of law of large numbers.

As will soon be clear, it is much more convenient to do the analysis of the ML method using

$$L(y^N, \theta) = -\sum_{n=1}^{N} \log f(y_i, \theta),$$

which is called the negative log-likelihood function. Moreover, note that

$$\mathbb{E}\, L(y_n, \theta) = \mathbb{E}\, L(y_m, \theta) \qquad \text{for} \quad n, m \in [1, N],$$

by application of the independence hypothesis exhibited by the sequence y^N. Thus, without loss of generality, we can work with $L(y_1, \theta)$.

For the first proposition of the ML method, we need the first partial derivatives of $L(y_1, \theta)$ with respect to θ, i.e.,

$$L_\theta(y_1, \theta) \triangleq \frac{\partial}{\partial \theta} L(y_1, \theta) = (\partial/\partial\theta) f(y_1, \theta)/f(y_1, \theta).$$

Henceforth, the partial derivate of any function with respect to one of its arguments will be denoted as above, i.e., $h_x(x, y) = (\partial/\partial x) h(x, y)$.

Proposition 2.1.1 *We have*

$$\mathbb{E}\, L_\theta(y_1, \theta) \Big|_{\theta=\mathring{\theta}} = 0.$$

PROOF. Clearly

$$\mathbb{E}\, L_\theta(y_1, \mathring{\theta}) = \mathbb{E}\, \frac{f_\theta(y_1, \mathring{\theta})}{f(y_1, \mathring{\theta})}. \tag{2.2}$$

The left hand side of (2.2) can be written as

$$\begin{aligned}\int \frac{f_\theta(x,\dot{\theta})}{f(x,\dot{\theta})} f(x,\dot{\theta})dx &= \int f_\theta(x,\dot{\theta})dx \\ &= \frac{\partial}{\partial \theta}\int f(x,\dot{\theta})dx,\end{aligned}$$

and since $\int f(x,\dot{\theta})dx = 1$, we get the claim of the proposition. ∎

Let us now introduce the asymptotic cost function

$$W(\theta) = \mathbb{E}\, L(y_1,\theta).$$

Then, based on Proposition 2.1.1 we get

$$W_\theta(\theta)\Big|_{\theta=\dot{\theta}} = 0.$$

Now, the partial derivative with respect to θ of the gradient process $W_\theta(\theta)$ is given by

$$W_{\theta\theta}(\theta) = \mathbb{E}\, f_\theta(y_1,\theta)\cdot f_\theta^T(y_1,\theta)/f^2(y_1,\theta).$$

Proposition 2.1.2 *$W_{\theta\theta}(\dot{\theta})$ is symmetric and positive semi-definite.*

REMARK. $I(\dot{\theta}) = W_{\theta\theta}(\dot{\theta})$ is called the Fisher information matrix. Since $I(\dot{\theta})$ is the slope of $W_\theta(\theta)$ at $\dot{\theta}$, it provides information about the accuracy of the estimator $\hat{\theta}_N$.

Propositions 2.1.1 and 2.1.2 imply that the cost function $W(\theta)$ is locally minimized at $\dot{\theta}$. The so-called "up" part of the approach has thus been completed. What remains is to find a computable solution of the deterministic function $W(\theta)$. (Observe that $W(\theta)$ is not computable since it involves an expectation and in practice only y^N is available. Note also that if we assume the validity of the modeling assumptions, then y^N represents just one of the many possible realizations of the sequence of random variables y^N.) With the help of the law of large numbers we can arrive at a computable approximation for the cost function $W(\theta)$.

Proposition 2.1.3 *Under some suitable conditions, (c.f., [III81]), we have*

$$\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} L(y_n,\theta) = W(\theta), \quad \textit{and} \quad \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} L_\theta(y_n,\theta) = W_\theta(\theta), \tag{2.3}$$

with probability 1.

Propositons 2.1.1–2.1.3 provide a solid theoretical justification for the ML method. Indeed, if the data is a realization of the assumed class of densities, then except on a set of probability zero, we have

$$\mathring{\theta} = \lim_{N\to\infty} \arg\max_{\theta\in D} \frac{1}{N}\sum_{n=1}^{N} L(\mathrm{y}_n,\theta).$$

The asymptotic properties of the ML estimator are captured by the next two propositions.

Proposition 2.1.4 *Under some suitable conditions, (c.f., [III81]), we have*

$$\hat{\theta}_N - \mathring{\theta} = -\frac{1}{N} I^{-1}(\mathring{\theta}) L_\theta(y^N,\mathring{\theta}) + O_M(1/N).$$

where $O_M(\cdot)$ is defined in 2.4.1.

(Hints: Make a Taylor series expansion of $(1/N)L_\theta(y^N,\theta)$ about $\hat{\theta}_N$ and evaluate at $\mathring{\theta}$. Use the i.i.d. assumption and the law of large numbers for the second term in the Taylor expansion. Finally apply the central limit theorem.)

Combining Propositions 2.1.3 and 2.1.4, it is immediate that the MLE $\hat{\theta}_N$ converges to the "true" parameter $\mathring{\theta}$ with probability 1. Therefore, the ML method for i.i.d random variables is strongly consistent.

The last proposition provides an estimate of the variance of the estimator.

Proposition 2.1.5 *We have under some suitable conditions, (c.f., [IH81]),*

$$\mathbb{E}\,(\hat{\theta}_N - \mathring{\theta})(\hat{\theta}_N - \mathring{\theta})^T = \frac{1}{N} I^{-1}(\mathring{\theta}) + O(1/N).$$

Observe that the higher the Fisher information matrix, the lower the variance of the MLE estimator, resulting in better estimation. Also Proposition 2.1.5 shows that the MLE asymptotically achieves the well-known Cramer-Rao lower bound for the variance of any estimator (c.f., [Cai88]).

For the sake of simplicity of exposition we have limited the scope of the ML method to independent random variables. However all previous results can be properly generalized for certain types of dependent random variables like those generated as outputs of Gaussian ARMA processes (c.f., [Cai88]). In this more general case the joint density of the random variables y^N loses the pleasant structure—that of being the direct product of the density of the simple random variables—and thus the analysis becomes more involved.

2.2 A Major Limitation of the Classical Maximum Likelihood Method

The classical ML method admits the comparison of different parameter-values of a given parametric model class (see Section 2.1). That is, its application is confined to parametric models with known dimension. It is certainly advantageous to investigate the applicability of the ML method to cases in which the parametric dimension of the model is not assumed to be known a-priori. This shall be the main objective of this section. It will be shown that a direct, or in other words naive, use of the classical ML method is not fit to tackle these more complex modeling problems.

Let us continue with a similar set-up as to that of Section 2.1, but with the distinction that the parametric dimension of the "true" model class is unknown. We then need to first modify the notation slightly so as to cover this more general case.

The "true" model order is denoted by $\mathring{k}$, the "true" parameter by $\mathring{\theta}^{\mathring{k}}$, and the domain where $\mathring{\theta}^{\mathring{k}}$ belongs by $D^{\mathring{k}}$. This type of notation will be used repeatedly in this thesis when the dimension of the model is not known. Otherwise, when working in the context of a known model order, the previous simpler notation will be maintained.

We shall only study the case of over-parametrization. Thus the classes considered will be of the form $\{f(\cdot,\theta^k), \theta^k \in D^k \subset \mathbb{R}^k, k \geq \mathring{k}\}$. (We only cover this case for simplicity of exposition since otherwise the "true" model will not belong to the set of tentative model classes, and as a result the problem would become more complex.)

Note that a direct application of the ML method to the case of overparametrized model classes indexed by k leads us to the following definition for the ML estimator:

$$\hat{\theta}_N^k = \arg\min_{\theta^k \in D^k} L(\mathrm{y}^N, \theta^k). \tag{2.4}$$

What we shall then call the "naive" formulation of the ML method is the claim that the "goodness" of a model class with respect to a given data set y^N can be captured by the likelihood function.

Before introducing the next proposition, we would like to stress that the parameter $\mathring{\theta}^k$ is obtained by adjoining $k - \mathring{k}$ zeros to the assumed true parameter $\mathring{\theta}^{\mathring{k}}$.

Proposition 2.2.1 *We have*

$$\lim_{N\to\infty} \mathbb{E}\left(L(y^N, \hat{\theta}_N^k) - L(y^N, \mathring{\theta}^k)\right) = -\frac{1}{2}\dim\mathring{\theta}^k.$$

(Hint: Make a Taylor series expansion about $\hat{\theta}_N^k$ and used Proposition 2.1.5.)

Since $L(y^N, \mathring{\theta}^k) = L(y^N, \mathring{\theta}^{\mathring{k}})$ is independent of the model order k, Proposition 2.2.1 implies that $L(y^N, \hat{\theta}_N^k)$ does not penalize over-parametrization. Indeed if $k > k' > \mathring{k}$ then

$$L(\mathrm{y}^N, \hat{\theta}_N^k) < L(\mathrm{y}^N, \hat{\theta}_N^{k'}). \tag{2.5}$$

Therefore, in this case, the "naive" model selection approach will wrongly conclude that the k-th model class would be preferred in the ML sense over the k'-th model class. Thus the ML method fails to be a proper choice for a universal model selection criterion. Since modeling is basically concerned with finding data constraints that will shorten the representation of the original data set, we are forced to reject this "naive" approach since it contradicts the spirit of modeling.

For the sake of clarity, let us illustrate Proposition 2.2.1 by a simple linear regression example. Assume that the "true" model for the random variables y^N is given by the regression model

$$y_n = x_n^T \bar{\theta}^{\bar{k}} + e_n \qquad x_1 = 0, \tag{2.6}$$

where (e_n) is an i.i.d. random process with density $\mathcal{N}(0, \sigma)$. Consider the model classes described by

$$y_n = x_n^T \theta^k + e_n \qquad x_1 = 0, \tag{2.7}$$

where $\dim \theta^k = k \geq \bar{k}$. Note that by adjoining the proper number of zeros to the vector $\bar{\theta}^{\bar{k}}$ one obtains $\bar{\theta}^k$, which is the parameter value in the k-th model class which makes (2.7) a "perfect" fit. For some $\theta^k \in D^k$ define the prediction error process

$$\epsilon_n(\theta^k) = y_n - x_n^T \theta^k.$$

Then, it is straightforward to see that

$$L(y^N, \theta^k) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(\epsilon_n(\theta^k)\right)^2 \quad \text{and} \quad L(y^N, \bar{\theta}^k) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (e_n)^2.$$

Now let us compute the MLE of $\bar{\theta}^k$. Observe that in the regression case the MLE $\hat{\theta}_N^k$ coincides with the LSE (least square estimate). Now applying Proposition 2.2.1 one gets

$$\lim_{N \to \infty} \mathbb{E} \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(\left(\epsilon_n^k(\hat{\theta}_N^k)\right)^2 - (e_n)^2 \right) = -\frac{1}{2} \dim \bar{\theta}^k. \tag{2.8}$$

According to (2.8) and since e^N is independent of k, the more parameters we use, the smaller the cumulative square prediction error process $\sum_{n=1}^{N} \left(\epsilon_n^k(\hat{\theta}_N^k)\right)^2$, or the

greater the likelihood of the data y^N! Thus the "naive" ML method fails to penalize over-parametrization.

It is interesting to note however, that the second moment of the ML estimator is sensitive to over-parametrization. The next proposition shows that the covariance of the MLE penalizes over-parametrization.

Proposition 2.2.2 *We have for* $k \geq \overset{*}{k}$

$$\mathrm{Cov}(\hat{\theta}_N^k - \bar{\theta}^k) \geq \mathrm{Cov}(\hat{\theta}_N^{\overset{*}{k}} - \bar{\theta}^{\overset{*}{k}}).$$

(Hint: Apply the matrix inversion lemma to the matrix $\mathrm{Cov}(\hat{\theta}_N^k - \bar{\theta}^k)$.)

The above proposition shows that it is not difficult to find other criteria that will succeed in penalizing over-parametrization. However, this and other methods usually lack the appealing interpretation of the classical ML method. We shall then seek a method that will not fail the over-parametrization test, and for that matter any existing test, but that will recapture, in a sense, the essence of the ML method. This shall be provided by the stochastic complexity theory which will be presented in Chapter 4. In order to lay down the main concepts of this theory we need to look at some of the basic elements of the information and coding theory.

2.3 Information and Coding Theory

The origin of information theory dates back to the pioneering work of Shannon in the late 40's (c.f., [Abr63]). His initial motivation was the engineering problem of how to transmit information through noisy channels. Nevertheless, he was quick to realize the implications of his theory much beyond his original problem formulation. Since then, the theory of information has had an impact on a variety of fields such as

linguistics, semantics, psychology, biology, economics, music and the arts, and even philosophy. More recently, the theory of information has been used by Rissanen as the starting point for the stochastic complexity theory (c.f., [Ris78]), a general theory for extracting models from data. The natural link between the stochastic complexity and information theories is the main reason that we are interested in understanding the foundations of the latter theory.

In order to introduce the basic ideas of information theory, we first need to bring forward some of the elementary notions of coding theory.

Definition 2.3.1 Let $\mathcal{A}$ be a finite alphabet, $\mathcal{B}$ a finite set of words composed of combinations of the binary symbols $\{0,1\}$, and $C : \mathcal{A} \rightarrow \mathcal{B}$ a one-to-one mapping. Then $\mathcal{A}$ is called the *source alphabet*, $C(\cdot)$ the *code*, $\mathcal{B}$ the *code alphabet*, and the elements of $\mathcal{B}$ *codewords*.

Note that since $C(\cdot)$ is defined as a one-to-one mapping, the code is nonsingular (i.e., all codes are different), and uniquely decodable. These are properties which are clearly indispensable for any proper code.

Let $l : \mathcal{B} \rightarrow \mathbb{N}$ be the mapping associating each element of $\mathcal{B}$ with the length of its corresponding codeword as expressed by adding the number of the codeword binary digit representation. Now define the composed mapping $L = lo C$, and let

$$L_{\max} \triangleq \max_{a \in \mathcal{A}} L(a).$$

Denote $S(a)$ the set of all nodes which are extensions of a at level $L_{\max}$.

To guarantee that codewords can be decoded as they are recieved without using any bits from subsequent codewords, a further condition has to be imposed on the code $C(\cdot)$.

Definition 2.3.2 A code $C(\cdot)$ is said to be a *prefix code* if and only if for all $a, a' \in \mathcal{A}, a \neq a'$, $S(a) \cap S(a') = \emptyset$.

The important feature about prefix codes is that they can be instantaneously decoded while codewords are received, meaning that there is no need, for example, to use separating commas between codewords. Unless otherwise specified, codes $C(\cdot)$ will be considered to be prefix codes.

Prefix codes can be shown to be constrained in the size of their corresponding codewords: they cannot be made arbitrarily small as expressed by the well-known inequality due to Kraft in 1949.

Proposition 2.3.1 (Kraft Inequality) *If $C(\cdot)$ is a prefix code then*

$$\sum_{a \in \mathcal{A}} 2^{-L(a)} \leq 1.$$

(Hint: Use the disjoint property of prefix codes.)

The converse of Proposition 2.3.1 is also true.

Proposition 2.3.2 *Given $L(\cdot)$ satisfying the Kraft inequality, there exists a prefix code $C(\cdot)$ with codelength $L(\cdot)$*
(Hint: Arrange $L(a)$'s in increasing order, and use lexicographic ordering (c.f., [Abr63]).)

In general, it is not difficult to obtain many different types of code mappings $C(\cdot)$ for any given source $\mathcal{A}$ even if they are constrained to be prefix codes. Therefore, a criterion is needed to choose among the numerous available prefix codes. Obviously one desires the prefix codes to have associated short codelengths. It is at this crucial point that some sort of probabilistic model for the emission of the source symbols in $\mathcal{A}$ is most useful. Doing so will allow for a meaningful definition of coding optimality.

The frequency of occurrence of each letter in the source alphabet can be assumed to define a probability distribution for that alphabet. The idea is then to assign short codewords to letters with a high probability of occurrence and conversely, long codewords to letters with a low probability of occurrence. Let us assume that $p(\cdot)$

represents a "good" probabilistic model for the distribution of the letters of an alphabet $\mathcal{A}$ with respect to a particular language. Based on this probabilistic model of the source $\mathcal{A}$, the optimal coding is defined as follows:

Definition 2.3.3 Let the *average codelength* of a prefix code $C(\cdot)$ be

$$\overline{L} = \sum_{a \in \mathcal{A}} p(a) L(a).$$

Then, the *optimal coding* is defined as

$$C_{\text{opt}} = \arg \min_{C \text{ prefix}} \overline{L}.$$

Note that $C_{\text{opt}}(\cdot)$ strongly depends on the assumed underlying probability distribution $p(\cdot)$.

Shannon defined the *informative value* of a letter $a \in \mathcal{A}$ as

$$I(a) = \log\left(1/p(a)\right). \tag{2.9}$$

Observe from (2.9) that if a letter is very unlikely, then its information content will be high implying a large cost of removing uncertainty. This sort of connection gives a meaningful physical interpretation to the information measure $I(\cdot)$.

The main step in the search for the optimal coding $C_{\text{opt}}(\cdot)$ is the following proposition due to Shannon.

Proposition 2.3.3 *Given any finite or countable alphabet $\mathcal{A}$, with probability distribution $p(\cdot)$ over $\mathcal{A}$, we have for any prefix code with codelength $L(\cdot)$ the Shannon inequality*

$$\sum_{a \in \mathcal{A}} p(a) L(a) \geq -\sum_{a \in \mathcal{A}} p(a) \log p(a) = \sum_{a \in \mathcal{A}} p(a) I(a). \tag{2.10}$$

Let us define the entropy of an information source $\mathcal{A}$ as

$$H(p) \triangleq -\sum_{a \in \mathcal{A}} p(a) \log p(a).$$

$H(p)$ can be interpreted as a measure of the level of disorder of the symbols of $\mathcal{A}$. For example, one can show that if the probability distribution is evenly distributed—that is, source symbols are equiprobable—then the resulting entropy is maximized (c.f., [Abr63]). Note that from Shannon's inequality, $H(p)$ is the lower bound beyond which no prefix code $C(\cdot)$ can exceed.

REMARK. Although Proposition 2.3.3 is non-constructive, it does provide two very important features:

i) a universal yardstick with the help of which, in principle, the information inherent in different sources can be compared by means of the asymptotic lower bound $H(p)$;

ii) a connection between the information measure and the codelength since $C_{\text{opt}}(\cdot) = I(\cdot)$, which provides an important interpretation for the information measure itself.

The proof of the Shannon inequality, is mainly based on the following important inequality:

Proposition 2.3.4 (Kullback-Leibler inequality) *Given any two distributions $f(x)$ and $g(x)$ over a finite or countable set $\mathcal{X}$, then*

$$-\sum_{x\in\mathcal{X}} f(x)\log g(x) \geq -\sum_{x\in\mathcal{X}} f(x)\log f(x).$$

(Hint: Use the concave property of the log function).

REMARK. The Kullback-Leibler measure

$$M(g,f) = -\sum_{x\in\mathcal{X}} f(x)\log g(x) + \sum_{x\in\mathcal{X}} f(x)\log f(x),$$

which computes a distance between two probability distributions, represents the main contribution of the theory of information and coding to the field of statistics.

Note that the Kullback-Leibler inequality also holds for continuous variables. Let ξ be a random variable over the set $\mathcal{X}$ with density function $f(x)$, then if $g(x)$ is any other density function for ξ we have

$$-\int_{\mathcal{X}} f(x)\log g(x)dx \geq -\int_{\mathcal{X}} f(x)\log f(x)dx.$$

Since $-\log f(x)$ achieves the lower bound of the Shannon inequality, it can be viewed as a sort of optimal "codelength" in the continuous case. Is it possible to achieve the lower bound $H(p)$ in the discrete case? If we set

$$L(x) = [-\log_2 f(x)] + 1,$$

where $[b]$ denotes the integer part of b, then clearly $L(x)$ satisfies the Kraft inequality. Thus, there exists a prefix code with codelength $L(x)$. Moreover, we have

$$\begin{aligned}\sum_{x\in\mathcal{X}} f(x)L(x) + \sum_{x\in\mathcal{X}} f(x)\log_2 f(x) &\leq \sum_{x\in\mathcal{X}} f(x)(L(x) + \log_2 f(x)) \\ &= \sum_{x\in\mathcal{X}} f(x) = 1,\end{aligned}$$

which says that the codelength $L(x)$ differs by only one bit from the optimal code. Hence for a large alphabet the lower bound $H(p)$, in practice, can be assumed to have been achieved.

An important observation is that if we have equality in the Kraft inequality, then $2^{-L(x)}$ is a probability distribution. The importance of this distribution is that it can be taken to represent a constructive model for the emission of the symbols $x \in \mathcal{X}$. Note, moreover, that a correspondence between codelengths and probability distributions has been established. The consequences of this correspondence are of paramount importance to the way we interpret modeling, for the optimal code has

a corresponding distribution which can be viewed as the distribution that assigns maximum probability to the observed data.

An illustrative example of this correspondance can be given by recalling that the ML estimator of the "true" parameter of a density governing a sequence of i.i.d. random variables is

$$\widehat{\theta}_N = \arg\min_{\theta \in D} \sum_{n=1}^{N} -\log f(\mathrm{y}_n, \theta). \tag{2.11}$$

Notice that (2.11) also represents a means of finding the shortest description, i.e. shortest codelength, for the data y^N. Therefore, the ML criterion and the search for the shortest codelength coincide when one model class—in this case a class of densities—is given and what is thus left to determine is the value of the parameter in that given class. As we have previously seen, the ML method cannot be directly extended when different model classes are to be compared since we have proved that the method does not penalize overparametrization. In Chapter 4 we will show how to compare different model classes in a way that resembles the ML notion.

2.4 L-Mixing Processes

Any linear rational stable filter would produce a weakly dependent stochastic process if driven by white noise. Since this is one of the most frequently used models for the realization of stochastic processes, (c.f. [Cai88]), weakly dependent processes appear naturally in system identification.

A type of weakly dependent process, known as L-mixing, has been shown to play a fundamental role in the analysis of system identification methods [Ger89c]. This is because all stochastic processes relevant to system identification are L-mixing processes, and moreover they are invariant under the usual operations performed in estimation. An early development of this kind of mixing, referred to as "exponential

stability", can be found in [Lju76] and [RC79]. See also [Cai88]. In this presentation we will mainly follow [Ger89c].

The defining property of a weakly dependent process is that its distant past contributes a negligible information pattern to its process present. Therefore, weakly dependent processes behave much like independent processes when a subsequence with an appropriate lag is extracted from the original weakly dependent sequence of random variables. To clarify these ideas, we introduce the following definitions.

Consider the stochastic process $(x_n(\theta))$ defined on $\mathbb{Z} \times D$, where $\mathbb{Z}$ denotes the set of natural numbers, $D \subset \mathbb{R}^k$. We assume, unless otherwise specified, that $n \geq 0$.

Definition 2.4.1 The stochastic process $(x_n(\theta))$ is said to be *M-bounded* if for all $1 \leq q < \infty$

$$M_q(x) = \sup_{\substack{n\geq 0 \\ \theta \in D}} \mathbb{E}^{1/q}|x_n(\theta)|^q < \infty.$$

If $(x_n(\theta))$ is M-bounded we write $x_n(\theta) = O_M(1)$. Similarly, if c_n is a positive sequence we write $x_n(\theta) = O_M(c_n)$ if $x_n(\theta)/c_n = O_M(1)$.

Definition 2.4.1 extends naturally to the particular case of stochastic processes which do not depend on a parameter, or to those which degenerate into a random variable.

Example 2.4.1 For any stable matrix A and M-bounded process (u_n), the stochastic process (x_n) generated by the state space equation

$$x_{n+1} = Ax_n + Bu_n, \qquad x_0 = 0 \tag{2.12}$$

is M-bounded. (Hint: use the triangular inequality).

We say that a stochastic process (x_n) tends to a random variable x in the M-sense if for all $q \geq 1$ we have

$$\lim_{n\to\infty} \mathbb{E}^{1/q}|x_n - x|^q = 0.$$

Similarly, we can define differentiation in the M-sense.

A stochastic process (x_n) is $\mathcal{F}_n$-adapted if the sets

$$\{(\omega, m) : x_n(\omega) \in B \in \mathcal{B}(\mathbb{R}), m \leq n\}$$

are $\mathcal{F}_n \times \mathcal{B}(\mathbb{R})$ measurable.

Let $(\mathcal{F}_n)$ and $(\mathcal{F}_n^+)$ be families of independent monotone increasing and monotone decreasing σ-algebras respectively. A typical example is provided by the σ-algebras

$$\mathcal{F}_n = \sigma\{e_i : i \leq n\} \qquad \text{and} \qquad \mathcal{F}_n^+ = \sigma\{e_i : i > n\},$$

where (e_i) is an i.i.d. sequence of random variables.

Definition 2.4.2 A stochastic process $(x_n(\theta))$ is said to be *L-mixing with respect to $(\mathcal{F}_n, \mathcal{F}_n^+)$ uniformly in θ* if it is $\mathcal{F}_n$-adapted, M-bounded, and with τ a positive integer and

$$\gamma_q(\tau, x) = \sup_{\substack{n \geq \tau \\ \theta \in D}} \mathbb{E}^{1/q} |x_n(\theta) - \mathbb{E}(x_n(\theta)|\mathcal{F}_{n-\tau}^+)|^q,$$

we have for any $1 \leq q < \infty$

$$\Gamma_q = \Gamma_q(x) = \sum_{\tau=1}^{\infty} \gamma_q(\tau, x) < \infty.$$

The phrase "uniformly in θ" in Definition 2.4.2 is omitted for stochastic processes which do not depend on a parameter. We would like to recall that L-mixing process were first introduced in [Ger89c].

Example 2.4.2 If the input process (u_n) in Example 2.4.1 is an i.i.d. sequence, then the output process (x_n) is L-mixing.

PROOF. Iterating (2.12) we get for $m \leq n$

$$x_n = A^{(n-m)} x_m + \sum_{i=m}^{n} A^{n-i} B u_i.$$

Clearly $x_n - \mathbb{E}(x_n|\mathcal{F}_m^+) = \sum_{i=m}^{n} A^{n-i} B u_i$, and the result follows using the triangular inequality and the fact that A is stable. ∎

Example 2.4.3 Discrete time stationary Gaussian ARMA processes are L-mixing. (Hint: Use a state space representation).

If (x_n) is an L-mixing process, then by definition, taking $m < n$, (x_n) can be approximated by an $\mathcal{F}_m^+$-measurable random variable with an error decreasing exponentially with m. For this reason, it becomes convenient to decompose L-mixing processes as $x_n = x_{n,m}^+ + x_{n,m}^0$, where $x_{n,m}^+ = \mathbb{E}(x_n|\mathcal{F}_m^+)$.

One of the main reasons L-mixing processes are so useful is that they are invariant under the usual operations performed in system identification.

Theorem 2.4.1 *Let (x_n) and (y_n) be L-mixing processes and $c \in \mathbb{R}$, then:*

(a) (cx_n) is L-mixing.

(b) $(x_n + y_n)$ is L-mixing.

(c) $(x_n \cdot y_n)$ is L-mixing.

PROOF. (a) and (b) are trivial. To prove (c) let $m < n$, then for any $1 \leq q < \infty$ we have by the Cauchy-Schwarz and Jensen inequalities

$$\begin{aligned} \|x_n y_n - x_{n,m}^+ y_{n,m}^+\|_q &\leq \|x_n(y_n - y_{n,m}^+)\|_q + \|y_{n,m}^+(x_n - x_{n,m}^+)\|_q \\ &\leq M_{2q}(x)\gamma_{2q}(y, n-m) + M_{2q}(y)\gamma_{2q}(x, n-m). \end{aligned}$$

Since (x_n) and (y_n) are L-mixing the result follows. ∎

Note that property (c) of Theorem 2.4.1 is not shared by other types of mixing processes.

Based on a well known lemma, it is sufficient to find just one $\mathcal{F}_m^+$ measurable random variable which approximates x_n fairly well to verify the L-mixing property.

Lemma 2.4.1 *Let ξ be an M-bounded, $\mathcal{F}$-measurable random variable, and let $\mathcal{F}'$ be some σ-subalgebra of $\mathcal{F}$. Then for any $\mathcal{F}'$-measurable random variable η, and for all $1 \leq q < \infty$, we have*

$$\mathbb{E}^{1/q}|\xi - \mathbb{E}(\xi|\mathcal{F}')|^q \leq 2\mathbb{E}^{1/q}|\xi - \eta|^q.$$

If x_n is L-mixing then a strengthened Hölder-inequality is obtained. Analogous inequalities for uniform mixing stationary sequences are given in [IL71], and for strong mixing stationary sequences in [Dav68].

Lemma 2.4.2 *Let x_n be an L-mixing process such that $Ex_n = 0, \forall n \geq 0$. Let $m < n$ and consider an $\mathcal{F}_m$ measurable M-bounded random variable η. Then for $1 < p \leq \infty$, $1 < q \leq \infty$, such that $(1/p) + (1/q) = 1$*

$$|\mathbb{E}\, x_n\eta| < 2\gamma_q(t-s)M_q(\eta). \tag{2.13}$$

PROOF. Since $x^+_{n,m}$ is independent of $\mathcal{F}_m$, we can write

$$\begin{aligned} \mathbb{E}\, x_n\eta &= \mathbb{E}(x^+_{n,m} + x^0_{n,m})\eta \\ &= \mathbb{E}\, x^+_{n,m}\mathbb{E}\,\eta + \mathbb{E}\, x^0_{n,m}\eta. \end{aligned}$$

Note that $\mathbb{E}\, x^+_{n,m} = -\mathbb{E}\, x^0_{n,m}$, thus

$$|\mathbb{E}\, x_n\eta| \leq |\mathbb{E}\, x^0_{n,m}||\mathbb{E}\,\eta| + \mathbb{E}\,|x^0_{n,m}\eta|. \tag{2.14}$$

Using the monotonicity of the L_q norms for the first term of (2.14), and applying Hölder's inequality to the second term, we get (2.13). ∎

Example 2.4.2 illustrates that L-mixing processes passed through certain types of stable linear filters remain L-mixing. This property of L-mixing processes holds for a general class of stable filters.

Definition 2.4.3 Let a linear filter be described by

$$x_n = \sum_{m=0}^{n} \phi(n,m)u_m$$

where $\phi(n,\cdot)$ is locally in $l_2[0,\infty)$ for all $n \geq 0$. Set

$$\psi(l) = \sup_{n-m=l} |\phi(n,m)|.$$

Then the filter is said to be *stable* if

$$\phi^* = \sum_{l=0}^{\infty} \psi(l) < \infty.$$

Example 2.4.4 The filter in Example 2.4.2 is stable. For the exponential smoothing case, i.e. $A = B = \lambda, \lambda \in \mathbb{R}$, we get $\phi^* = 1$.

Theorem 2.4.2 *The output process of a linear stable filter, stable in the sense of Definition 2.4.3, which satisfies*

$$\phi^{**} = \sum_{l=0}^{\infty} l\,\phi(l) < \infty.$$

and whose input process u_n is L-mixing, is also L-mixing. Moreover, for $1 \leq q < \infty$ we have

$$M_q(x) < \phi^* M_q(u), \qquad \text{and} \qquad \Gamma_q(x) < \phi^{**} M_q(u) + \phi^* \Gamma_q(u).$$

PROOF. (A continuous-time version of this proof is given in [Ger89c]). ∎

The next theorem is a moment inequality for L-mixing processes which resembles Burkholder's inequality.

Theorem 2.4.3 ([Ger89c]) *Let $(x_n), n \geq 0$ be an L-mixing process with $\mathbb{E}\,x_n = 0, \forall\ n \geq 0$, and let (f_n) be a deterministic sequence. Then for all $1 \leq m < \infty$*

$$\mathbb{E}^{1/2m} \sup_{1 \leq N' \leq N} \Big| \sum_{n=1}^{N'} f_n x_n \Big|^{2m} \leq C_m \Big(\sum_{n=1}^{N} f_n^2 \Big)^{1/2} M_{2m}^{1/2}(x) \Gamma_{2m}^{1/2}(x),$$

where C_m depends only on m.

Corollary 2.4.1 *Let (x_n) be as in Theorem 2.4.3, then*

$$\frac{1}{N}\sum_{n=1}^{N} x_n = O_M(N^{-1/2}).$$

Corollary 2.4.2 *Let (x_n) be as in Theorem 2.4.3, and let $0 < \lambda < 1$, then*

$$\sum_{n=1}^{N}(1-\lambda)^{N-n}\lambda x_n = O_M(\lambda^{1/2}).$$

Define the process $\Delta x/\Delta\theta \triangleq |x_n(\theta+h) - x_n(\theta)|/|h|$, where $\theta \neq \theta + h \in D$.

Definition 2.4.4 The stochastic process $x_n(\theta)$ is *M-Lipschitz-continuous in θ* if the process $\Delta x/\Delta\theta$ is M-bounded, i.e. if for all $1 \leq q < \infty$

$$M_q(\Delta x/\Delta\theta) = \sup_{\substack{n\geq 0 \\ \theta\neq\theta+h\in D}} \mathbb{E}^{1/q}|x_n(\theta+h) - x_n(\theta)|^q/|h| < \infty.$$

Example 2.4.5 If $(x_n(\theta))$ is absolutely continuous with respect to θ a.s. and the gradient of the process $(x_n(\theta))$ is M-bounded, then $(x_n(\theta))$ is M-Lipschitz-continuous.

Now let $(x_n(\theta))$ be a measurable, separable, M-bounded stochastic process, and also M-Lipschitz continuous in θ for $\theta \in D$. By Kolmogorov's continuity theorem, (c.f., [IH81]), the realizations of $(x_n(\theta))$ are continuous in θ with probability 1. Thus taking a compact domain $D_0 \subset \text{int}D$

$$x_n^* = \max_{\theta\in D_0}|x_n(\theta)|$$

is well-defined for almost all ω's. As the realizations of $x_n(\theta)$ are continuous, x_n^* is measurable with respect to $\mathcal{F}$, that is x_n^* is a random variable. Let us estimate its moments.

Theorem 2.4.4 *For all positive integers q and $s > q$*

$$M_q(x^*) \leq C(M_{qs}(x) + M_{qs}(\Delta x/\Delta\theta))$$

where C depends only on k, q, s and D_0, D.

The following theorem is a very useful result which, among others things, implies the validity of a uniform strong law of large numbers for L-mixing processes.

Combining Theorems 2.4.3 and 2.4.4 and setting $f_n = 1$, we get the following corrolary.

Corollary 2.4.3 *Let the assumptions of Theorems 2.4.3 and 2.4.4 hold, then*

$$\max_{\theta \in D_0} |\frac{1}{N} \sum_{n=1}^{N} x_n(\theta)| = O_M(N^{-1/2}),$$

and also for $0 < \lambda < 1$

$$\max_{\theta \in D_0} |\sum_{n=1}^{N} (1-\lambda)^{N-n} \lambda x_n(\theta)| = O_M(\lambda^{1/2}).$$

The following inequality will prove useful when estimating tail-probabilites of L-mixing process in Section 5.5.4.

Theorem 2.4.5 ([Ger89b]) *Let* (u_n), $n \geq 0$ *be a zero-mean, bounded L-mixing process such that* $\Gamma_\infty(u) < \infty$, *and let* (f_n), $n \geq 0$ *be a measurable, locally square summable function. Then*

$$\exp\left(\sum_{n=0}^{N} f_n u_n - 2M_\infty(u)\Gamma_\infty(u) \sum_{n=0}^{N} (f_n)^2\right) \leq 1.$$

If (x_n) is an L-mixing process and $(f(x))$ is an absolutely continuous function which grows at most polynomially together with its first derivatives, then one can show that the process $f(x_n)$ is also L-mixing. However, if we take a discontinuous function g, say $g(x) = \mathrm{I}_{x>c}$, I the characteristic function of the set $\{x > c\}$, then one cannot conclude that $(g(x_n))$ is L-mixing. Therefore, it is not true that the level set

$$A_{c,n} = \{\omega : x_n > c\}$$

can be approximated by $\mathcal{F}_m^+$-measurable sets. These observations are the basis for what are called L_0-mixing processes.

Definition 2.4.5 A stochastic process $(x_n(\theta))$, $n \geq 0$ is L_0-mixing uniformly in θ (with respect to $(\mathcal{F}_n, \mathcal{F}_n^+)$) if for any $q \geq 1$ and any $c > 0$

$$\Gamma_{q,c} = \sum_{\tau=1}^{\infty} \gamma_q^c(\tau) < \infty.$$

Obviously definition 2.4.5 also applies to processes which are not parameter dependent. We summarize four basic facts about L_0-mixing processes which can be found in [Ger92b]:

Theorem 2.4.6 *If a stochastic process (x_n) is L_0-mixing then for all $s \geq 1$*

$$\gamma_q(\tau, x) \leq 4^s \Gamma_{q,1/s}(x) n^{-s}. \tag{4.1}$$

Conversely, if for all $s \geq 1$, $\gamma_q(\tau, x) \leq C_s n^{-s}$, then (x_n) is L_0-mixing, and for any $c > 0$ and $s > 0$

$$\Gamma_{q,c}(x) \leq C_s^{1/c} s/(s-c).$$

Theorem 2.4.7 *Let (x_n), $n \geq 0$ be an L_0-mixing process, and let $I \subset \mathbb{R}$ be a fixed nonempty open interval. Then there exists a sequence of real numbers $\delta_n \in I$ such that the process $y_n = \mathrm{I}_{x>\delta_n}(x_n)$ is L_0-mixing, and for any $r \geq 1$ and $c > 0$*

$$\Gamma_{r,c}(y) \leq 2C_0 \Gamma_{r,c/(r+1+c)}^{(r+1+c)/(r+1)}(x)$$

where C_0 depends only on I and r.

Theorem 2.4.8 *Let $x = (x_n(\theta))$ be as in Theorem 5.2 and assume that $(x(\theta))$ and $\Delta x/\Delta\theta$ are L_0-mixing with respect to $(\mathcal{F}_n, \mathcal{F}_n^+)$. Then the process $x^* = (x_n^*)$ is also L_0-mixing with respect to $(\mathcal{F}_n, \mathcal{F}_n^+)$ and for any $c > 0$ and $r > p$*

$$\Gamma_{q,c}(x^*) \leq 2(\Gamma_{rq,c}(x)) + \Gamma_{rq,c}(\Delta x/\Delta\theta)).$$

Theorem 2.4.9 *Let (u_n) be an L_0-mixing process and define the process (x_n) by*

$$x_n = (1 - \lambda)x_{n-1} + \lambda u_n$$

with $0 < \lambda < 1$. Then (x_n) is L_0-mixing, and for all $q \geq 1$ and $c > 0$

$$\Gamma_{q,c}(x) = O(\lambda^{-1+c/2}).$$

Chapter 3

Prediction Error Method for ARMA Processes

The model selection, change–point detection, and adaptive control methods to be presented in this thesis rely heavily on the prediction-error method (PEM) for the identification of parameters in stochastic systems. This scheme has been extensively studied in such works as [And71], [LS84], [Lju87], [Cai88], and [HD88]. We shall present the prediction-error method for autoregressive moving-average ARMA processes, since this is the model on which the above problems will be formulated. Observe that the PEM method coincides with the conditional maximum likelihood method when the input is Gaussian white noise (c.f., [Cai88]). The conditions that will be imposed in this section to ARMA systems will be assumed to hold thereafter unless otherwise specified.

3.1 PEM: Off-Line Case

Let $(y_n), n = 0, \pm 1, \pm 2, \ldots$ be a second order stationary ARMA (p, q) process described by

$$y_n + a_1^* y_{n-1} + \ldots + a_{p^*}^* y_{n-p^*} = e_n + c_1^* e_{n-1} + \ldots + c_{q^*}^* e_{n-q^*}.$$

In a shorthand notation $A^* y = C^* e$, where A^*, C^* are polynomials of the backward shift operator, i.e. $A^*(z^{-1}) = \sum_{i=0}^{p^*} a_i^* z^{-i}$, and $C^*(z^{-1}) = \sum_{i=0}^{q^*}$, with $a_{p^*}^* \neq 0$ and $c_{q^*}^* \neq 0$, and $a_0^* = c_0^* = 1$.

Condition 3.1.1 *The polynomials $A^*(z^{-1})$ and $C^*(z^{-1})$ are stable and relative prime.*

To describe the noise process let us assume that we are given a probability space $(\Omega, \mathcal{F}, P)$ and a pair of families of σ-algebras $(\mathcal{F}_n, \mathcal{F}_n^+)$, $n \geq 0$ such that $\mathcal{F}_n \subset \mathcal{F}$ is increasing and $\mathcal{F}_n^+ \subset \mathcal{F}$ is decreasing. Moreover, $\mathcal{F}_n$ and $\mathcal{F}_n^+$ are independent for all n.

Condition 3.1.2 *The input noise (e_n) is a second order stationary, L-mixing process with respect to $(\mathcal{F}_n, \mathcal{F}_n^+)$, and furthermore*

$$\mathbb{E}(e_n | \mathcal{F}_{n-1}) = 0 \qquad \textit{and} \qquad \mathbb{E}((e_n)^2 | \mathcal{F}_{n-1}) = \sigma^2,$$

for all n. (The concept of L-mixing is described in Section 2.4).

Let θ denote the $k \triangleq p + q$-dimensional vector composed of the coefficients of the polynomials $A(z^{-1})$ and $C(z^{-1})$, and $D \subset \mathbb{R}^k$ be an open domain such that the polynomials $A(z^{-1})$ and $C(z^{-1})$ corresponding to $\theta \in D$ are stable. Moreover, let $D_0 \subset D$ be a compact domain with $\theta^* \in \text{int} D_0$, where $\text{int} D_0$ denotes the interior of D_0. For $\theta \in D$ define an estimated noise process $(\epsilon_n(\theta))$ by the difference equation

$$C\epsilon(\theta) = Ay,$$

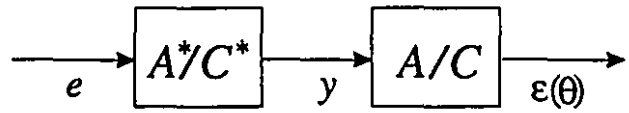


Figure 3.1: Generation of the prediction error process $(\epsilon_n(\theta))$.

where for $n \leq 0$ we set $y_n = \epsilon_n(\theta) = 0$.

Figure 3.1 illustrates the computation of the prediction error process $(\epsilon_n(\theta))$.

The cost function associated with the off-line time-invariant predicition-error method is given by

$$V_N(\theta) = \frac{1}{2}\sum_{n=1}^{N}(\epsilon_n(\theta))^2. \tag{3.1}$$

We shall define the off-line time-invariant estimator $\hat{\theta}_N$ as the parameter $\theta \in D$ that minimizes $V_N(\theta)$. It can be shown that this minimization is equivalent to solving

$$\frac{\partial}{\partial\theta}V_N(\theta) = 0, \tag{3.2}$$

or equivalently

$$\sum_{n=1}^{N}\epsilon_{\theta n}(\theta)\cdot\epsilon_n(\theta) = 0, \tag{3.3}$$

where differentiation is taken both in the almost sure and M-sense (c.f., Definition 2.4.1). (The differentiation notation was introduced on page 9.) More precisely, if (3.3) has a unique solution in D_0, then $\hat{\theta}_N$ is that solution. Otherwise, $\hat{\theta}_N$ is arbitrarily subjected to the condition that $\hat{\theta}_N \in D_0$. Note that the estimator $\hat{\theta}_N$ is measurable by the measurable selection theorem.

The asymptotic cost function of the off-line prediction-error method is given by

$$W(\theta) = \lim_{N\to\infty}\frac{1}{2}\mathbb{E}(\epsilon_N(\theta))^2.$$

It can be shown that

$$\lim_{N\to\infty}\sup_{\theta\in D_0}\left|\frac{1}{N}V_N(\theta) - W(\theta)\right| = 0 \qquad \text{a.s.} \tag{3.4}$$

and that the same uniform law of large numbers holds for the gradient processes $\partial V_N/\partial\theta$ and $\partial^2 V_N/\partial\theta^2$. (See e.g. [Lju76], [Han73]).

Under Conditions 3.1.1 and 3.1.2, the asymptotic equation $\partial W(\theta)/\partial\theta = 0$ has a unique solution in D_0 and the Hessian

$$R^* = \frac{\partial^2}{\partial\theta^2} W(\theta) \Big|_{\theta=\theta^*}$$

is non-singular (c.f., [ÅS74]). This fact and (3.4) imply that for almost all ω, the "likelihood equation" (3.1) has a unique solution in D_0 for $N > N_0(\omega)$ whenever $\theta^* \in \text{int}D_0$. A precise statement about the uniqueness of a solution is given by the following theorem:

Theorem 3.1.1 ([Ger89e]) *For each fixed $d > 0$ and any $m \geq 1$ the equation*

$$\frac{\partial}{\partial\theta} V_N(\theta) = 0,$$

has a unique solution in D_0. Moreover, this solution is also in the sphere $\{\theta : |\theta - \theta^*| < d\}$ with at least probability $1 - O(N^{-m})$.

A characterization of the estimation error of the off-line prediction-error identification method is provided by the next theorem:

Theorem 3.1.2 ([Ger89e]) *Under Conditions 3.1.2 and 3.1.1 we have*

$$\hat{\theta}_N - \theta^* = -(R^*)^{-1} \frac{1}{N} \sum_{n=1}^{N} \epsilon_{\theta n}(\hat{\theta}_{N-1}) e_n + O_M(N^{-1}).$$

An immediate consequence of Theorem 3.1.2 is the following result:

Theorem 3.1.3 ([Ger90b]) *Under the conditions of Theorem 3.1.2 we have*

$$\hat{\theta}_N - \theta^* = O_M(N^{-1/2}). \tag{3.5}$$

Note that the right hand side of (3.5) gives an almost sure upper bound for the $L_q(\Omega, \mathcal{F}, P)$ norm of the estimation error. For the latter, the law of the iterated logarithm applies (c.f., [ACH82]).

3.1.1 Off-Line PEM with Fixed Gain

For the time variant off-line estimation case, we use the prediction error algorithms with fixed gain given in [Ger89c], which "weighs down" past data with geometric rate. In this case the cost-function associated with this estimation method is given by

$$V_N^\lambda(\theta) = \sum_{n=1}^{N} (1-\lambda)^{N-n} \lambda (\epsilon_n(\theta))^2 \qquad 0 < \lambda < 1,$$

where λ is called the forgetting factor or gain of the algorithm. The off-line time variant prediction error estimator $\hat{\theta}_N^\lambda$ of θ_N^* is given as the solution of

$$\frac{\partial}{\partial \theta} V_N^\lambda(\theta) = V_{\theta N}^\lambda(\theta) = 0. \tag{3.6}$$

More precisely, if a unique solution of (3.6) exists in D, then $\hat{\theta}_N^\lambda$ is the D-valued random variable representing such a solution. Unfortunately, the probability of the "exceptional sets" of Ω, for which (3.6) has no solution, does not tend to 0 as $N \to \infty$. But this difficulty is dealt with in [Ger92b].

A characterization of the estimation error of the off-line small gain prediction-error identification method is given by the following theorem:

Theorem 3.1.4 ([Ger92b]) *Under Conditions 3.1.2 and 3.1.1 and (e_n) L_0-mixing we have*

$$\begin{aligned} \hat{\theta}_N^\lambda - \theta^* &= -(R^*)^{-1} \sum_{n=1}^{N} (1-\lambda)^{N-n} \lambda \epsilon_{\theta n}(\hat{\theta}_{N-1}^\lambda, \theta^*) e_n + r_N \\ &= O_M(\lambda^{1/2}), \end{aligned}$$

where $r_N = O_M(\lambda)$.

3.2 PEM: On-Line Case

Numerous engineering problems—found in areas such as robotics, power systems, aerospace—require solutions that ought to be implemented in real time. Since the dynamics of physical plants are frequently altered during operation due, for example, to changes in their operating conditions, wearing out of their mechanical parts, failures in their components, on-line identification methods are an essential component in the synthesis of automatic supervisory systems of plants.

In the early stages of development of the theory of recursive estimation, quite a few ad hoc methods, which are still very popular among practitioners in the field, were discovered. For instance, we could mention the extended least square, the instrumental variable, and the recursive maximum likelihood methods.

Among one of the best and most powerful is the recursive prediction-error method. We will present this method for the special case of the on-line estimation of the parameters of ARMA systems (c.f., [LS84], [Lju87], [Cai88], and [SS89]).

While recursive estimation of time invariant systems has attracted much attention, the recursive estimation of time varying systems has been almost completely neglected. However, a simple method for getting recursive estimators for the parameters of a time variant system has been known for some time. While simulation results show reasonable performance (e.g., [LS84] and Section 3.4), the lack of theoretical analysis has apparently discouraged many practitioners in the area from applying it. However, this drawback has been eliminated since many of the important theoretical aspects of the problem have been recently resolved. An off-line estimation method was developed and analyzed in [Ger89c], while a general time varying Ljung's scheme was presented and analyzed in [Ger89f].

The recursive time-invariant prediction error algorithm is summarized as follows.

Let us assume that an initial guess $\hat{\hat{\theta}}_0^0$ of the true parameter θ^* is given. Let $\hat{\hat{A}}_{N-1}^0$ and $\hat{\hat{C}}_{N-1}^0$ denote the polynomials corresponding to $\hat{\hat{\theta}}_{N-1}^0$. Assuming that processes $(\hat{\hat{\theta}}_n^0)$ and (ϵ_n^0) have been generated for $n \leq N-1$, we define ϵ_N^0 by the equation

$$\left(\hat{\hat{C}}_{N-1}^0 \epsilon^0\right)_N = \left(\hat{\hat{A}}_{N-1}^0 y\right)_N, \tag{3.7}$$

where the initial conditions are set to $y_n = \epsilon_n^0 = 0$ for $n \leq 0$. The left hand side of (3.7) means that the linear filter corresponding to the operator $\hat{\hat{C}}_{N-1}^0$ acts on the process (ϵ^0) with the evaluation being performed at time N. The right hand side of (3.7) is interpreted in a similar fashion.

It is easy to see that the gradient of ϵ_N^0 with respect to any $\theta \in D$ can be computed by

$$\left(\hat{\hat{C}}_{N-1}^0 \frac{\partial}{\partial \theta} \epsilon^0\right)_N = -\phi_{N-1}$$

where

$$\phi_{N-1} = (-y_{N-1}, \cdots - y_{N-p}, \epsilon_{N-1}^0, \cdots \epsilon_{N-q}^0)^T.$$

Now let $\hat{\hat{R}}_{N-1}^0$ denote the estimate of the Hessian $(\partial^2/\partial\theta^2)W(\theta)$, with initial guess $\hat{\hat{R}}_1^0 = cI, c > 0$. Then $\hat{\hat{\theta}}_N^0$ and $\hat{\hat{R}}_0^N$ are computed by the following recursion

$$\hat{\hat{\theta}}_{N-}^0 = \hat{\hat{\theta}}_{N-1}^0 - \frac{1}{N}\left(\hat{\hat{R}}_{N-1}^0\right)^{-1} \frac{\partial}{\partial\theta}\epsilon_N^0 \cdot \epsilon_N^0 \qquad N \geq 1 \tag{3.8}$$

$$\hat{\hat{R}}_{N-}^0 = \hat{\hat{R}}_{N-1}^0 + \frac{1}{N}\left(\left(\frac{\partial}{\partial\theta}\epsilon_N^0\right)\left(\frac{\partial}{\partial\theta}\epsilon_N^0\right)^T - \hat{\hat{R}}_{N-1}^0\right) \qquad N \geq 2. \tag{3.9}$$

The random variables $\hat{\hat{\theta}}_{N-}^0$ and $\hat{\hat{R}}_{N-}^0$ ought to be adjusted if they violate the boundedness conditions now described. Let $D_\theta \subset D$ and D_R be compact domains in $\mathbb{R}^{p+q}$ and $\mathbb{R}^{p\times p}$, respectively. Define $(\hat{\hat{\theta}}_N^0, \hat{\hat{R}}_N^0) = (\hat{\hat{\theta}}_{N-}^0, \hat{\hat{R}}_{N-}^0)$ if $(\hat{\hat{\theta}}_{N-}^0, \hat{\hat{R}}_{N-}^0) \in D_\theta \times D_R$ and $(\hat{\hat{\theta}}_N^0, \hat{\hat{R}}_N^0) = (\hat{\hat{\theta}}_0^0, \hat{\hat{R}}_0^0)$ if $(\hat{\hat{\theta}}_{N-}^0, \hat{\hat{R}}_{N-}^0) \notin D_\theta \times D_R$. Note that the time is not reset!

The domain $D_\theta \subset \mathbb{R}^{p+q}$ should be chosen in such a way as to guarantee the exponential stability of the time varying filter given by (3.8) and (3.9), which is

achieved by imposing Condition 3.2.1 below. Let the projection of D_θ on $\mathbb{R}^q$ be denoted by D_c. For each $c \in D_c$ there corresponds a polynomial $C(z^{-1})$, with which we can associate a companion matrix

$$\tilde{C} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & 0\dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ -c_1 & -c_2 & \dots & -c_q \end{bmatrix}$$

The set of these companion matrices will be denoted by $D_{\tilde{C}}$.

Condition 3.2.1 *The truncation domain D_θ is small enough in the sense that the matrices $\tilde{C}$ in $D_{\tilde{C}}$ are jointly stable, i.e. there exists a symmetric positive definite matrix U such that $\tilde{C}^T U \tilde{C} < b U$ with some $0 < b < 1$.*

REMARK. Although Condition 3.2.1 is certainly restrictive, it is inherent in Ljung's scheme. Indeed, we have to assume a-priori that the time-varying filter given by (3.8) and (3.9) is "slowly time-varying", and hence exponentially stable. Whether the local analysis of Ljung's scheme given in [Ger89f] can be "globalized" still remains an interesting question.

Clearly, we should assume that $\theta^* \in D_\theta$, but in any case this will be implied by Condition 3.2.2. As for D_R we assume that it is an arbitrary compact domain of symmetric positive definite matrices such that $R^* = (\partial^2/\partial\theta^2)W(\theta,\theta^*)|_{\theta=\theta^*} \in \text{int}D_R$. Other requirements on D_θ and D_R will be imposed by Condition 3.2.2.

To further specify the properties of D_θ and D_R, we consider the associated ordinary differential equation

$$\dot{\theta}(t) = -R(t)^{-1}\frac{\partial}{\partial\theta}W(\theta(t)) \tag{3.10}$$

$$\dot{R}(t) = G(\theta(t)) - R(t) \tag{3.11}$$

where

$$G(\theta) = \mathbb{E}\left(\frac{\partial}{\partial\theta}\epsilon(\theta)\right)^T\left(\frac{\partial}{\partial\theta}\epsilon(\theta)\right).$$

The right hand side of this ordinary differential equation is defined in $D \times \mathbb{R}^+(p \times p)$ where $\mathbb{R}^+(p \times p)$ denotes the set of symmetric positive definite $p \times p$ matrices. Now, since the Jacobian matrix associated to (3.10) and (3.11) linearized at point (θ^*, R^*) has the following structure

$$J = \begin{bmatrix} -I & 0 \\ X & -I \end{bmatrix},$$

the eigenvalues of the matrix J are -1. Thus (3.10) and (3.11) have a locally stationary point (θ^*, R^*) and moreover this equilibrium point is asymptotically stable (c.f., [ÅS74], [Cai88]).

It is essential for the analysis (c.f, [Ger89f]) that the solution trajectories of (3.10) and (3.11) starting from $(\hat{\bar{\theta}}_0^0, \hat{\bar{R}}_0^0)$ do not hit the boundary of $D_\theta \times D_R$. This can be ensured by the following condition which is a more explicit formulation of the usual assumption that the initial guess must be "good enough".

Condition 3.2.2 *$D_\theta \times D_R$ is a domain of attraction for (3.10) and (3.11) (i.e. for any initial value $(\theta(0), R(0)) \in D_\theta \times D_R$ the solution $(\theta(t), R(t))$ of (3.10) and (3.11) converges to (θ^*, R^*)). In addition, $(\theta^*, R^*) \in \text{int}D_{\theta,R}$, and $(\hat{\bar{\theta}}_0, \hat{\bar{R}}_0) \in \text{int}D_{\theta,R}$ where $D_{\theta,R}$ is a compact domain invariant for (3.10) and (3.11), and $D_{\theta,R} \subset D_\theta \times D_R$. Finally, the image of $D_{\theta,R}$ under the flow, say ϕ_t, defined by (3.10) and (3.11), is in $\text{int}D_{\theta,R}$ for any small $t > 0$.*

Condition 3.2.3 *For the "memory" of the input process (e_n) given by $\gamma_q(\tau, e)$ we have that for any $q \geq 1$ there exists $c > 0$ which may depend on q such that $\gamma_q(\tau, e) = O(\tau^{-1-c})$. In addition, we have for some $\delta > 0$*

$$\sup_{n\geq 0} \mathbb{E} \exp(\delta e_n^2) < \infty.$$

Theorem 3.2.1 ([Ger89f]) *Under Conditions 3.1.2-3.1.1, and Conditions 3.2.1-3.2.3 we have*

$$\hat{\bar{\theta}}_N^0 - \theta^* = O_M(N^{-1/2}) \quad \text{and} \quad \hat{\bar{R}}_N^0 - R^* = O_M(N^{-1/2}).$$

3.2.1 On-Line PEM with Fixed Gain

The recursive prediction error algorithm with fixed gain is summarized as follows. Let us assume that an initial guess $\hat{\bar{\theta}}_0^\lambda$ of the true parameter θ^*, is given. Assuming that processes $(\hat{\bar{\theta}}_n^\lambda)$ and (ϵ_n^λ) have been generated for $n \leq N-1$, we define ϵ_N^λ by the equation

$$\left(\hat{\bar{C}}_{N-1}^\lambda \epsilon^\lambda\right)_N = \left(\hat{\bar{A}}_{N-1}^\lambda y\right)_N, \tag{3.12}$$

where $\hat{\bar{A}}_{N-1}^\lambda$, and $\hat{\bar{C}}_{N-1}^\lambda$ denote the polynomials corresponding to $\hat{\bar{\theta}}_{N-1}^\lambda$. The initial conditions in (3.12) are set to $y_n = \epsilon_n^\lambda = 0$ for $n \leq 0$. It is easy to see that the gradient of ϵ_N^λ with respect to any $\theta \in D$ can be computed by

$$\left(\hat{\bar{C}}_{N-1}^\lambda \frac{\partial}{\partial \theta}\epsilon^\lambda\right)_N = -\phi_{N-1}$$

where

$$\phi_{N-1} = (-y_{N-1}, \cdots - y_{N-p}, \epsilon_{N-1}^\lambda, \cdots \epsilon_{N-q}^\lambda)^T.$$

Then $\hat{\bar{\theta}}_N^\lambda, \hat{\bar{R}}_N^\lambda$ are computed by the following recursion:

$$\hat{\bar{\theta}}_N^\lambda = \hat{\bar{\theta}}_{N-1}^\lambda - \left(\frac{1}{N} + \lambda\right)\left(\hat{\bar{R}}_{N-1}^\lambda\right)^{-1} \frac{\partial}{\partial \theta}\epsilon_N^\lambda \cdot \epsilon_N^\lambda \tag{3.13}$$

$$\hat{\bar{R}}_N^\lambda = \hat{\bar{R}}_{N-1}^\lambda + \left(\frac{1}{N} + \lambda\right)\left(\left(\frac{\partial}{\partial \theta}\epsilon_N^\lambda\right)\left(\frac{\partial}{\partial \theta}\epsilon_N^\lambda\right)^T - \hat{\bar{R}}_{N-1}^\lambda\right) \tag{3.14}$$

where $0 \leq \lambda \leq 1$ is the fixed gain of the algorithm.

The recursive prediction error algorithm was written in a form which includes both its time invariant and time variant versions. For if $\lambda = 0$, the popular time invariant prediction error algorithm version is obtained, whereas if $\lambda > 0$, the algorithm has the

capabilities of tracking time variant parameters. The role of λ, when $\lambda > 0$, is similar to that of the off-line counterpart, that is, to "weigh down" past data with geometric rate. In (11) and (12) the choice of $1/N + \lambda$, as the fixed gain for the algorithm, is chosen so as to reduce the uncertainty due to initial conditions at the start of the recursive algorithm, and to track the time varying parameters afterwards.

3.3 On-Line and Off-Line PEM's Link

A heuristic derivation of the RML method was obtained by considering an approximate recursion of the solution of the likelihood equation. This all but forgotten derivation nonetheless indicates that there should be an intimate relation between the nonrecursive or off-line and the recursive or on-line maximum-likelihood estimator.

Theorem 3.3.1 ([Ger91d], [Ger90b]) *Under Conditions 3.1.1–3.2.3 we have*

$$\widehat{\theta}_N - \widehat{\widehat{\theta}}_N^0 = O_M(\log N/N).$$

3.4 A Simulation of the Recursive PEM

Here, we shall illustrate the recursive prediction-error methods introduced in the previous sections. A process (y) will be generated by appending in time a time invariant and a time variant ARMA system. This is done so as to allow us to illustrate: i) how the time invariant version of the PEM will give consistent parameter estimates when the parameters of the system are time invariant; ii) how the time variant version of the PEM will provide tracking capabilities of the time variant parameters.

Let the time invariant ARMA(2,1) system then be given by

$$y_N + a_1^* y_{N-1} + a_2^* y_{N-2} = e_{N-1} e_N + c_1^* e_{N-1}, \tag{3.15}$$

with

$$a_1^* = -.7 \qquad a_2^* = .8 \qquad c_1^* = -.4. \tag{3.16}$$

Then (3.15) generates the process (y) for $N < 500$.

Now let the slowly time varying ARMA(2,1) system be given by

$$y_N + a_{N,1}^* y_{N-1} + a_{N,2}^* y_{N-2} = e_N + c_{N,1}^* e_{N-1}, \tag{3.17}$$

where the time variant parameters $a_{N,1}^*, a_{N,2}^*$ and $c_{N,1}^*$ are obtained by linearly moving from the time invariant parameters in (3.16) to the parameters

$$a_{N_f,1}^* = -.7 \qquad a_{N_f,2}^* = .2 \qquad c_{N_f,1}^* = -.7. \tag{3.18}$$

The process (y) is finally generated by (3.17) for $N = 500, \ldots, 1000$. (A rigorous description of this type of time-variant system is given in Section 5.5.) Both ARMA processes are driven by a Gaussian white noise process (e) with mean 0 and variance 1. We now run two recursive PEM's in parallel. The time-invariant prediction error method provides the estimates $\hat{\hat{a}}_{1,N}^0$, $\hat{\hat{a}}_{2,N}^0$, and $\hat{\hat{c}}_{1,N}^0$ of a_1^*, a_2^*, and c_1^* respectively; whereas the time-invariant prediction error method with fixed gain $\lambda = .0113$ gives the estimates $\hat{\hat{a}}_{1,N}^\lambda$, $\hat{\hat{a}}_{2,N}^\lambda$, and $\hat{\hat{c}}_{1,N}^\lambda$ of $a_{1,N}^*$, $a_{2,N}^*$, and $c_{1,N}^*$, respectively. Both the parameter estimates and the "true" parameters are plotted in Figures 3.2–3.4.

Next, is a similar simulation to the one just introduced with the only difference that we now take the fixed gain $\lambda = .02$. Moreover, instead of having a slowly time variant change in the dynamics we have an abrupt jump at $N = 500$, with the dynamics remaining constant after the jump. The initial values of the parameters of the polynomials of the ARMA system before $N = 500$ are given by (3.16) whereas the values of the parameters after $N = 500$ are given by (3.18). Parameter estimates and "true" parameters are now plotted in Figures 3.5–3.7.

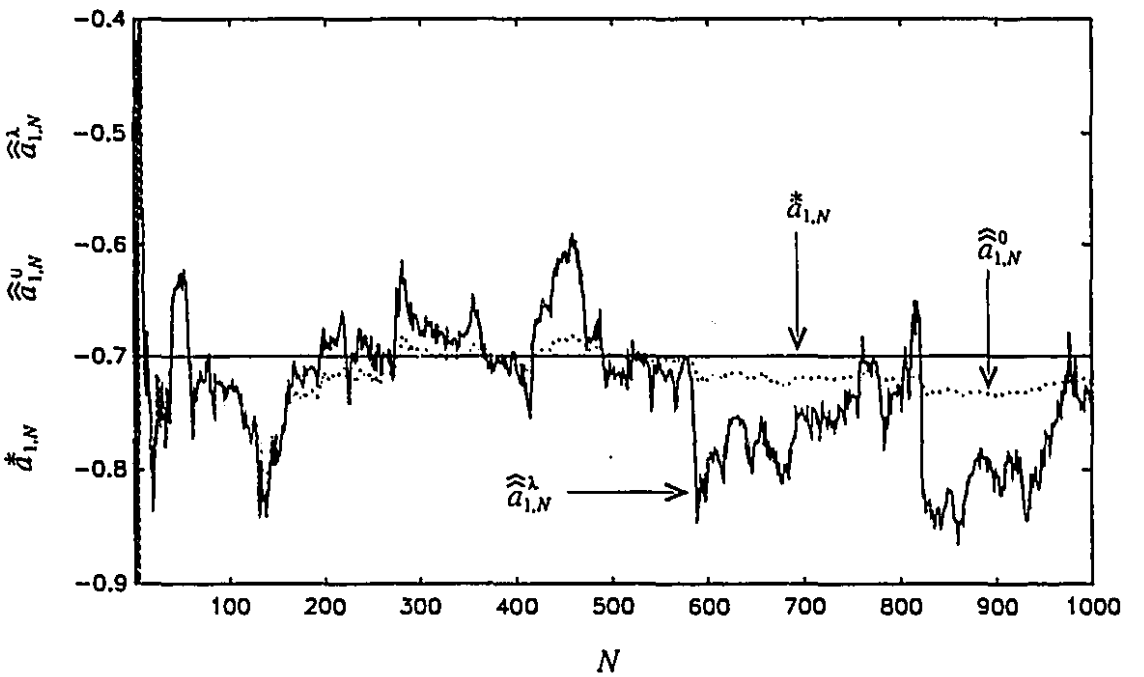


Figure 3.2: The true parameter $a^{*}_{1,N}$, and its time invariant and time variant estimates $\hat{\hat{a}}^{0}_{1,N}$, and $\hat{\hat{a}}^{\lambda}_{1,N}$ respectively.

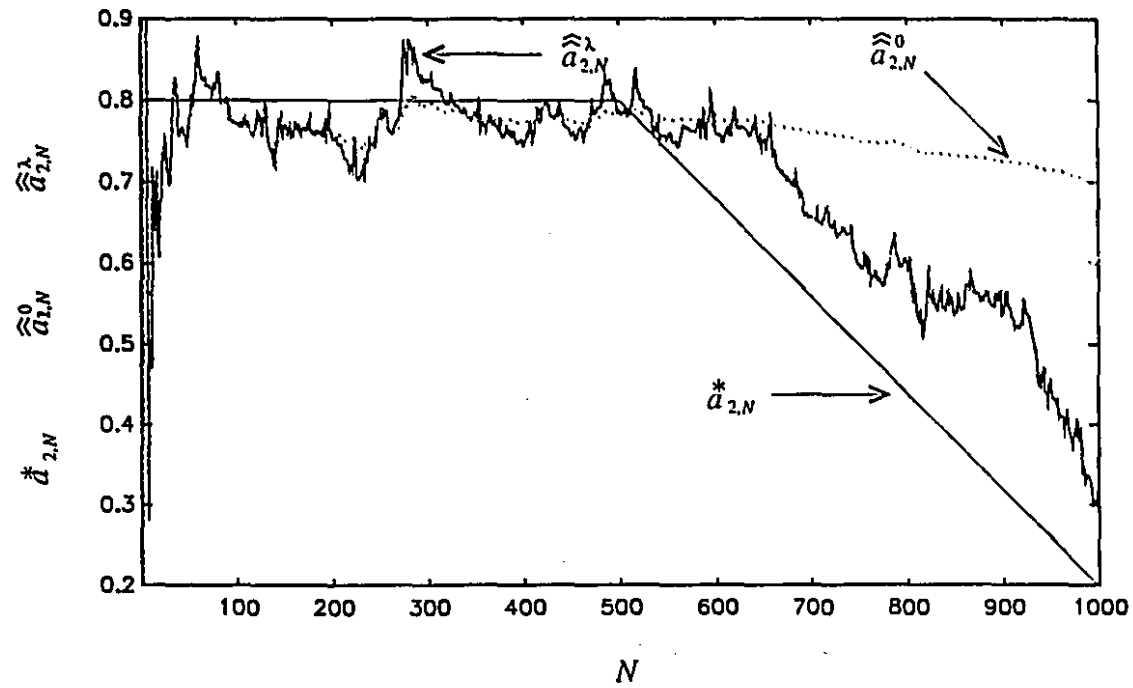


Figure 3.3: The true parameter $a^{*}_{2,N}$, and its time invariant and time variant estimates $\hat{\hat{a}}^{0}_{2,N}$, and $\hat{\hat{a}}^{\lambda}_{2,N}$ respectively.

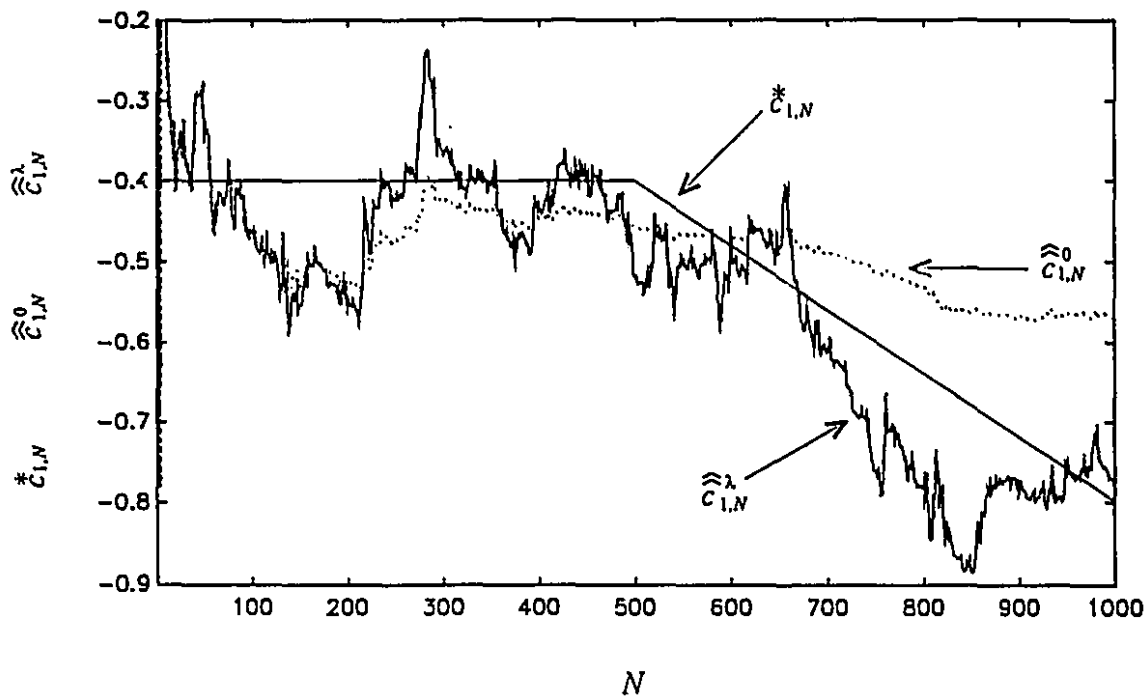


Figure 3.4: The true parameter $c^{*}_{1,N}$, and its time invariant and time variant estimates $\hat{\hat{c}}^{0}_{1,N}$, and $\hat{\hat{c}}^{\lambda}_{1,N}$ respectively.

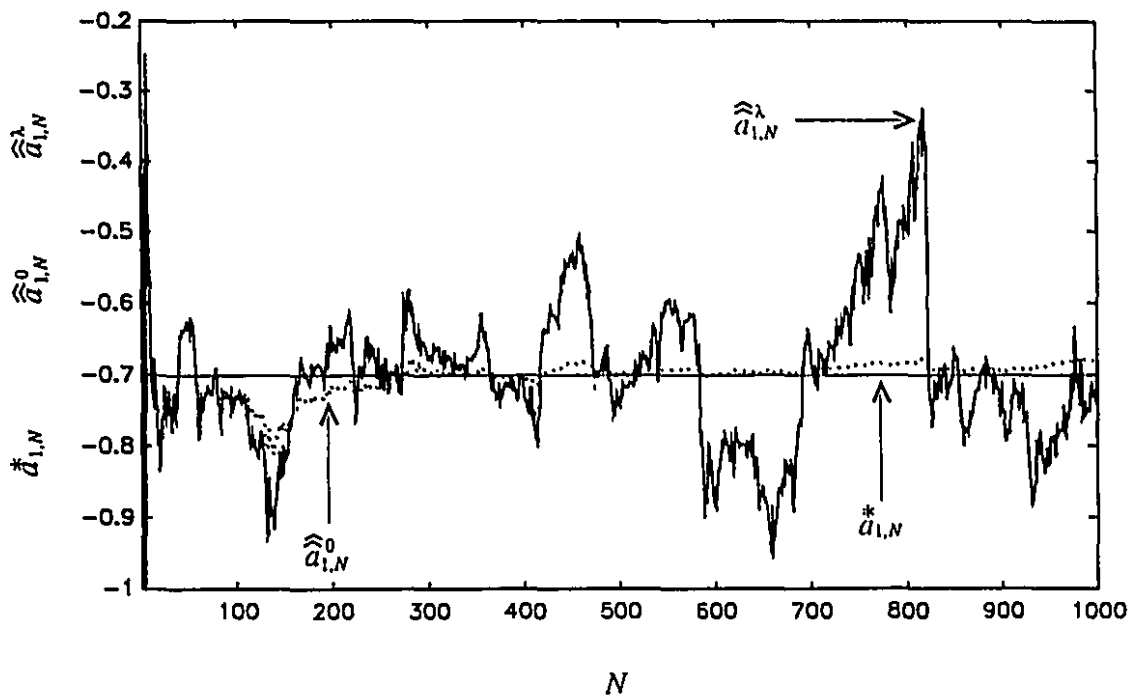


Figure 3.5: **The true parameter $a^{*}_{1,N}$, and its time invariant and time variant estimates $\hat{\hat{a}}^{0}_{1,N}$, and $\hat{\hat{a}}^{\lambda}_{1,N}$ respectively.**

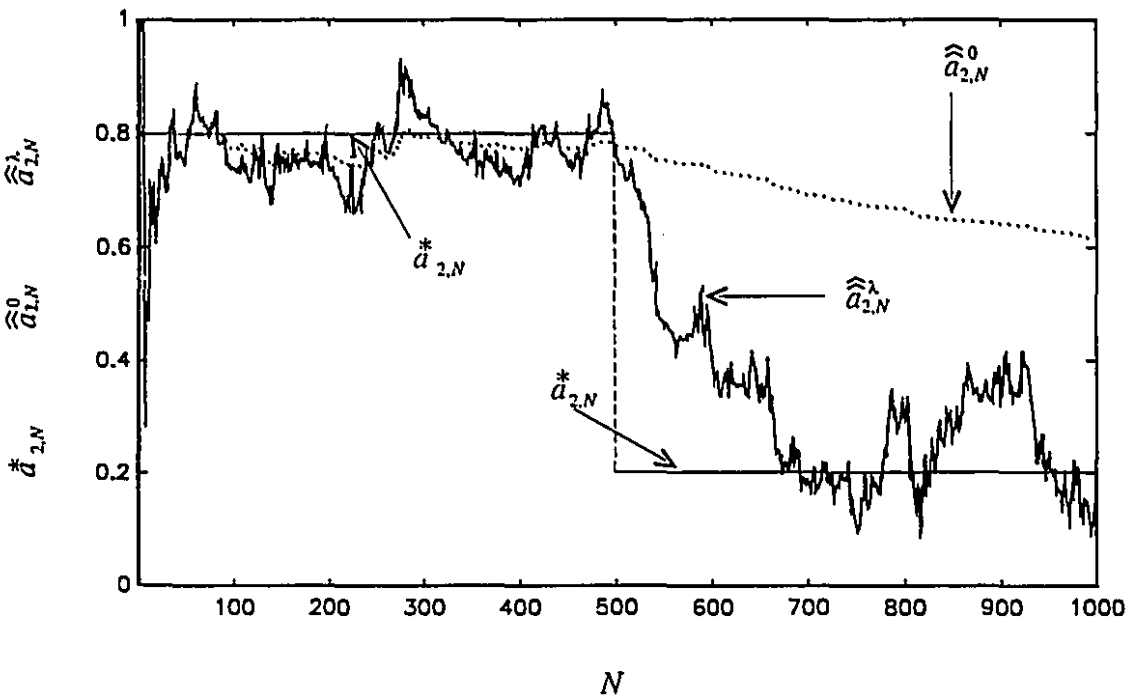


Figure 3.6: The true parameter $a^{*}_{2,N}$, and its time invariant and time variant estimates $\hat{\hat{a}}^{0}_{2,N}$, and $\hat{\hat{a}}^{\lambda}_{2,N}$ respectively.

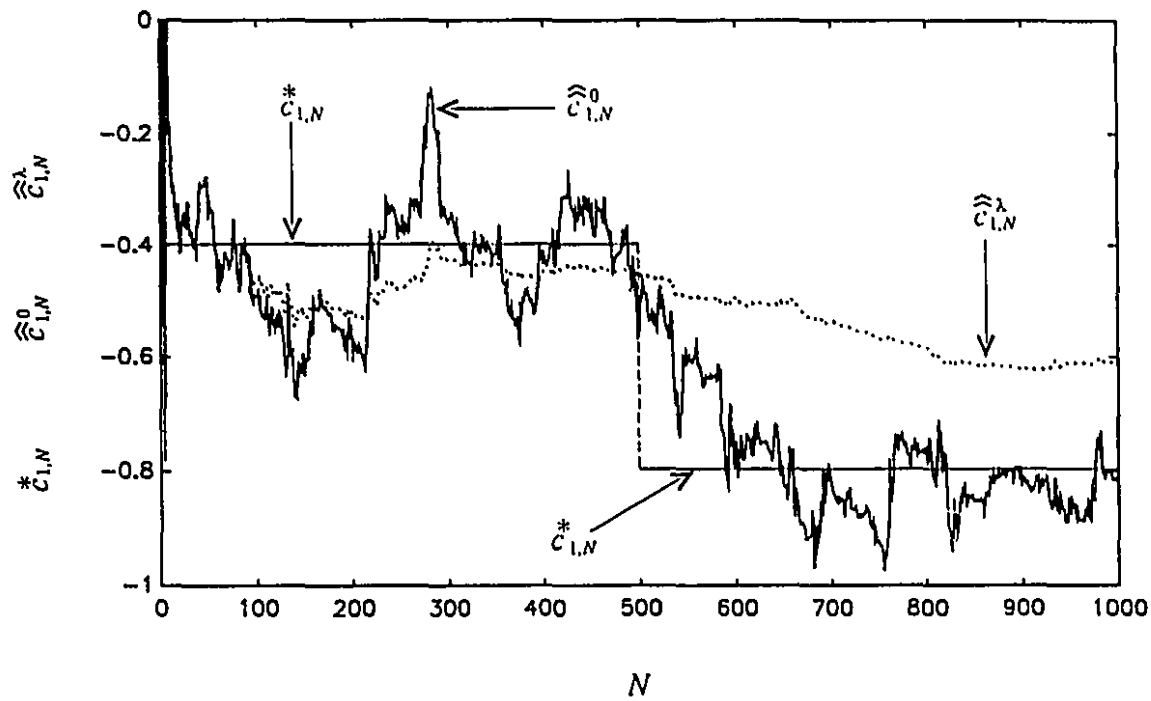


Figure 3.7: The true parameter $c^{*}_{1,N}$, and its time invariant and time variant estimates $\hat{\hat{c}}^{0}_{1,N}$, and $\hat{\hat{c}}^{\lambda}_{1,N}$ respectively.

Chapter 4

Model Order Selection

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality. Albert Einstein

4.1 Introduction

A major constituent of scientific research is the construction of mathematical models for elements of our perceivable world, which use as raw material, data provided by our ever more sophisticated instruments. Mathematical models (thereafter simply called models) act as abstract and compact languages to assist us in the endless search for a better understanding of reality. Hence, the primary aim of such intellectual produce is to attempt to unearth what could be referred to allegorically as the mechanism that generated a particular set of data. By extracting the regular features of data in the form of models, the complexity of observable events would tend to be reduced. Any such found regularities could be interpreted as reflections of nature's relative order.

Can models exactly match the particular reality they intend to explain, and thus in some sense allow for a mirroring of the total "true" nature of physical phenomena?

The first known affirmative answer dates back to Pythagoras who believed that nature itself was numbers. Many centuries later, Galileo Galilei sustained that the great book of nature was written in terms of mathematical characters, like triangles, circles, and other geometric figures. Numerous other great thinkers like Descartes, Leibniz, and Kant also held the viewpoint of mathematical models as being ingrained in reality. This perspective, although always somehow confronted, began to be strongly challenged at the beginning of this century, a period in which firmly established theories, like Newton's classical mechanics, began to be improved upon or even abandoned. As a result of this deeper look at reality through clever and elaborated experiments in mainly the subatomic world, absolute certainty gave way to the notion of the perpetual inherent imperfection of any theory or, in particular, any model of reality. Although this conviction is constantly gaining further ground a *definite* answer to such a profound philosophical question cannot be provided with absolute certainty.

Statistics foremost mission lays in the construction of probability models for data. The prevalent tendency in this field has been to assume that a given data set is the outcome of a "true", albeit almost never fully specified, distribution. Thus, that the data is actually considered as being generated by some sort of mechanism governed by such a distribution. (One could assume that the distribution are of Normal, Binomial, Exponential or Poisson type for example.) We would then expect, actually require, the field of statistics to offer well established methods by which to identify from the data that "true" type distribution in some categorical and explicit way. Unfortunately this is not the case: no general applied rigorous data-dependent methodology is available by which to obtain that sort of "true" type distribution. Moreover, physical justifications for the existence of "true" models can, in a way, be proven only applicable in a handful of cases. (One could mention, for example, gambling games.) Therefore we could rightfully argue that the very initial steps of

the statistical modeling process are in general dealt with in an ad-hoc manner.

After a "true" type model has been assumed by any kind of dubious method, the statistical discipline gives rigorous and involved techniques by which to determine whatever has been left unspecified in those a-priori assumptions. Examples of such techniques abound in the highly developed theory of estimation which incorporates elaborate procedures to allow for the making of decisions about the modeled phenomenon in "optimal" ways. However, a note of caution must be raised since all these a-posteriori claims are contingent on the correctness of the a-priori assumptions, assumptions that always contain some degree of subjectivity. As a result, we could face the risk of making unfounded statements about the studied phenomenon, while unsoundly making rigorous only the last phases of the modeling process.

Is it possible to choose a model in a wholly data-dependent manner? Tackling this question requires a proper understanding of the relationship and tradeoff between model complexity and model fit. For instance, a complex model might be capable of matching data with high precision. However, if the description of the model itself (in some well specified sense) turns out to be as lengthy as that of the data, then no overall reduction in the complexity of the original string of data would be achieved. A simple example is the polynomial fitting of data. Assume that the dimension of the polynomial is taken to be equal to the number of data points. Then we are guaranteed to have a perfect fit, but with a model complexity matching or superseding that of the raw data. If this type of procedure is employed, it will certainly defeat the modeling objective of striving to maximally reduce the complexity of events by extracting their regular features using compact mathematical descriptions. Hence a necessary tradeoff between model complexity and model fit needs to be taken into account.

Even though the statistical discipline has notably enlarged our understanding of numerous aspects of our perceivable reality, the bulk of this theory fails to answer such preliminary questions as how to choose between models of different complexity.

For example, the classical least square method (c.f., [Cai88]) succeeds in providing the parameter values of a polynomial of fixed order in some optimal way, but breaks down when the polynomial order needs to be determined. Thus, most of the statistical modeling methods either totally fail to consider the above complexity-fit tradeoff, or they address it in a form not disassociated from what could be after-all subjective considerations.

Let us look at the Minimum Prediction Error (MPE) modeling scheme, which can encompass a wide variety of modeling selection theories. This method is defined in terms of two scalar criterion functions, one which penalizes the lack of fit of the model with respect to the data, while the other penalizes the complexity of the model. (For a general and rigorous treatment of the MPE method the reader is referred to [Cai88].) For instance take a family of functions $\{h(\cdot,\theta), \theta \in D\}$ (one could think, for example, of the family of polynomials) and consider the predictor sequence $\hat{y}_1, \ldots, \hat{y}_N$ for a set of data $y_1, \ldots, y_N$ given by the predictor models

$$\hat{y}_n = h(x_n, \theta) \qquad 1 \leq n \leq N,$$

with $x_1, \ldots, x_N$ some known deterministic sequence. An MPE criterion can then be constructed with the help of the loss functions: $l(\cdot,\cdot)$, which penalizes the lack of fit of the model with respect to the data, and $\kappa(\cdot)$, which weighs the complexity of the model itself. Finally one defines the criterion function as

$$L_N(\hat{y}) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \kappa(h), \tag{4.1}$$

Although $L_N(\hat{y})$ directly addresses the issue of model complexity versus model fit, it does not provide in its general formulation any clue on how to properly balance these two issues. Even if a subjectively chosen criterion based on (4.1) is found to satisfy properties like consistency and efficiency, it will hardly achieve what is of outmost importance, that of a solid physical interpretation for such a criterion.

Since any successful modeling method will in some form weigh the complexity-fit tradeoff, and since the MPE is defined in such a generality so as to encompass all reasonable weigh types, one can certainly claim that all modeling methods are just special cases of the MPE method. However, this type of reasoning is in a sense deceptive since the MPE method by itself fails to guide us towards concrete and meaningful modeling theories, theories that should be entrenched with solid physical interpretations outside of their internal logical consistencies.

It is essential to stress the basic recurring theme in all of the statistical modeling methods, that is, the assumption of the existence of a "true" distribution for the data. Under this category we can also include those methods addressing the issue of model complexity, and even those which use an approximating family of models not containing the "true" distribution. It is then not surprising that justifications for these procedures are based on how well they succeed in providing an estimate, i.e. an approximation, of the "true" model. The bottom line is that such a "true" model is made accountable for all interpretations, predictions and decisions. Moreover, this sort of rationale lacks any real base, since subjective features are necessarily associated with the assumption of a "true" model. As a result we could say that the classical inference methods might all have the appearance of being objective techniques but that in fact they hide the real issue behind the cloud of "true" models. These methods therefore fail to be wholly data-dependent, which represents a serious drawback since it could be argued that in very general terms the only available "truth" is the given data.

Let us now briefly show how the most renowned modeling procedures, derived from the Kullback-Leibler distance between two probability densities, are developed. We shall use as an example the well-known AIC criteria (c.f., [Aka73]).

Assume that the data y^N is a realization of an i.i.d. sequence of random variables

with density function $g(\cdot)$. Let the family of density functions $\mathcal{M}(\theta) = \{f(\cdot,\theta); \theta \in D\}$ be used to fit a model to the data $\mathbf{y}^N$. The "true" density $g(\cdot)$ is not necessarily included in the model class $\mathcal{M}(\theta)$. Now recall that the Kullback-Leibler distance between two densities is given by

$$M(f,g) = -\int_{-\infty}^{\infty} \log f(\cdot,\theta) g(x) dx + \int_{-\infty}^{\infty} \log g(x) g(x) dx. \tag{4.2}$$

Since the second term in (4.2) is constant,

$$L(\theta) = -\mathbb{E} \log f(\cdot,\theta)$$

can be used as a basis for model selection. Note, however, that $L(\theta)$ cannot be used in its present form to solve the selection problem, since it requires the computation of an expectation. Moreover, the expectation is with respect to the unknown density $g(\cdot)$. Therefore it is necessary to find some computable approximation of $L(\theta)$. The construction of practical modeling methods derived from cost functions like (4.2) can then be carried out as follows. (Since here the purpose is only to introduce the main ideas of how these types of modeling methods can be developed, we refer the reader to [Aka73] for the list of technical conditions needed to apply this type of methodology.)

(i) Let

$$\mathring{\theta} = \arg\min_{\theta \in D} L(\theta)$$

and

$$\hat{\theta}_N = \arg\min_{\theta \in D} \hat{L}_N(\theta).$$

where

$$\hat{L}_N = -\frac{1}{N} \sum_{i=1}^{N} \log f(y_i,\theta)$$

That is, $\hat{\theta}_N$ corresponds to the maximum likelihood estimator of $\mathring{\theta}$. Proceed by performing a Taylor series expansion of $\hat{L}_N$ about $\mathring{\theta}$ to get the crucial approximation result

$$L(\hat{\theta}_N) \approx L(\mathring{\theta}) + \frac{1}{2N}\mathrm{tr}\Gamma^{-1}\Sigma, \tag{4.3}$$

where matrix Σ is given by

$$\Sigma = \frac{\partial}{\partial\theta}L(\theta)\ \Big|_{\theta=\mathring{\theta}},$$

and the matrix Γ by

$$\Gamma = \frac{\partial^2}{\partial\theta^2}L(\theta)\ \Big|_{\theta=\mathring{\theta}}.$$

(ii) Compute unbiased estimators for the terms of the right hand side of (4.3). For $L(\mathring{\theta})$ one gets

$$\hat{L}_N = \frac{1}{N}\sum_{i=1}^{N}\log f(y_i, \hat{\theta}_N)$$

whereas for Σ and Γ one obtains

$$\hat{\Sigma}_N = \frac{\partial}{\partial\theta}\hat{L}_N(\theta)\ \Big|_{\theta=\mathring{\theta}}$$

and

$$\hat{\Gamma}_N = \frac{\partial^2}{\partial\theta^2}\hat{L}_N(\theta)\ \Big|_{\theta=\mathring{\theta}},$$

respectively.

The final criterion subsequently reduces to

$$L_N = -\frac{1}{N}\sum_{i=1}^{N}\log f(y_i, \hat{\theta}_N) + \frac{1}{2N}\mathrm{tr}\Gamma_n^{-1}\Sigma_n \tag{4.4}$$

It is interesting to note that if there exists a density $f(\cdot, \theta)$ which can completely match the "true" density $g(\cdot)$ then one can show that $\Gamma = \Sigma$. Thus $\mathrm{tr}\Gamma_n^{-1}\Sigma_n = \dim\theta$,

and the criterion function (4.4) simplifies to the well known Akaike's criterion (AIC) given by

$$AIC(k) = \sum_{i=1}^{N} -\log f(y_i, \widehat{\theta}_N) + k.$$

Note that the assumption of the existence of a "true" distribution is used in an essential way to arrive at a suitable utility function. This is because the above method is derived as an approximation of the minimum Kullback-Leibler distance between the "true" density $g(\cdot)$ and the densities in $M(\theta)$.

Many of the well-known modeling procedures are derived in a fashion similar to the above. They moreover constitute what are generally agreed to be the most rigorous techniques available for model selection. One could mention for example the BIC (c.f., [Sch78] and [Saw78]), the Pearson Chi-squared, and the Cramér-von Mises criteria among others (c.f., [LZ86]). The first is based on an asymptotically unbiased estimator for the Kullback-Leibler distance, whereas the other two use as a starting point, different notions for the "distance" between models. For instance the Cramér-von Mises distance is defined as

$$L(\theta) = \mathbb{E}\,(g(\cdot) - f(\cdot, \theta))^2,$$

and the steps which lead to its final criterion are done in a similar manner to that which leads to the AIC criterion.

What could be raised as another salient inadequacy in all those modeling methods, aside from their use of "true" models, is the absence of a solid physical interpretation of their final criteria. As a consequence, the methods lack the intuitive appeal so necessary when dealing with practical problems. Furthermore, and in particular for the very popular AIC criterion, it generally fails to be consistent (c.f., [Ris89]).

A well-known modeling procedure which is wholly data-dependent is the cross-validation method (c.f., [Sto74]). Its basic idea is the following: fit a model to a data set of size $N - m$, and then perform a validation test. For example, use the linear

regression model given by (2.6) to model the data y^N, and define the prediction error process

$$\epsilon_n(\theta) = y_n - x_n^T\theta.$$

For the sake of simplicity let $m = 1$ (usually refered to as the "one item out cross-validation criterion"). Now compute the following sequence of least square estimators

$$\hat{\theta}(j) = \arg\min_{\theta \in D} \left\{ \sum_{i=1, i\neq j}^{N} (\epsilon_n(\theta))^2 \right\}.$$

Then the criterion is based on the cumulative sum of the squares of the off-sample prediction errors

$$L_N(k) = \frac{1}{N}\sum_{n=1}^{N}(\epsilon_n(\hat{\theta}(n)))^2.$$

The degree of the polynomial is then taken as that value of k which corresponds to the minimum value of the criterion $L_N(k)$. The main objections to the use of this modeling method are as follows. Firstly, it is a heuristic approach with no rigorous indication of how to choose the crucial parameter m. Secondly, it was shown in [Sto77] that the cross-validation method is asymptotically equivalent to the AIC method, and thus the criticisms of the latter also hold for the former.

A possible way out of the modeling problem dilemma is to elaborate a modeling theory which will a-priori grant some degree of imperfectness to any model. What will then become meaningful is the ability to compare tentative model classes of different complexity in the light of available data, and give definite answers about the goodness-of-fit of a model class only in relation to tentative competing models.

In the mid-seventies a novel modeling approach was initiated by Jorma Rissanen [Ris78] which could in a sense be considered a natural outgrowth of the theory of algorithmic complexity developed among others by Chaitin, Solomonoff, Kolmogorov (c.f., [KA87], [Lev73], [ML74], and [ZL70]). This modeling endeavor, presently known as stochastic complexity, has constituted a major breakthrough in the way we view

modeling and moreover it has had profound practical repercussions. It has become by now one of the most respected methods for statistical inference. (For some early developments see [Ris78], [Sch78], and [Shi80]. Some recent surveys are given in [Ris87], [Ris89], and [GR91].) As a radically different conceptual formulation with respect to previous modeling methods, the stochastic complexity is characterized by the following distinguishable features:

(i) Not based on "true" model assumptions.

(ii) Subjective only in the selection of tentative but not necessarily forever definite model classes.

(iii) Unique-type criterion function is defined for all modeling problems.

(iv) The criterion function weighs model complexity and model fit in a natural way.

(v) Universal in that any model classes—parametric or non-parametric—can be compared irrespective of their distinct complexity through a common benchmark: the associated total code length.

(vi) Computable, unlike the theory of algorithm complexity.

Let us stress that aside from the suggestive choice of model classes, the stochastic complexity modeling method is not subjected to arbitrary or subjective choices. Moreover, it is totally data-dependent and provides us with a clear and concrete physical justification outside of the internal consistency of the theory.

In the following two sections we shall present the stochastic complexity theory in more detail.

4.2 Stochastic Complexity

The art of modeling has a natural connection to the theory of information and coding (refer to Section 2.3 for a short introduction of this theory). The link is established by the fact that finding data constraints (if one's goal is to send the same amount of information as that contained in the original data set) amounts to reducing the number of bits to be transmitted. Conversely, obtaining shorter descriptions for the data gives rise to models which better represent the constraints of raw data.

As a simple example consider the case of a data y^N such that $\mathrm{y}_n \geq 0; \forall n \in [1, N]$. Then, only the binary digit

$$b = \begin{cases} 1, & \mathrm{y}_n \geq 0 \ \forall n \in [1, N]; \\ 0, & \exists\, n \in [1, N] \text{ such that } \mathrm{y}_n < 0. \end{cases}$$

would be needed to represent a possible inherent constraint of the so-called generating mechanism, instead of using a binary digit per datum as the representation. The search for the shortest encoding of a set of data can be taken to be equivalent to the search for best models, an issue which will be further elaborated. (Encoding of a set of data means a more compact data representation with respect to the original data set, but that maintains the same amount of information as that which is contained in the raw data.)

Under the assumption that a set of data y^N behaves according to a "true" probability model, say $f(\cdot, \dot{\theta})$ with $\dot{\theta} \in D$, Shannon proved that the shortest encoding in the mean per symbol codelength sense can be asymptotically attained by the entropy, that is $-\mathbb{E} \log f(\cdot, \dot{\theta})$ (see Section 2.3). If a model deviates somehow from the "true" distribution $f(\cdot, \dot{\theta})$ then the Kullback-Leibler inequality (c.f., Proposition 2.3.4) can be used, in principle, as a basis for measuring the resulting degradation in coding performance. For instance let $\widehat{\theta} \in D$ be an estimator of the "true" parameter $\dot{\theta}$ then

$$\mathbb{E} \log f(\cdot, \widehat{\theta}) - \mathbb{E} \log f(\cdot, \dot{\theta}) \geq 0, \tag{4.5}$$

which vanishes if and only if $\hat{\theta} = \mathring{\theta}$ under proper identifiability conditions (e.g., $\mathbb{E} \log f(\cdot, \theta_1) - \mathbb{E} \log f(\cdot, \theta_2) = 0$ implies $\theta_1 = \theta_2$ for all $\theta_1, \theta_2 \in D$).

Observe that the left hand side of (4.5) corresponds for fix k to the log-likelihood equation which in Section 2.2 was shown to be an unsound model selection criterion since it failed to penalize the complexity of models. Therefore, if we are to base a theory of modeling upon the theory of information and coding, some essential modifications must be performed. To begin with we should remove the assumption of a "true" model. Then we should associate to any tentative model class a certain degree of uncertainty by properly weighing its complexity. However, this will require a redefinition of the notion of Shannon's information, more along the lines of algorithmic complexity, which shall be discussed shortly.

Before doing so let us introduce Theorem 4.2.1 which gives a generalization of the Kullback-Leibler inequality in that it will remain valid when models of different complexities are involved. This theorem will prove essential as a theoretical foundation of the stochastic complexity theory.

First, the proof of Theorem 4.2.1 requires the following technical condition for the smoothness of the tails of parametric densities.

Definition 4.2.1 Let $\{f(y^N, \theta); \theta \in D^k\}$, where D^k is a compact subset of $\mathbb{R}^k$ with nonempty interior, be a set of compatible probability density functions (i.e., $\sum_{y_{N+1}} P(y^N, y_{N+1}; \theta) = P(y^N; \theta)$) Then the *tail-condition* is satisfied if there exist estimators $\hat{\theta}_N = \hat{\theta}_N(y^N)$ such that for any real constant $c > 0$

$$\sum_{N=1}^{\infty} P(N^{1/2}|\hat{\theta}_N - \theta| > c \log N) < \infty$$

uniformly in θ.

The tail-condition imposes a constraint on the convergence of the parameter estimates $\hat{\theta}_N$'s. It has by now been verified in many important case like for in the AR

case in [HMP89] and [HD89], and for Gaussian ARMA processes in [Ger89c]. (The extension of this last result to multivariable linear stochastic systems is straightforward.)

Theorem 4.2.1 (c.f. [Ris86]) *Assume that the tail-condition is satisfied, and let $g(\cdot)$ be any set of compatible probability density functions defined on the data y^N. Then, for all $\theta \in D^k$,*

$$\varliminf_{N\to\infty} \; \mathbb{E}_\theta \left(\log f_N(y^N, \theta) - \log g_N(y^N) \right) / \log N \geq \frac{k}{2} \tag{4.6}$$

except for a subset of D^k with Lebesgue measure zero.

REMARK. Theorem 4.2.1 provides an asymptotically sharp lower bound for the code-length no matter what encoding procedure is used.

By comparing (4.6) with (4.5) we see that Theorem 4.2.1 does indeed generalize the Kullback-Leibler inequality. According to this theorem, no matter of how well we manage to provide an estimate $g_N(y^N)$ to approximate an assumed true density $f_N(y^N, \theta)$, the generalized Kullback-Leibler distance between these two probability densities cannot be made asymptotically smaller than the lower bound $(k/2)\log N$. This result follows the intuitive notion that complex models should be intrinsically more difficult to estimate due to the increase of a-priori model uncertainty. In principle, Theorem 4.2.1 reveals something about the merit of the estimation algorithm when its performance is compared to the lower bound $(k/2)\log N$. We shall later show that this theorem can be taken as a theoretical foundation of the stochastic complexity theory.

Can the information content of a data set be computed directly from the data itself without turning to typically poorly justified a-priori "true" model assumptions?

An affirmative answer to this question has been offered by the theory of algorithm complexity (c.f., [KAS7], [Lev73], [ML74], and [ZL70]).

In the theory of algorithm complexity, the information of a set of data is defined as the length of the shortest computer program which succeeds in duplicating the original data. The computer programs are those which can be obtained by a Universal Turning Machine (UTM) (The salient property of a UTM is that of making any statements in the theory independent of the type of computer.) An important characteristic of this set-up is that all partial recursive functions can be obtained with the help of the UTM.

This innovative definition of information is a totally data-dependent concept. It is the data which is imposing the shortest computer program and thus its associated algorithm information can be taken as an inherent property of the data. This is in sharp contrast to Shannon's information which is based on an a-priori "true" model class for the data. Unfortunately, the theory of algorithm complexity fails to provide a methodology by which to construct those shortest programs. This non-computability impairs its use as a practical modeling theory. Nevertheless, we should stress the contribution of the theory of algorithm complexity—aside from its paramount offerings to the field of computer science—for deepening the conceptual understanding of the modeling problem.

One could say that the stochastic complexity theory has flourished as a result of the blending of the theories of algorithm complexity, and information and coding. To paraphrase Rissanen, it successfully adds the missing components to each of its founding theories. It does so by replacing the UTM with probability model classes on one side, and by computing the information in the data relative to the model class chosen on the other. This emerging theory is now computable and detached from the assumptions of "true" models, making the whole modeling process data-dependent.

The collection of models from which tentative explanations of data are sought are

of the form of parametrized conditional compatible k-dimensional distributions

$$\mathcal{M}_k = \{P(\cdot,\theta); \theta \in D^k\}, \qquad \text{or} \qquad \mathcal{M}_k = \{P(\cdot,\theta), \pi(\theta); \theta \in D^k\}.$$

Model classes of the type $\mathcal{M}_k = \{P(\cdot,\theta), \pi(\theta); \theta \in D^k\}$ which are given in terms of a probability distribution $\pi(\theta)$ associated to the parameters of the model, do not have the connotation that one usually associate with them in the Bayesian framework. They do not intent to represent data independent prior knowledge but just to provide another possible way in which to encode the data with.

The utility function of the stochastic complexity theory by which models are compared, is constructed in terms of the least code length needed to encode the data with the help of a model class. Since the model itself—that is, its structure and parameters—has to be conveyed so that the receiver can replicate the original data, the encoding of the model is also counted as part of the overall cost of encoding the data. The criterion function is then defined as the least number of binary digits needed to encode the data with respect to the particular model class plus the number of bits needed to represent model structure and parameters. What is fundamental about the stochastic complexity utility function is that since both data and models are encoded in a similar manner, both model complexity and model fit are measured using the same benchmark: the codelength. Therefore, the competing issues of model complexity and model fit are taken into account in a natural sort of way.

The first general modeling method based on this coding notion was developed by J. Rissanen in [Ris78] . It rested on a two-part code construction and led to what is called the Minimum Description Length (MDL) criterion. We shall shortly describe it here, before introducing its latest but more abstract version, so as to make the ideas expressed above more transparent.

As a first step one computes the cost of encoding the data y^N with respect to a particular model in $\mathcal{M}_k$, which as seen previously can be taken as $-\log P(\mathrm{y}^N, \theta)$. Then

one needs to encode the model itself. This encoding can be done by first truncating the parameter θ up to a precision $\delta = 2^{-N_d}$, where N_d is the truncated parameter's total number of digits in its binary digit representation, and then constructing a prefix code for the truncated θ with codelength $-k \log \delta$. As a result, the overall codelength is given by

$$L(\mathrm{y}^N, \theta) \triangleq -\log P(\mathrm{y}^N, \theta) - k \log \delta. \tag{4.7}$$

The two-part MDL modeling criterion is then derived by first solving the following optimization problem:

$$\min_{\theta \in D^k, \delta} L(\mathrm{y}^N, \theta). \tag{4.8}$$

Note that this minimization involves two conflicting factors. For example, decreasing the precision δ decreases the length necessary for the encoding of the parameter θ (second term in (4.7), while increasing the cost of encoding the data y^N (first term in (4.7). This is so because

$$\tilde{\theta}_N(\delta_0) \triangleq \arg \min_{\theta \in D^k} \{-\log P(\mathrm{y}^N, \theta), \delta = \delta_0\} \tag{4.9}$$

will in general deviate from $\hat{\theta}_N$ which is the solution of (4.9) when no truncation is involved (i.e., $\delta \to 0$).

To solve the double minimization in (4.8), we proceed by performing a Taylor expansion of (4.7)

$$L(\mathrm{y}^N, \tilde{\theta}) \approx L(\mathrm{y}^N, \hat{\theta}_N) + \frac{1}{2} \bar{\delta}^T \Sigma \bar{\delta} - k \log \delta \tag{4.10}$$

where

$$\Sigma = \frac{\partial^2}{\partial \theta^2} L(\mathrm{y}^N, \theta) \Big|_{\theta = \hat{\theta}_N}$$

Now for the sake of simplicity assume $\Sigma = NI$, where I is the identity matrix with dimension k. Then simply minimize the dominant δ-term of the expansion (4.10), that is

$$\delta^* \triangleq \min_{\delta} \{N \frac{1}{2} |\bar{\delta}|^2 - k \log \delta\},$$

from which a simple computation gives $\delta^* = (N)^{-1/2}$. Finally, substituting δ^* in (4.10) we arrive at what is known as the two-part codelength MDL criterion:

$$\mathrm{MDL}(k) = -\log P(\mathrm{y}^N, \widehat{\theta}_N) + \frac{k}{2}\log N. \tag{4.11}$$

From the above construction we can conclude that the MDL criterion asymptotically reaches the minimum codelength among members of the model class $\mathcal{M}_k$ relative to this particular two-part coding strategy and the estimation method used to obtain $\widehat{\theta}_N$. We will later show that the MDL criterion does indeed reach the minimum codelength independently of, in this case, the particular two-part coding strategy.

Note that the MDL criterion asymptotically penalizes the increase of parametric complexity more heavily than the way the AIC does.

The stochastic complexity is an information theoretic measure of the complexity of a string of data relative to a model class. The model which achieves the stochastic complexity is the representative of all that can be known from the data about the "mechanism" that generated the data relative to the given model class. This stems from the fact that by definition the stochastic complexity reaches the least possible code length for the data and thus no more regular features can be extracted from it with the given model class. Model classes can then be compared by their associated stochastic complexity. We can loosely say that the length of the shortest encoding for a set of data y^N with respect to a model class $\mathcal{M}_k$ is a *stochastic complexity* of the data relative to the given model class.

There are some asymptotically equivalent computations of stochastic complexity. Their difference lies in whether the encoding is done in an on-line, batch or semi-batch manner. The non-predictive stochastic complexity (c.f., [Ris89]) is defined as

$$I(\mathrm{y}^N, \mathcal{M}_k) = -\log \int_D P(\mathrm{y}^N, \theta) d\pi(\theta). \tag{4.12}$$

It can be deduced by letting $\delta \to 0$ in the two-part code construct (c.f., [Ris89]). If a model class $\mathcal{M}_k = \{P(\cdot, \theta); \theta \in D^k\}$ is used then we can defined the conditional

distribution

$$\pi(\theta, y^N) = P(y^N, \theta) \left(\int P(y^N, \theta) d\theta \right)^{-1}$$

to construct $I(\mathrm{y}^N, \mathcal{M}_k)$.

Notice that the non-predictive stochastic complexity, defined as a mapping $I : (\mathrm{y}^N, \mathcal{M}_k) \to \mathbb{R}_+$, is detached from any subjective considerations since it has no free parameters to choose from. This is in sharp contrast to the MPE methods, like (4.1), where there is a multiple choice of criterion functions depending on the weighting of model fit and model complexity.

The main justifications for the stochastic complexity theory are as follows:

(i) For many important cases it can be proved that $I(\mathrm{y}^N, \mathcal{M}_k)$ is asymptotically a stochastic complexity of the data y^N with respect to the model class $\mathcal{M}_k$.

Those are the cases in which the tail-condition is known to be satisfied, as for example in a Gaussian ARMA set-up. The justification that in fact the non-predictive stochastic complexity $I(\mathrm{y}^N, \mathcal{M}_k)$ does correspond to a stochastic complexity of a set of data, at least for models satisfying the tail-condition, can be easily derived from Theorem 4.2.1. Indeed, by rewriting (4.6) as

$$-\mathbb{E}_\theta \log g_N(y^N) \geq -\mathbb{E}_\theta \log f_N(y^N, \theta) + \frac{k - \epsilon}{2} \log N \tag{4.13}$$

for all $\epsilon > 0$ and for a large enough N, and comparing it with (4.11) we find that the MDL criterion reaches asymptotically the lowest bound for the total codelength. Also since the non-predictive stochastic complexity can be shown to be asymptotically equivalent to the MDL as the truncation precision tends to zero (c.f. [Ris89]), $I(\mathrm{y}^N, \mathcal{M}_k)$ also reaches the lower bound.

(ii) The stochastic complexity theory can be taken as a generalization of the maximum likelihood method when models of different complexity are to be compared. Let $P(x) = 2^{-I(x)}$. Then $P(x)$ is a probability function with the property of globally maximizing the likelihood of the data.

(iii) To any type of prediction errors we can associate codelengths whose minimized value is the stochastic complexity.

It is easy to show that up to a constant any prediction error measure can be viewed as an equivalent codelength derived criterion [Ris89]. For example, let $\widehat{y}_N$ be any predictor for a (y_N) Gaussian ARMA (p, q) process. Now let $\epsilon_N = \widehat{y}_N - y_N$. Then

$$f_\theta(y_{N+1}|y^N) = K(y^N)2^{(\epsilon_N)^2}$$

with $K(y^N)$ such that $\int f_\theta(x|y^N)dx = 1$ defines a proper density function and thus we can associate the codelength

$$-\log f_\theta(y_{N+1}|y^N) = (\epsilon_N)^2 - \log K(y^N).$$

Moreover, we can affirm the following theorem since the tail-condition is known to be satisfied for Gaussian ARMA processes (c.f., [GR86] and also [Kab88] for a related result).

Theorem 4.2.2 *Let a Gaussian ARMA(p, q) satisfy Conditions 3.1.1 and 3.1.2. Let (ϵ_n) be any prediction error process, then*

$$\varliminf_{N\to\infty} \mathbb{E} \sum_{n=1}^{N} (\epsilon_n^2 - e_n^2)/\sigma^2(p+q)\log N \geq 1$$

except for a set of ARMA parameters of measure 0.

4.2.1 Predictive Stochastic Complexity

The non-predictive stochastic complexity given in (4.12) is computed off-line thus making it impractical for a large class of important problems, like those involving dynamical systems which frequently ought to be solved in real time. For this reason, an approximation of the stochastic complexity which is computed on-line was introduced by [Ris86] (see also [Daw84]). It is called predictive stochastic complexity. It is also an information theoretic measure of the complexity of a string of data, but in this case it is not only relative to a model class but also to a particular estimation method. The on-line feature distinguishes the predictive stochastic complexity theory from other modeling methods such as the AIC or BIC, (c.f. [Aka70], [Aka73], [Sch78], [Saw78]) which are inherently off-line.

It will be shown that the predictive stochastic complexity is a mathematically well understood criterion, which can be used in solving model selection problems in real time. It can also be used as a fundamental tool to solve other important problems such as adaptive control, the estimation of nonparametric transfer functions, and change-point detection (c.f., [BG90], [GB91], [GB90], and [BG92a]). For some applications in other areas of statistics the reader is referred to [QR89] and [WB68].

Predictive stochastic complexity is defined in terms of predictive encoding, which can also be considered a universal coding procedure (for related works in this area see [LJ74], [Ris84], and [ZL78]). Let us first then introduce the predictive encoding concept through an example of its implementation.

Suppose we pick a model class represented by the family of densities $\{f(\cdot,\theta), \theta \in D\}$ to model a set of data y^N. Assume for the sake of simplicity that $k = 1$. Now, recall that for any trial parameter $\theta \in D$, a prefix code $C(\mathrm{y}_i,\theta)$ with $i = 1,\ldots,N$ can be constructed (c.f., Section 2.3) such that for an observation y_i we get the associated code length

$$L(\mathrm{y}_i,\theta) = -\log f(\mathrm{y}_i,\theta).$$

The predictive encoding proceeds as follows:

(i) Pick up any initial guess $\theta_0 \in D$ and encode the first observation y_1 with the code $c_1 = C(y_1, \theta_0)$.

(ii) Compute the ML estimator $\hat{\theta}_1$ of θ^* using y_1.

(iii) Use the estimator $\hat{\theta}_1$ to encode the next datum y_2 as $c_2 = C(y_2, \theta_1)$.

(iv) Go to item (ii) and repeat the procedure until the whole data sequence is encoded.

As an end result we get the generation of the sequence of codes $c_1, \ldots, c_N$.

Only if we can decode the sequence $c_1, \ldots, c_N$, and get the original data y^N sequentially as the data is transmitted, could we then conclude that we had constructed a predictive encoding procedure. Let us show that this is the case. As a first step assume that the information about the model class $\{f(\cdot, \theta), \theta \in D\}$, and the initial condition θ_0 is known (i.e., it has already been transmitted to the decoder). The decoding is done as follows:

(i) When the decoder receives the code c_1, it can certainly compute y_1 by solving the equation $C(y_1, \theta_0) = c_1$.

(ii) Then, it can compute $\hat{\theta}_1$ by solving the ML equation.

(iii) Once c_2 is received, the observation y_2 can be recovered by solving $C(y_2, \hat{\theta}_1) = c_2$. Again $\hat{\theta}_2$ is obtained form the ML equation.

(iv) The decoding is then repeated sequentially until the sequence $c_1, \ldots, c_N$ has been exhausted.

A very important consequence of the predictive encoding procedure is that

$$F(\mathrm{y}^N, \theta_0) = f(\mathrm{y}_1, \theta_0) f(\mathrm{y}_2, \hat{\theta}_1(\mathrm{y}_1)) f(\mathrm{y}_3, \hat{\theta}_2(\mathrm{y}^2)) \cdots f(\mathrm{y}^N, \hat{\theta}_{N-1}(\mathrm{y}^N)),$$

can easily be shown to define a density for the data y^N. This shows that a density $F(\mathrm{y}^N, \theta_0)$ for the data $\mathrm{y}_1, \ldots, \mathrm{y}_N$ is being learnt or constructed, as data becomes available. Notice that this is done through a truly sequential construction.

One of the main features of the predictive encoding procedure is that it can be employed as a methodology for choosing amongst different model classes. For instance, take the case where two model classes $\mathcal{M}_1 = \{f(\cdot, \theta), \theta \in D_1\}$, and $\mathcal{M}_2 = \{g(\cdot, \psi), \psi \in D_2\}$, are considered for a given data sequence. Then, the predictive encoding procedure will provide us with coding sequences $c_1^f, \ldots, c_N^f$, $c_1^g, \ldots, c_N^g$, and densities $F(\mathrm{y}^N, \theta_0)$ and $G(\mathrm{y}^N, \psi_0)$. By comparing the total code lengths $\sum_{i=1}^N c_i^f$ and $\sum_{i=1}^N c_i^g$, the model class that associates the minimum total code length is then chosen. Moreover and for example, if the model class $\mathcal{M}_1$ is finally chosen then

$$F(\mathrm{y}^N, \theta_0) \leq G(\mathrm{y}^N, \theta_0). \tag{4.14}$$

Therefore, according to (4.14), the best model corresponds to the constructed density that gives us the maximum probability with respect to the data. Clearly this resembles the ML idea and represents one of the basic facts that serves as a foundation of the theory of stochastic complexity.

Definition 4.2.2 For the model class described by the family of densities $\{f(\cdot, \theta); \theta \in D\}$, the *predictive stochastic complexity* is defined by

$$I_p(\mathrm{y}^N, \mathcal{M}_k) = \sum_{n=1}^{N} -\log f(\mathrm{y}_n, \hat{\theta}_{n-1})$$

where $\hat{\theta}_{n-1}$ is an estimator obtained using only the data y^{N-1} by means of some type of estimation algorithm.

Since the predictive stochastic complexity can be shown to be asymptotically equivalent to the non-predictive stochastic complexity (c.f., [Ris89]) then I_p also reaches the lower bound as specified in Theorem 4.2.1.

Example 4.2.1 For the linear regression given by (2.6), define the prediction error process

$$\epsilon_n(\theta) = \mathrm{y}_n - x_n^T\theta.$$

Then it is easy to see that the predictive stochastic complexity is given by

$$\sum_{n=1}^{N} \left(\epsilon_n(\widehat{\theta}_{n-1})\right)^2.$$

Note that the prediction error $\epsilon_n(\widehat{\theta}_{n-1})$ is—using Rissanen's terminology—"honest" since for its computation we only use data which precedes the moment n. Compare this to the AIC approach where $\epsilon_n(\widehat{\theta}_N)$ is not computable at time n.

Let us now turn to the analysis of predictive stochastic complexity for the i.i.d. case. Recall that at step n the code length is given by $-\log f(\mathrm{y}_n, \widehat{\theta}_{n-1})$. Assume that the data was actually generated by the density with parameter θ^*. Then clearly the optimal encoding will be $-\log f(\mathrm{y}_n, \theta^*)$. We would like to investigate how much we have to "pay" for not knowing θ^*. The following theorem was proved by Davisson for the AR case (c.f., [Dav65]).

Theorem 4.2.3 (Davisson's formula) *Under certain regularity conditions (c.f., [Dav65]) imposed on the density $f(\cdot, \theta^*)$,*

$$\mathbb{E}_{\theta^*}(-\log f(\xi_n, \widehat{\theta}_{n-1}) + \log f(x_n, \theta^*)) = \frac{k}{2n}(1 + o(1)).$$

(Hint: Perform a second order Taylor-series expansion of the left hand side).

The above theorem can be interpreted as a statement about the difference between two mean per symbol code lengths: one based on the knowledge of θ^*, and the other on the model class only.

Example 4.2.2 The cummulative effect of parameter uncertainty for the linear regression of Example 4.2.1, is given by

$$\mathbb{E}\frac{1}{2}\sum_{n=1}^{N}(\epsilon_n^2(\widehat{\theta}_{n-1}) - e_n^2) = -\frac{k}{2}\log N(1+o(1)).$$

If instead of the "honest" estimators $\widehat{\theta}_{n-1}$ we were to use $\widehat{\theta}_N$, the cumulative effect of parameter uncertainty for the regression case would be

$$\mathbb{E}\frac{1}{2}\sum_{n=1}^{N}(\epsilon_n^2(\widehat{\theta}_N) - e_n^2) = -\frac{k}{2}(1+o(1)) \tag{4.15}$$

Observe that the left hand side of (4.16) cannot be interpreted as a codelength, since $\widehat{\theta}_N$ is unknown to the decoder. However, if we transmit $\widehat{\theta}_N$, which increases the length of the message by ck bits, $c > 1/2$ a constant, then decoding becomes possible. Let

$$J(k) \triangleq \mathbb{E}\frac{1}{2}\sum_{n=1}^{N}\epsilon_n^2(\widehat{\theta}_N) + ck \tag{4.16}$$

then models can be compared through their associated $J(k)$'s values. Note that for $c = 1$, (4.16) corresponds to Akaike's information criterion.

One main disadvantage of AIC type criteria is that they are inherently off-line. Therefore, for example, they are not suitable for use in real-time control systems. This is a direct consequence of its definition since at each time n future values of the data are needed to compute the estimate $\widehat{\theta}_N$.

It is easy to see that $J(k)$ penalizes overparametrization. Indeed, let $\psi^* = (\theta^*, 0)$ be an extended parameter, say $\dim\psi^* = k' > k$. Then we have

$$\mathbb{E}\frac{1}{2}\sum_{n=1}^{N}(\epsilon_n^2(\widehat{\psi}_N) - e_n^2) = -k'(1+o(1)). \tag{4.17}$$

Substracting $\sum_{n=1}^{N} e_n^2$ from equation (4.17) we conclude that

$$\mathbb{E}\,(J(k') - J(k)) = (k'-k)(1+o(1)).$$

Although the predictive stochastic complexity is computationally intensive in its original form, a suitable modification for the ARMA case with great potential for generalization is now available. The important fact is that it does not affect the asymptotic properties of the original predictive stochastic complexity (c.f., Theorem 3.6 in [GR91]). Ongoing research indicates that this important step can also be carried out in the multivariable case.

4.3 Model Order Selection for ARMA Models using Predictive Stochastic Complexity

The present section will be limited to the specific but difficult problem of finding the best model order for a set of data among ARMA models of different order. This is an important problem in the statistical theory on linear stochastic systems. The AR case was analyzed by [HMP89] and [HD89]. The latter work is based on the work of [Wei87]. The analysis of the significantly more difficult ARMA case was settled in [Ger89c]. This work provides a computationally feasible version of predictive stochastic complexity which will be used in the model order selection of ARMA systems. Moreover, we shall show that the predictive stochastic complexity modeling method is consistent for a certain types of ARMA models.

The main result of [Ger89c] is that under certain not too restrictive conditions we have

$$\lim_{N\to\infty} \sum_{n=1}^{N} ((\epsilon_n^0)^2 - (e_n)^2)/\sigma^2(p+q)\log N = 1 \quad \text{a.s.} \tag{4.18}$$

where (e_n) is the input noise process, $\sigma^2 = \mathbb{E}\,(e_n)^2$, ϵ_n is an "honest" prediction error, and p, q are overestimated values of the true orders p^*, q^* (but only one of them is strictly greater than the true order).

While this result is certainly of great interest it may be criticized from a practical

point of view since the sensitivity of the criterion $\sum_{n=1}^{N}(\epsilon_n^0)^2$ to overestimation is not very marked. Using instead the fixed gain prediction-error process ϵ_n^λ, the sensitivity of the criterion to overmodeling is increased, as it will be described later. However, fixed gain recursive estimation methods are not as well understood as the traditional estimation methods. Hence, the asymptotic properties of the predictive stochastic complexity associated with a fixed gain recursive prediction error seems to be very difficult at present. A less challenging project, which was undertaken in [Ger92b], is the analysis of predictive stochastic complexity associated with the off-line fixed-gain estimator. Some of the result that follow can be found in [GB90] and [BG92a].

4.3.1 Technical Conditions and Main Theorems

Let us first introduce some of the technical conditions which are needed to solve the modeling problem for ARMA classes. Let $(y_n), n = 0, \pm 1, \pm 2, \ldots$ be a second order stationary ARMA(p,q) process satisfying the following difference equation:

$$A^* y = C^* e.$$

Condition 4.3.1 *The input process (e_n) is a discrete-time, second order stationary, L_0-mixing process, and satisfies Condition 3.1.2.*

Let

$$D_{p^*,q^*} = \{(p,q) : p \geq p^* \text{ and } q = q^* \text{ or } p = p^* \text{ and } q \geq q^*\}. \tag{4.19}$$

denote the set of model orders describing overestimated structures. For each $(p,q) \in D_{p^*,q^*}$ we define the "true" parameter by appropriately augmenting $(p-p^*)+(q-q^*)$ zeros to the parameter-vector $(a_1^*, \ldots a_{p^*}^*, c_1^*, \ldots c_{q^*}^*)$.

Let $G \subset \mathbb{R}^{p+q}$ denote the set of θ's such that the corresponding polynomials $A(z^{-1})$ and $C(z^{-1})$ are stable. G is an open set. Let D^* and D be compact domains such that $\theta^* \in D^* \subset \text{int}D$ and $D \subset G$. Here int$D$ denotes the interior of D.

Let us define

$$S_N^\lambda(p,q) = \sum_{n=1}^{N} \left(\epsilon_n^\lambda\right)^2.$$

(For the definition of ϵ_n^λ refer to Section 3.2.) In the Gaussian case $S_N^\lambda(p,q)$ is a predictive stochastic complexity relative to the ARMA(p,q) model class and the recursive fixed-gain off-line prediction error estimation method. Note that $\epsilon_n^\lambda(\hat{\theta}_{n-1}^{\lambda})$ is "honest" in the terminology of [Ris86], i.e. to generate the prediction error process we only use data preceding the moment n.

Now let us denote

$$\Delta p_N^\lambda(p,q) = S_N^\lambda(p+1,q) - S_N^\lambda(p,q),$$

and

$$\Delta q_N^\lambda(p,q) = S_N^\lambda(p,q+1) - S_N^\lambda(p,q).$$

Moreover, denote $\Delta_N^\lambda(p,q)$ whenever $\Delta p_N^\lambda(p,q)$ or $\Delta q_N^\lambda(p,q)$ can be considered. The next theorem captures the excess of predictive stochastic complexity between consecutive ARMA(p,q) model classes when only one of the model orders is overestimated.

Theorem 4.3.1 ([Ger92b]) *Under Conditions 3.1.2 and 3.1.1 and for* $(p,q) \in D_{p^*,q^*}$, *we get*

$$\overline{\lim_{N\to\infty}} \left| N^{-1}\Delta_N^\lambda(p,q) - \lambda\sigma^2/2 \right| = C\lambda^{1/2}, \quad a.s. \tag{4.20}$$

where C *is a nonrandom constant.*

We recall (c.f., [Ger86]) that when the predictive stochastic complexity is taken relative to the recursive time invariant off-line prediction error estimation method, we have for $(p,q) \in D_{p^*,q^*}$

$$\lim_{N\to\infty} (\log N)^{-1}\Delta_N^0(p,q) = \sigma^2 \quad \text{a.s.} \tag{4.21}$$

Comparing (4.20) and (4.21) we can clearly see that the fixed-gain predictive stochastic complexity $S_N^\lambda(p,q)$ is qualitatively much more sensitive to overparametrization

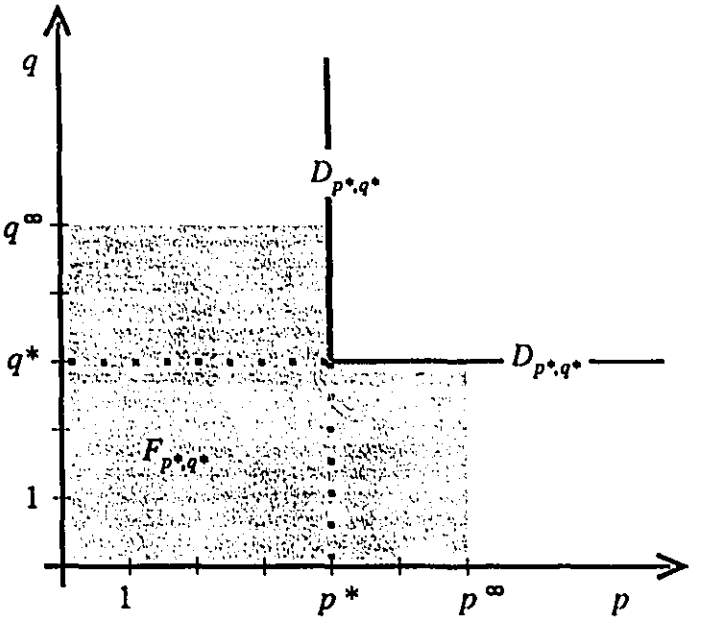


Figure 4.1: **The under-parametrization and over-parametrization regions.**

than the time invariant predictive stochastic complexity $S_N^0(p,q)$. The startling fact is that the "badness" of the estimator increases qualitatively the "badness" of over-parametrization.

Let p^∞, q^∞ be a-priori upper bounds for the unknown true order models p^*, q^*, and let us consider the set of model order pairs

$$F_{p^*,q^*} = \{(p,q) : p \le p^\infty \text{ and } q \le q^* \text{ or } p \le p^* \text{ and } q \le q^\infty\}.$$

In Figure 4.1 the regions D_{p^*,q^*} and F_{p^*,q^*} are illustrated.

It is a well-known maxim that in the underparametrization region, more precisely for $(p,q) \in F_{p^*,q^*} \backslash D_{p^*,q^*}$ we get for small λ's

$$\lim_{N\to\infty} N^{-1}\Delta_N^0(p,q) < \delta_1 < 0 \quad \text{a.s.}$$

where δ_1 is a constant (c.f., [DII89]). Simulation studies show that a similar result holds true for $\Delta_N^\lambda(p,q)$, i.e., for small λ's

$$\lim_{N\to\infty} N^{-1}\Delta_N^\lambda(p,q) < \delta_2 < 0 \quad \text{a.s.} \tag{4.22}$$

where δ_2 is also a constant.

4.3.2 Selecting the Best ARMA(p,q) Model

We shall see that the predictive stochastic complexity $S_N^\lambda(p,q)$ can be successfully used for model order estimation when λ is small. The following theorem gives the conceptual framework:

Theorem 4.3.2 ([GR91]) *Let the conditions of Theorem 4.3.1 and the validity of (4.22) hold, and denote $\widehat{p}_N, \widehat{q}_N$ the solution to the problem*

$$\min_{(p,q)\in F_{p^*,q^*}} S_N^\lambda(p,q).$$

Then for sufficiently large $N's$, $N > N_0(\omega)$, we have $\widehat{p}_N = p^*$ and $\widehat{q}_N = q^*$.

REMARK. $S_N^\lambda(p,q)$ provides the only real-time computable criterion of model order estimation for ARMA systems.

Let us now outline the scheme to find an optimal ARMA model in practice:

i) Compute $\Delta p_N^\lambda(1,0)$. If $\Delta p_N^\lambda(1,0) > 0$ then stop the search. The optimal model is AR(1). If instead $\Delta p_N^\lambda(1,0) < 0$ then the optimal model is not AR(1) and thus continue with (ii).

ii) Compute $\Delta q_N^\lambda(0,1)$. If $\Delta q_N^\lambda(0,1) > 0$ then stop the search. The optimal model is MA(1). If instead $\Delta q_N^\lambda(0,1) < 0$ then set $p = 1$ and $q = 1$ and continue the search according to the next item.

iii) Start moving from point $(1,1)$ along the line $\{(p,q); p = q\}$, in the plane with axis given by the tentative model orders p and q, as long as $\Delta p_N^\lambda(p,q)$ and $\Delta q_N^\lambda(p,q)$ are negative. That is, the decision to move along $\{(p,q); p = q\}$ is done by comparing the ARMA$(p+1,q)$ and ARMA$(p,q+1)$ model classes with the ARMA(p,q) model class. (Here we are applying (4.22) which characterizes the behaviour of both $\Delta p_N^\lambda(p,q)$ and $\Delta q_N^\lambda(p,q)$ in $F_{p^*,q^*} \backslash D_{p^*,q^*}$.) Now, when at least $\Delta p_N^\lambda(p,q)$ or $\Delta q_N^\lambda(p,q)$ becomes positive then $p = p^*$ and/or $q = q^*$. That is, according to (4.20) we are comparing ARMA model classes along the region D_{p^*,q^*}. To determine where we actually hit the region D_{p^*,q^*}, compute $\Delta p_N^\lambda(p-1,q)$ and $\Delta q_N^\lambda(p,q-1)$. Then one of the following situations might arise:

a) If $\Delta p_N^\lambda(p-1,q) < 0$ and $\Delta q_N^\lambda(p,q-1) < 0$ then $p = p^*$ and $q = q^*$.

b) If $\Delta p_N^\lambda(p-1,q) < 0$ and $\Delta q_N^\lambda(p,q-1) > 0$ then $p = p^*$ and $q \geq q^*$.

c) If $\Delta p_N^\lambda(p-1,q) > 0$ and $\Delta q_N^\lambda(p,q-1) < 0$ then $q = q^*$ and $p \geq p^*$.

For cases (b) and (c) where the search is not over, proceed as follows. For case (b) compute $\Delta q_N^\lambda(p,q-1)$ for decreasing values of q, and when q is such that $\Delta q_N^\lambda(p,q-1) < 0$ then set $q = q^*$ and stop the search. Case (c) is done similarly.

Since the above heuristic description of the model order selection method leaves many important details open, let us make the following comments:

i) Trouble may occur in $F_{p^*,q^*} \backslash D_{p^*,q^*}$, the region where we underestimate the model order. Although increasing p or q by 1 decreases $S_N^\lambda(p,q)$ due to undermodeling by an amount proportional to N, it also increases $S_N^\lambda(p,q)$ due to parameter uncertainty by an amount proportional to N. If λ is not small enough, then the effect of parameter uncertainty may dominate, causing the excess of predictive stochastic complexity to become posite and thus we may

stagnate in $F_{p^*,q^*}\backslash D_{p^*,q^*}$. Thus, it is important to set λ small in $F_{p^*,q^*}\backslash D_{p^*,q^*}$, or try different λ's to ensure that this does not happen. This issue is illustrated in Section 4.3.5

ii) It would be important to choose N under the constraint of a desired lower bound for

$$P(\Delta_N^\lambda(p,q) < 0 \mid (p,q) \in F_{p^*,q^*}\backslash D_{p^*,q^*}),$$

and

$$P(\Delta_N^\lambda(p,q) > 0 \mid (p,q) \in D_{p^*,q^*}),$$

i.e. the probabilities of correct decisions.

iii) Instead of fixing an a-priori value for N, we can take it as the minumum N such that $\Delta_N^\lambda(p,q)$ has linear trend with fixed probability. This should reduce the computation search time.

iv) All the results of this section are in terms of the off-line fixed gain prediction error method. The reason being that the asymptotic properties of the predictive stochastic complexity associated with fixed-gain recursive prediction-error have not yet been analysed. However, we applied this algorithm in the simulation and it indicates that the proposed procedure works exceptionally well in real time.

4.3.3 AR Model Order Selection Simulations

In this section we will solve a model order selection problem by applying AR models. The purpose of starting with this simpler problem—as opposed to applying ARMA models to the order selection problem—is that it is simplifies the illustration of the effect of the fixed gain λ on the model order selection procedure.

We start by generating 2000 data points by means of a computer simulation of a time invariant AR(4) system, driven by a Gaussian white noise input process with mean 0 and variance 1.

The time invariant AR(4) system is given by

$$y_N + a_1^* y_{N-1} + \cdots + a_4^* y_{N-4} = e_N$$

with

$$a_1^* = .5 \qquad a_2^* = -.3 \qquad a_3^* = -.2 \qquad a_4^* = .4.$$

The order selection strategy is based on comparing the AR(p) to the model class AR($p+1$) class. The comparison is done by computing the predictive stochastic complexity associated with each AR model and calculating their difference. Thus, define

$$\Delta_N^0(p) = S_N^0(p+1) - S_N^0(p).$$

which is the excess of predictive stochastic complexity between the AR($p+1$) and AR(p) model class, when applying the time invariant prediction error algorithm.

The model order selection scheme presented in Section 4.3.2 is simplified significantly in the AR case since we now only have to search for order models on a line instead of a plane (as in the ARMA case). Thus, we just have to simply keep increasing the order of the AR model until the difference of the stochastic complexities associated with the AR(p) minus the AR($p+1$) models is negative.

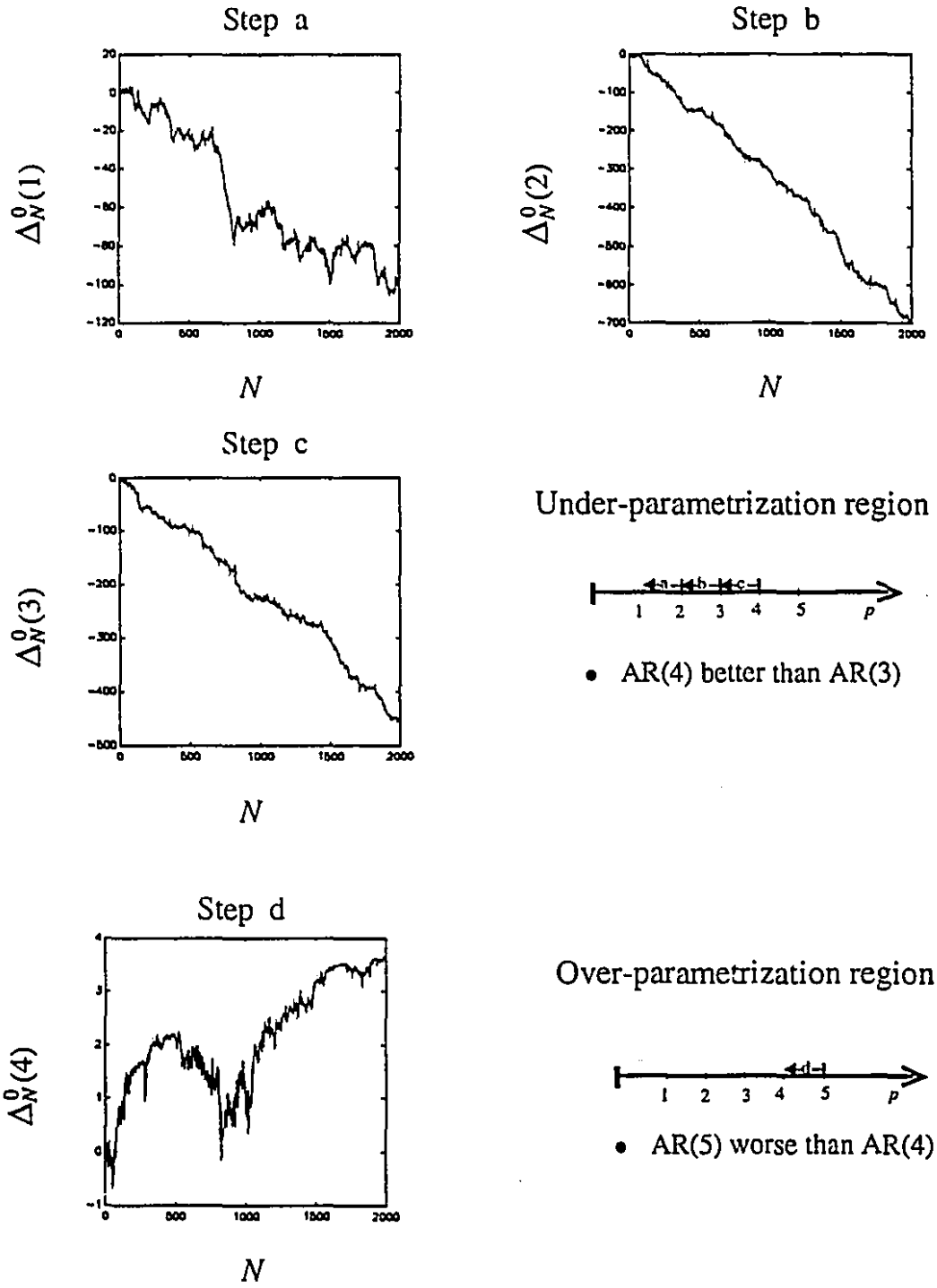


Figure 4.2: The difference of predictive stochastic complexities of adjacent AR models when using the time invariant prediction error estimation method.

Note the sort of logarithmic drift of the excess of predictive stochastic complexity in **Step d** of Figure 4.2. This is predicted by the theory when the time-invariant recursive prediction error method is used—see (4.18).

Figure 4.2 shows that there is not much point in increasing the order of the AR model beyond $p = 4$. Thus the true model order, $p^* = 4$, of the AR system was found.

Let us now repeat the previous experiment but instead use the fixed gain prediction error algorithm to generate the prediction error process. Thus, in this case define

$$\Delta_N^\lambda(p) = S_N^\lambda(p+1) - S_N^\lambda(p),$$

which is the excess of predictive stochastic complexity between the AR($p+1$) and AR(p) model class, when applying the prediction error algorithm with fixed gain λ.

We set the value of the fixed gain $\lambda = .0125$ and the results are illustrated in Figure 4.3. We then double the value of the fixed gain, that is we set $\lambda = .025$, and perform the experiment again. Figure 4.3 shows the results when using this value of λ. Similarly to the previous simulation, the same conclusion is drawn when using fixed gain: The scheme finds the true model order, $p^* = 4$, of the AR system.

The important observations that can be raised from these last two simulations are: i) The difference of the associated predictive stochastic complexities of adjacent AR models, when the model is overparametrized—see **Step d** of Figure 4.3 and 4.4—grows proportionally to the gain λ. This fact confirms the theoretical statement given in Theorem 4.3.1. Namely, the estimation "badness" increases the "badness" of overparametrization; ii) In the over-parametrization region, the difference of the associated predictive stochastic complexities of adjacent AR models have nearly linear drifts as opposed to the logarithmic drift obtained when the time invariant recursive prediction error method is used (i.e., **Step d** of Figure 4.2).

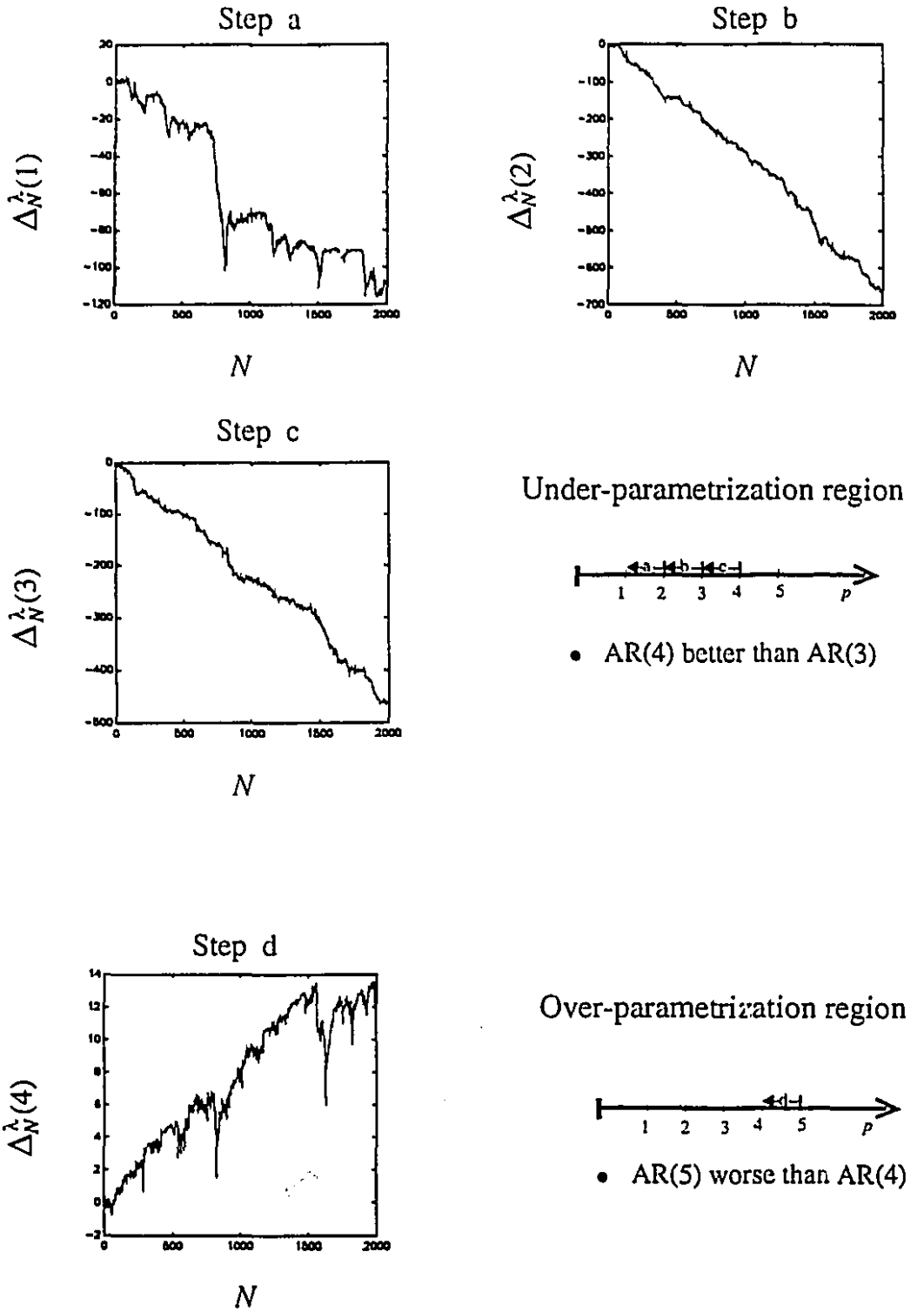


Figure 4.3: The difference of predictive stochastic complexities of adjacent AR models when using the fixed gain prediction error estimation method with $\lambda = .0125$.

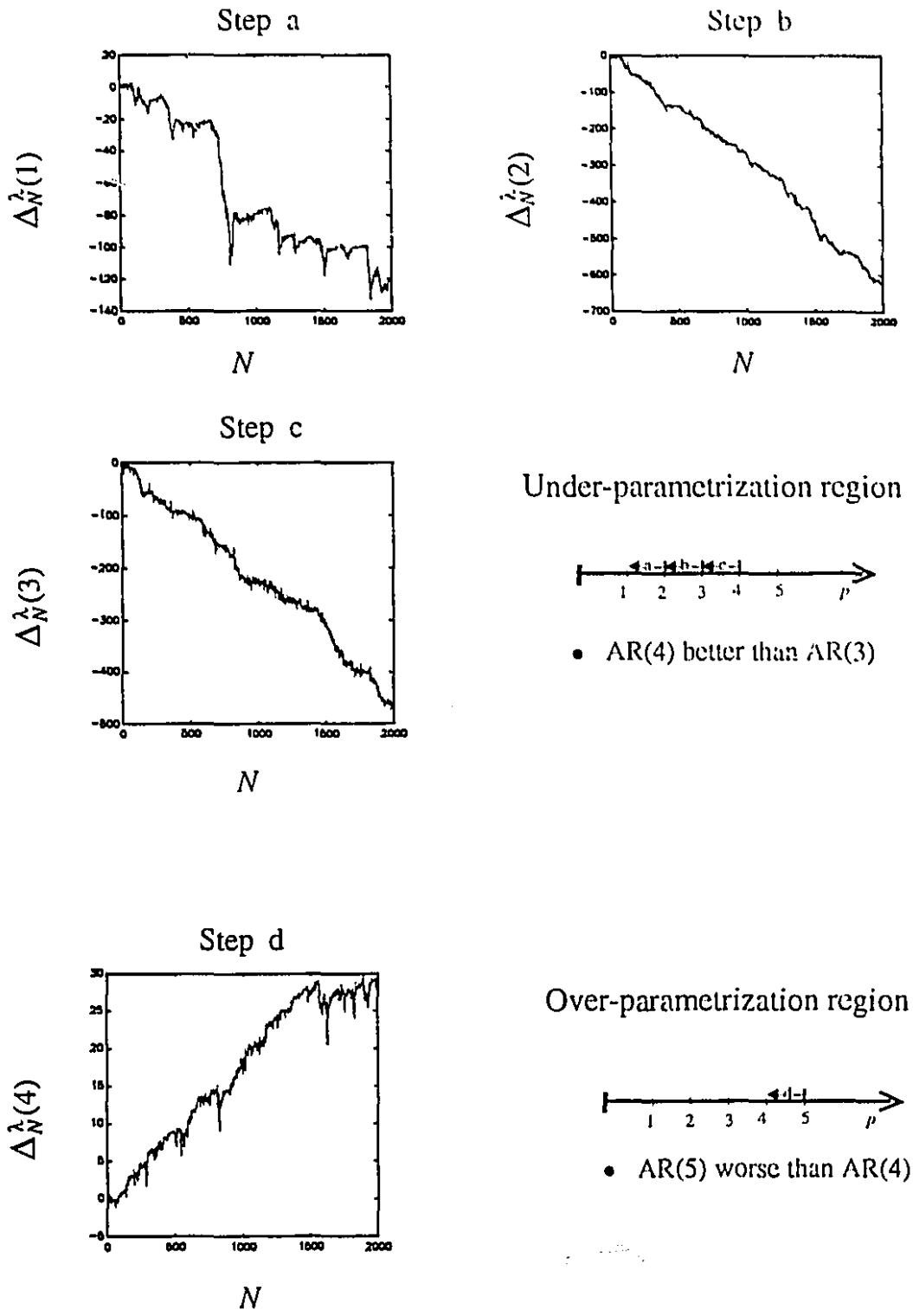


Figure 4.4: The difference of predictive stochastic complexities of adjacent AR models when using the fixed gain prediction error estimation method with $\lambda = .025$.

4.3.4 ARMA Model Order Selection Simulation

Let data be generated by the following ARMA(4,3) system

$$y_N + a_1^* y_{N-1} + \cdots + a_4^* y_{N-4} = e_N + c_1^* y_{N-1} + c_2^* y_{N-2} + c_3^* y_{N-3},$$

with

$$a_1^* = .3 \qquad a_2^* = .7 \qquad a_3^* = .3 \qquad a_4^* = .8,$$

and

$$c_1^* = .9 \qquad c_2^* = -.5 \qquad c_3^* = -.5.$$

The system is driven by a Gaussian white noise input process (e) with mean 0 and variance .5.

The simulation is run for 1000 data points, generating the realization y^N of y^N. In what follows we shall apply the model selection procedure presented in Section 4.3.2 to determine the best ARMA(p,q) model representation for the data y^N. The order selection strategy is based on comparing the ARMA(p,q) model with the ARMA models ARMA$(p+1,q)$ and ARMA$(p,q+1)$. The comparison is done by computing the predictive stochastic complexity associated with each ARMA model and calculating their difference. Recall that

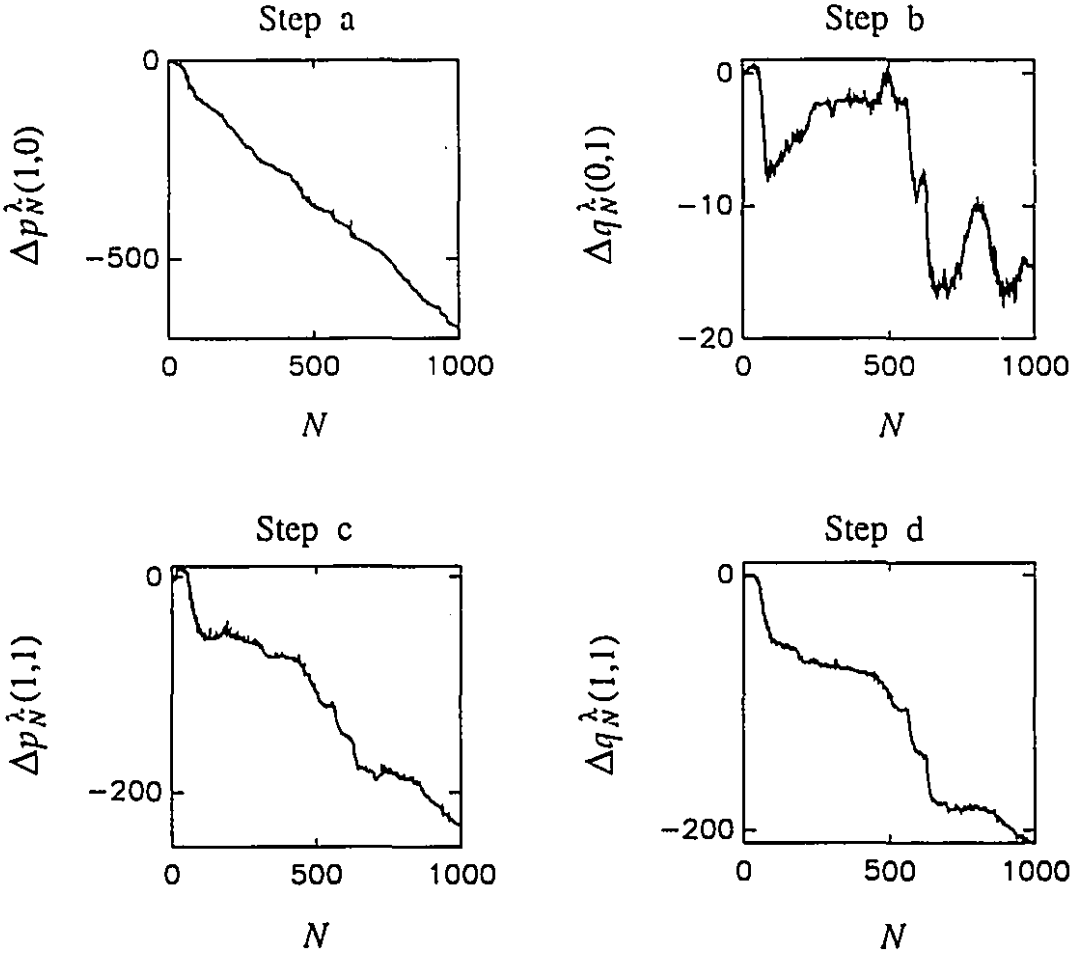
$$\Delta p_N^\lambda(p,q) = S_N^\lambda(p+1,q) - S_N^\lambda(p,q),$$

is the excess of predictive stochastic complexity of the ARMA$(p+1,q)$ model class when compared to the ARMA(p,q) model class, whereas

$$\Delta q_N^\lambda(p,q) = S_N^\lambda(p,q+1) - S_N^\lambda(p,q),$$

is the excess of predictive stochastic complexity of the ARMA$(p,q+1)$ model class when compared to the ARMA(p,q) model class.

Let us follow the model order selection outline of Section 4.3.2: Start with an AR(1) and MA(1) model classes and increase the orders of these models in the autoregressive and moving average directions until we hit the over-parametrization region. Figure 4.5 illustrate the behaviour of the difference of predictive stochastic complexity of adjacent ARMA model classes in the under-parametrization region. Since all the figures Figure 4.5 have negative drift we conclude that we should increase model orders p and q.



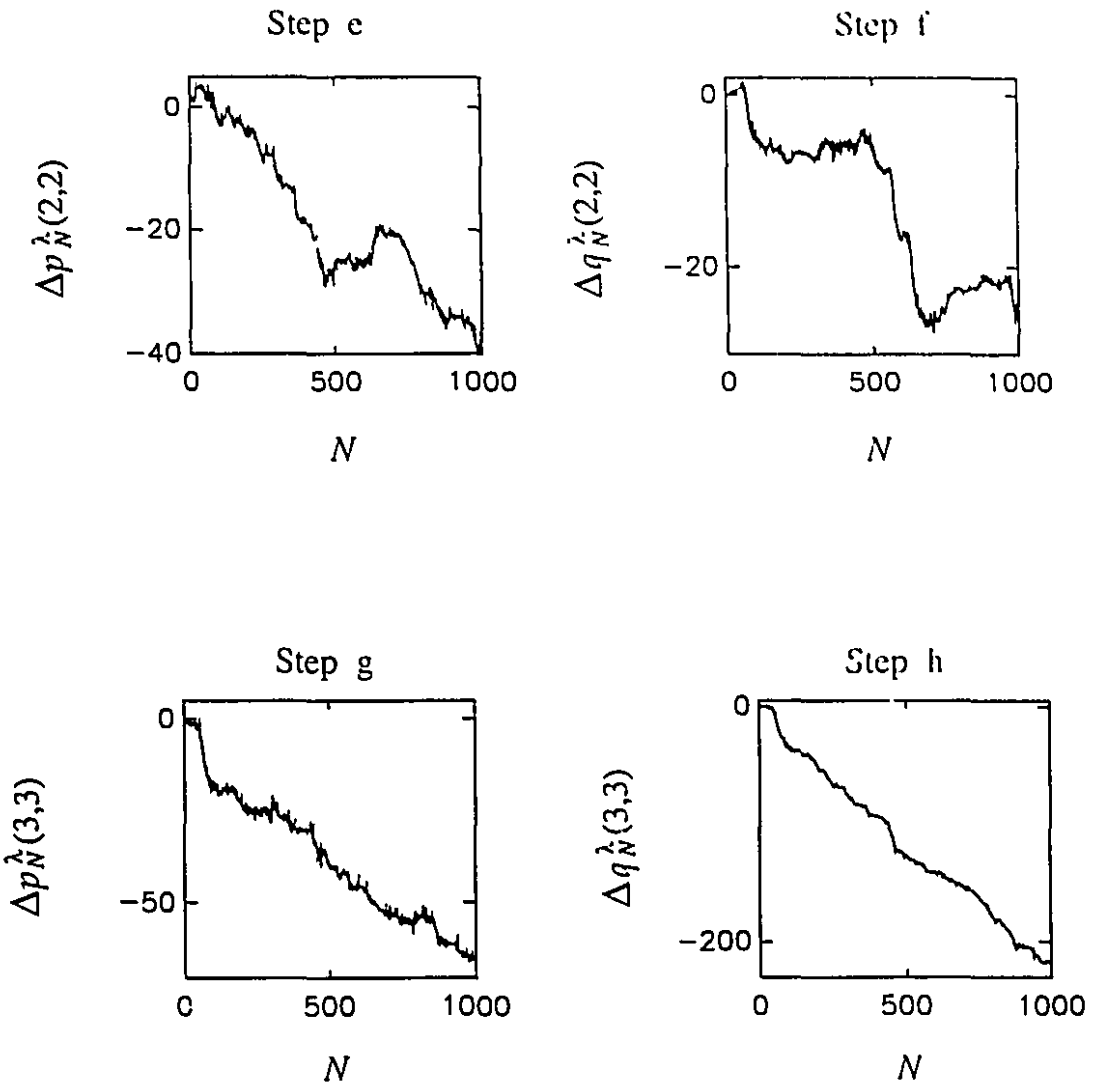


Figure 4.5: The difference of predictive stochastic complexities with gain $\lambda = .01$ for neighboring ARMA models in the under-parametrization region.

The search in the (p, q) coordinate system corresponding to the under-parametrization region is presented in Figure 4.6. The arrows in this figure have a one-to-one correspondence to the figures in Figure 4.6. For instance, the arrow e in Figure 4.5 represents the excess of predictive stochastic complexity taken with respect to the ARMA(3, 2) and ARMA(2, 2) model classes as plotted in Step e of Figure 4.5.

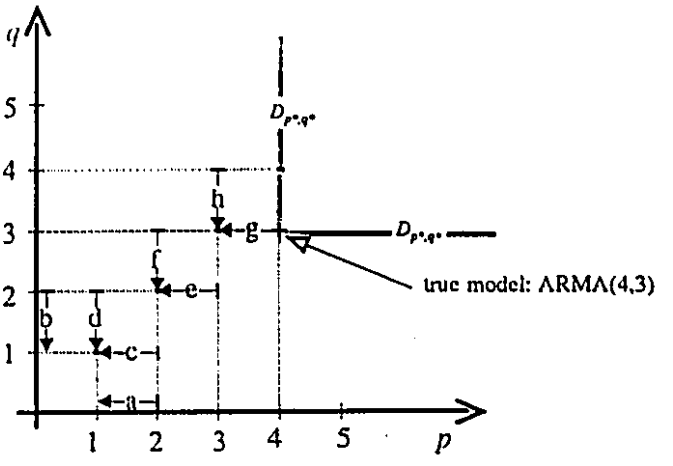


Figure 4.6: **First scheme steps: To reach over-parametrization region. (Each arrow corresponds to a figure in Figure 4.5.)**

Now, we increase the model orders of the ARMA classes in both directions. The results are illustrated in Figure 4.7. This figure shows that we actually did hit the over-parametrization region since $\Delta p^{\lambda}_{1000}(4,4) > 0$, and $\Delta q^{\lambda}_{1000}(4,4) > 0$. Therefore, we should decrease both model orders.

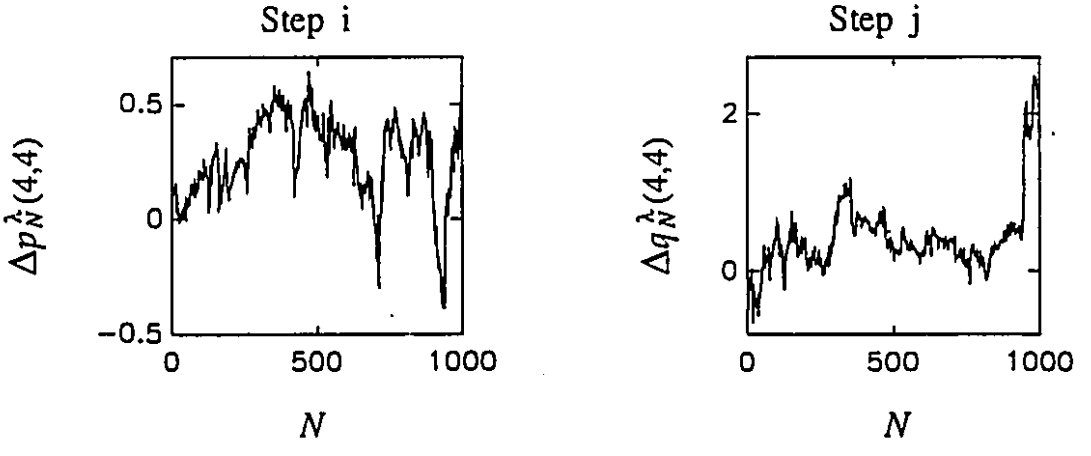


Figure 4.7: **The difference of predictive stochastic complexities with gain $\lambda = .01$ for neighboring ARMA models in the over-parametrization region.**

The search in the (p, q) plane is illustrated in Figure 4.8.

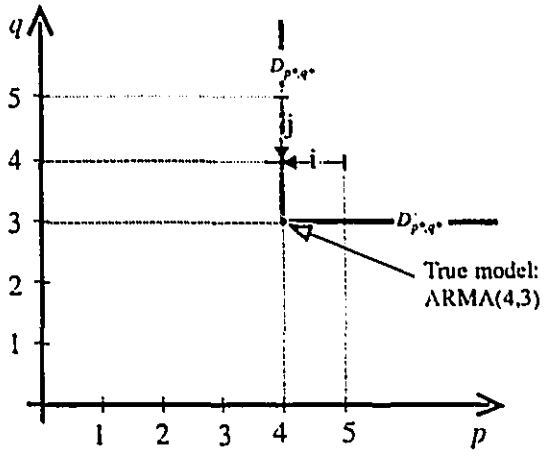
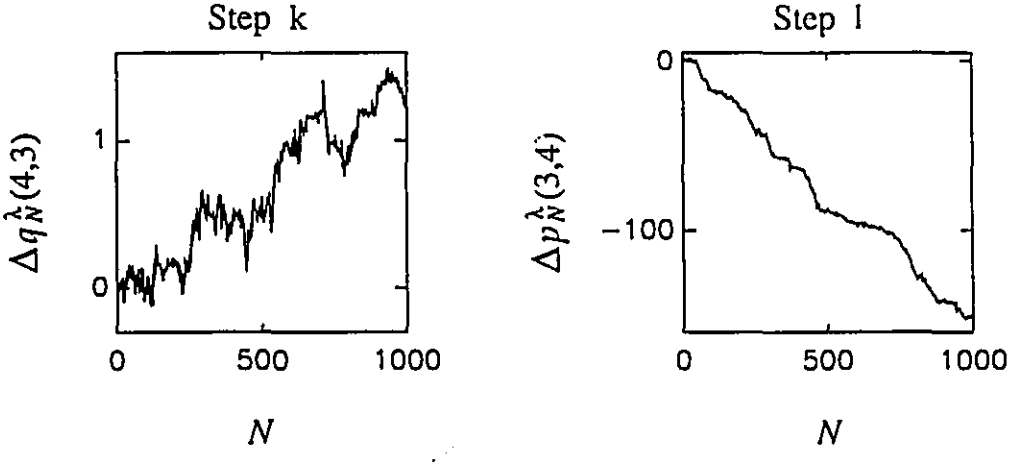


Figure 4.8: **Hit over-parametrization region. (Each arrow corresponds to a figure in Figure 4.5.)**

Finally, we decrease the model orders of the ARMA models classes until we get a negative drift for both $\Delta p_N^{\lambda}(p, q)$ and $\Delta q_N^{\lambda}(p, q)$. These last steps are shown in Figure 4.9 and Figure 4.10.



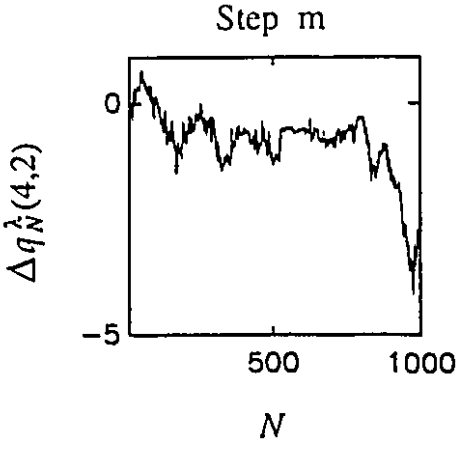


Figure 4.9: **The difference of predictive stochastic complexities with gain $\lambda = .01$ for neighboring ARMA models around the boundary of the over-parametrization region.**

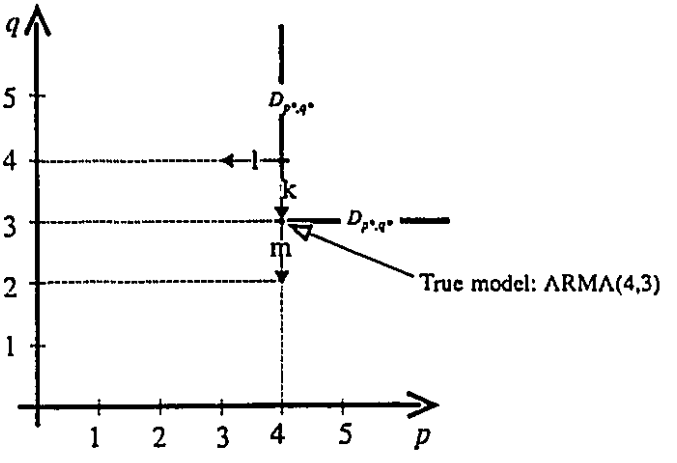


Figure 4.10: **Last search steps. True model order found.**

The optimal model was found to be ARMA$(4,3)$ which coincides with the order of the original ARMA model.

4.3.5 Simulation Example of Parameter Versus Model Order Uncertainty

This section will show how the effect of parameter uncertainty may dominate over model order uncertainty.

Let data be generated by the ARMA(2,1) $A^*y = C^*e$ system with parameter values

$$A^* = [1 \ -.5 \ .7] \qquad \text{and} \qquad C^* = [1 \ .01],$$

driven by a Gaussian white noise input process with mean 0 and variance .5. The gain of the fixed-gain recursive prediction error algorithm is set to $\lambda = .01$.

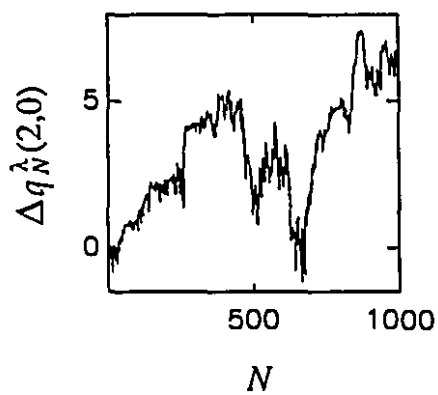


Figure 4.11: **The difference of predictive stochastic complexities with gain $\lambda = .01$ of an ARMA(2,1) and AR(2) models.**

In Figure 4.11 the difference of predictive stochastic complexities of an ARMA(2,1) model with an AR(2) model is illustrated. More precisely we compute $\Delta q_n^\lambda(2,1)$. Since $\Delta q_n^\lambda(2,1) > 0$ we conclude that $(2,0) \in D_{p^*,q^*}$ which does not correspond to the order of the original ARMA model. This is caused by the predominant effect of parameter uncertainty over model order uncertainty. When this occurs one can decrease λ or use the time invariant recursive prediction error algorithm.

Chapter 5

Change-Point Detection

5.1 Introduction

Change-point detection is the problem that deals with the estimation of the time or space location of quantitative and/or qualitative changes along the evolution of processes. In more concrete terms, a change-point represents a crucial behavioral alteration of the properties or characteristics of a physical system or signal which ought to be detected. The need for change-point detection procedures arises in both natural as well as human-made physical systems (referred to in the sequel as machineries or mechanisms), or combinations of both. Failures occurring in physical plants, in particular sensors and actuators, are typical examples of change-point type machine disruptions, whereas earthquakes and the diagnosis of brain and sound signals represent illustrations of natural systems. The applications of change-point detection techniques are currently being found in a variety of scientific activities such as mathematics, medicine, economics, and engineering.

When dealing with mechanisms, the term failure or fault detection is most commonly employed, since it accurately describes the fact that a total or partial breakdown of a component of a monitored machinery has occurred. For this type of physical

system, it is often the case that more than plain change-point detection schemes are needed as part of the overall supervisory strategy of physical plants. For example, some systems will also require the isolation of faults, that is the specific system's physical location of them. Methods that can achieve these two types of diagnostics are usually referred to as FDI (fault detection and isolation) methods. An even further step in a supervisory scheme is to entrench it with accommodation capabilities. That is, after the isolation step is completed successfully, adjustment to a new and acceptable configuration takes place. This could also include the on-line replacement of a broken part if provision for hardware component copies has been made beforehand. This more complete supervisory strategy is known as the FDIA (fault detection, isolation, and accommodation) scheme.

In contrast, when dealing with natural systems, the term change-point detection is most frequently employed. Clearly, there is a natural overlapping of the two camps since very often stochastic process signal models are used to monitor mechanisms. In this thesis we will use both terms, that is fault detection and change-point detection, indistinguishably.

Since modern machineries consist of interconnections of large numbers of components, most of them crucial to the overall system performance, the design of fault detection schemes is vital to any successful long-term system design. The role of failure detection schemes is to reveal possible malfunctions of components by giving some type of early warnings so that proper action can be taken. Successful implementations of these schemes will allow for the maintenance of an adequate level of plant performance and security, and if need be, might even avoid a catastrophe.

Traditionally, change-point detection schemes were utilized in quality control (c.f., [Pag54], [Tay68]), and the proposed solutions were known as control charts. Soon after, they were used to anticipate failures in physical systems. Typical examples are failures in sensors and actuators, fatigue of structures, and failures in nuclear plants.

These earlier applications have evolved to also include change-point detection in the recognition of signals, and as one of the basic features in the modeling stages of physical systems. For instance, they have been utilized for modeling variations in the operating conditions of physical systems. The basic fact that makes these schemes so useful is their ability to successfully deal with many kinds of non-stationary processes. Good illustrations are the modeling of speech, sequential segmentation of images (c.f., [BEG81]), and the diagnosis of ECG (electrocardiogram) and EEG (electroencephalogram) signals. For example in (c.f., [GWW+78] the detection and classification of cardiac arrhythmias from data coming from ECG signals has been favorably reported.

Other noteworthy examples of change-point detection applications are the design of fault detection in a robotic system [WF90], and the processing of geophysical signals [Bas86], where the jumps occurred as a consequence to signals traveling through different geological levels. Another example along this line is the study of earthquake produced ground motion to assist in the design of structures being built in seismic areas (c.f., [PD90]).

Fault detection has also entered the area of adaptive control. Since classical adaptive control theory is mainly suited to time invariant uncertain systems, or time variant systems with very small rates of variation, it became natural to extend the applicability of adaptive controllers to systems which present abruptly or slowly time-variant change-points (c.f., [RS73]). This can be attained by using the information given by a change-point estimator in the reshaping of the current control law.

For the most part, we shall not attempt to give a comprehensive review of all the work done in the last 20 years in the area of change-point detection. Although the literature of fault detection is not as extensive as in other fields in systems and control, a variety of survey papers and books have already been devoted to this area. The

reader is referred to the survey papers of [Wil76], [Mir80], [Ise84], [Bas88], [Lom89], and [Nik91]. There are also some important books written on the subject and these include [BB86], and [Tel86].

Our goal here is to underline the general problem formulation, and highlight some of its most essential features and difficulties. The presentation should also serve as a self-contained introduction to the topic, assisting readers unfamiliar with the area in understanding the change-point detection method which will be developed in subsequent sections.

5.2 First Elementary Detection Methods

In the early stages of the development of change-point detection procedures, a naive approach was used to tackle the problem. It consisted of using unprocessed measured signals, say $y(t)$, until they surpassed some a-priori defined fixed threshold h. This simply meant that the following trivial scheme was used:

$$y(t) = \begin{cases} < h & \text{no alarm;} \\ \geq h & \text{alarm on,} \end{cases}$$

which has been generally known as limit checking. Surprisingly enough, this type of procedure was still in use by parts of the Space Shuttle Columbia's monitoring system when it exploded in mid-air (c.f., [Cik86]).

Another earlier failure detection method which was implemented in machineries, consisted of processing the information given by exact hardware copies of the monitored parts of a plant. This method is called hardware redundancy. A faulty component will produce a different signal from that of the others and thus can be discerned using a simple memoryless voting system. For example, let (y_1), (y_2), and (y_3) be the signals produced by three identical system hardware components and consider

for some $c > 0$ the function

$$r(t) = I_{|y_1(t)-y_2(t)|>c} + I_{|y_1(t)-y_3(t)|>c} + I_{|y_2(t)-y_3(t)|>c},$$

where I_A denotes the indicator function of the set A. Then

$$r(t) = \begin{cases} < 2 & \text{no alarm;} \\ \geq 2 & \text{alarm on.} \end{cases}$$

Under the event $\{r(t) = 2\}$, discrete decision logic can be used to discard the faulty component. A more involved voting system approach can be found in [Bro74]. An intrinsic drawback of voting systems is that they have difficulties detecting what are usually referred to in the literature as soft faults, for instance small shifts in bias. In [FG82], a procedure is given for the estimation of sudden jumps in bias vectors for linear systems.

The hardware redundancy FDI procedure is still being used due to its simplicity and to the fact that FDI methods have not yet matured to the level of providing robust fault diagnosis schemes good enough to make voting systems obsolete. The drawbacks of hardware redundancy are evident: cost, and depending on the situation, also weight and physical space. Moreover, FDI schemes in voting systems will have an unacceptable long-term reliability. This is due to the fact that the component copies used by a voting system will all wear at similar rates. (Note that wearing is the main cause of hardware system's faults.)

Consider a fault detection scheme which can detect a system fault based on the signal produced by only one component. Manufacturing better hardware components will then add to the overall system reliability but in general could be either too costly or sometimes not possible to realize. As was just recently mentioned, using hardware redundancy will not significantly improve the long-term reliability. Any considerable long-term improvement in the scheme's reliability should then be obtained by ameliorating the software (i.e. the failure detection algorithms) rather than the hardware. An aeronautic example given in [DDDW77] shows that the number of hardware

backup components can even be effectively reduced, thus lowering the overall design cost.

The intrinsic drawbacks found in the first available change-point detection techniques, plus greater demands on safety, reliability and performance demanded more elaborate failure detection methods. The sophisticated and modern procedures originated in studies like the ones given in [Pag54], and [Shi61] in the beginning of the 60's, coinciding with the advent and rapid spread of computer technology. This coincidence was by no means accidental since the availability of digital computers opened a real possibility for the creation of sophisticated supervision systems. Another factor that contributed to the growth of this field was the concurrent progress that was taking place in control systems and system identification.

5.3 Change-Point Detection for Signals

In this section we will focus our attention on statistical methods for solving change-point problems in a stochastic framework. We shall mainly be concerned with the foundations of this problem.

5.3.1 The Off-Line Mathematical Formulation

Let us start with a formal mathematical statement of the problem. Let $y_1, \ldots, y_N$ be a set of data, and $\mathcal{M}$ a class of models, then the off-line change-point detection problem is defined as the issue of choosing between the hypothesis

$$H_0: \quad y_1, \ldots, y_N \quad \text{generated by } M_0 \in \mathcal{M},$$

and

$$H_1: \quad \exists \;\; 1 \leq \tau^* \leq N \quad \text{such that} \quad \begin{cases} y_1, \ldots, y_{\tau^*-1}, & \text{generated by } M_0' \in \mathcal{M}; \\ y_{\tau^*}, \ldots, y_N, & \text{generated by } M_1 \in \mathcal{M}. \end{cases}$$

H_0 represents the null hypothesis of no change-point in the data y^N and H_1, the alternative hypothesis indicating the presence of a change-point in the data sequence y^N. The assumption that τ^* is not known is the distinguishing feature of the change-point problem. If τ^* is known the result is the standard two-sample statistical problem (c.f., [Tay68]) Other related questions are the estimation of the change-point and the concurrent estimation of the models themselves.

Change-point detection problems for signals are always stated or transformed into a stochastic framework, meaning that the data is assumed to be a realization of a stochastic process, and the models are explicitly or implicitly described by probability distributions. Thus, for example, the models can correspond to conditional densities of the form

$$\mathcal{M} = \{P(y_n|y^{n-1}, \theta); \theta \in D_0\}.$$

As can be clearly seen from the problem statement, change-point detection falls into the more general topic of model selection. To choose between the multiple models implicitly given by hypothesis H_0 and H_1, is to decide which of these model descriptions best represents the set of data y^N. This represents one of the intrinsic difficulties of change-point detection problems since general model selection theories are only recently achieving adequate success (c.f., [Ris89]).

We are not aware of any work in the field of parametric failure detection in which the selection of the model class $\mathcal{M}$ was itself one of the central issues. The departing point has been the a-priori assumption that the model class $\mathcal{M}$ was the "true" model class representing the data. Therefore, no available technique had any provision for the comparison of different model classes $\mathcal{M}$ in its change-point detection formulation. We shall address this issue in Section 5.5 based on the ideas and tools provided by the stochastic complexity theory. Let us only mention now that under this framework the multitude of possible models represented by H_0 and H_1 correspond to only tentative explanations of the data. Thus the change-point will explicitly depend on

the particular model class, say $\mathcal{M}$, chosen. However, for the sake of simplicity, in the introductory exposition that follows let us assume that the data is actually generated according to one of the models in the class specified in the hypothesis H_0 and H_1.

To further simplify the discussion we will assume that the data exhibits only one change-point, and that without loss of generality $M_0 = M_0'$. Clearly, there is no loss of generality in imposing the first assumption if the change-point detection problem is to be solved on-line.

5.3.2 Change-Point Detection and Model Complexity

The complexity of the change-point detection problem depends above all on the assumption made about the model class $\mathcal{M}$. It can be a parametric or a nonparametric model class. In the nonparametric framework, we can mention the work of [BJ68] in which tests for a shift in the level of a stochastic process were developed, and [Pic85] for detecting a change in the spectrum of a time series. (See also [Pet79] and [DP86].) As an example of a non-parametric change-point detection method, let us present the Kolmogorov-Smirnov test defined by the statistic

$$S(\mathbf{y}^N) = \sup_x |\hat{F}_j(x) - \hat{F}_{N-j}(x)|$$

where

$$\hat{F}_j(x) = \frac{1}{j}\sum_{i=1}^{j} I_{\mathbf{y}_i \leq x} \qquad \hat{F}_{N-j}(x) = \frac{1}{N-j}\sum_{i=j+1}^{N} I_{\mathbf{y}_i \leq x}$$

are empirical distributions. Then the hypothesis of change H_1 is chosen if there is a $k \in [1, N]$ such that

$$S(\mathbf{y}^N) > h,$$

where the threshold h is set so as to guarantee a fixed false alarm probability. A localized version of the Kolmogorov-Smirnov test can be found in [DW77]. Recent works in the area of nonparametric fault detection include [Bha87] and [Car88].

In this dissertation, we will only look at the parametric case which is where the bulk of the research was done. As a result, it is more appropriate to describe the model class $\mathcal{M}$ as $\mathcal{M}(\theta)$, where the parameter θ is in a domain $D^k \subset \mathbb{R}^k$, k the dimension of the parametric model. Similarly, models M_0 and M_1 will be denoted by M_{θ_0} and M_{θ_1} respectively. In Sections 5.3.3, 5.3.7, and 5.3.8 we will discuss the issue of change-point detection and the complexity of the models used in situations of increasing complexity.

5.3.3 Change-Point Detection with Known Models

Among parametric change-point detection problems there are varying degrees of complexity. The simplest detection problem is realized when the models M_{θ_0} and M_{θ_1} are completely known in advance. This results from the fact that the only unknown is the change-point τ^*, thus limiting the selection to a finite set of possible models. Not surprisingly, all the work on change-point detection within the first decade of serious research in the area was carried out under this hypothesis. As unrealistic as this assumption may be, that research—as will be shown later when describing the work of Shirayev—laid many of the main foundations on how a proper change-point detection problem should be stated.

Let us show, through a simple example, how this simplified change-point problem is solved using one of the best known change-point detection methods: the likelihood ratio approach. Assume that the data y^N is a realization of an i.i.d. sequence of random variables with densities $f(\cdot, \theta_0)$ and $f(\cdot, \theta_1)$ before and after the change respectively. For instance, θ_0 and θ_1 might represent two different known means. Let us define the likelihood ratio between the hypothesis H_0 and H_1 at $1 \leq \tau \leq N$ by

$$\mathcal{L}_{\mathrm{H}_0/\mathrm{H}_1}(\tau) = \prod_{n=\tau}^{N} \frac{f(\mathrm{y}_n, \theta_1)}{f(\mathrm{y}_n, \theta_0)}.$$

Then, the likelihood ratio test, which provides an off-line solution, to the change-point detection problem is given by

$$\max_{\tau} \mathcal{L}_{\mathrm{H}_0/\mathrm{H}_1}(\tau) \geq h > 0, \tag{5.1}$$

for some properly set threshold h. If inequality (5.1) is satisfied for some $\tau \in [1, N]$, then the data is said to contain a change-point. Moreover, that particular value of τ represents the estimate of τ^*. This test enjoys some of the asymptotic optimality properties established by [Lor71] which will be described in Section 5.3.6.

Notice that even in this simple formulation the change-point detection problem is a multiple hypothesis testing problem. This is so since for each tentative τ we have a different hypothesis H_1. As a consequence, the change-point problem is a difficult problem to solve, since uniformly most powerful tests do not exist (see the discussion on page 106). Finally, for a multivariate normal mean likelihood ratio test version of the change-point problem, the reader is referred to [SW86].

5.3.4 On-Line Versus Off-Line Procedures

There are two general classes of change-point detection formulations: the off-line or a-posteriori change-point detection, and the on-line, sequential, or sometimes referred to as the quickest change-point detection problem. In the off-line formulation, which was presented in Section 5.3.1, a finite set of data is assumed to be given a-priori and the problem is to decide whether or not it contains a change-point. If it does then the scheme should estimate its time-location. Under this set up the generalized likelihood ratio approach is considered to be one of the most powerful methods of change-point detection (c.f., [Bas88]). Note that in the a-posteriori change-point detection problem it is possible to have multiple change-points in a given data set. Under this situation special care must be taken to solve the problem (c.f., [Mac74], and [Fed75]).

Note that the off-line set-up is in a sense simpler since one could in principle search for a very large number of, for example, possible parameters θ_1. Thus, the non-existence of time-constraints would avoid the need for more clever methods. Nevertheless, such searches could be computationally expensive up to the point of not been realistically implementable.

The on-line set-up has received a great deal of attention in the literature, the first piece of work dating back to [Pag54]. Under this set-up one does not have to worry about multiple change-points since the decision of change versus no-change has to be performed at the arrival of each new observation and its decision performed before the arrival of the next one. Another point that has to be taken into consideration is the very little time available between samples which excludes the possibility of intensive search procedures.

An example of an on-line implementation of the likelihood ratio test based on the statistic (5.1) is provided by the well-known and very much used cusum (cumulative sum) stopping rules, first introduced by [Pag54], of which a possible form of the stopping time is giving by

$$T = \min\left\{\tau > 0; S(\tau) = \sum_{n=1}^{\tau} \max_{1\leq\tau\leq N} \Lambda_{\mathrm{H}_0/\mathrm{H}_1}(\tau) \geq h\right\}, \tag{5.2}$$

where

$$\Lambda_{\mathrm{H}_0/\mathrm{H}_1}(\tau) = \log \mathcal{L}_{\mathrm{H}_0/\mathrm{H}_1}(\tau).$$

It is important to note that stopping rule (5.2) admits the recursive computation

$$S(\tau) = \Lambda_{\mathrm{H}_0/\mathrm{H}_1}(\tau) - \min_{1\leq n\leq\tau} \Lambda_{\mathrm{H}_0/\mathrm{H}_1}(n). \tag{5.3}$$

Another very important consideration when choosing among different detection rules is the efficiency associated with each statistic. For instance, the optimal test could be too computationally expensive. Thus a suboptimal rule might be preferred, especially when on-line implementations are needed.

In this thesis, we are particularly interested in providing on-line solutions to the change-point detection problem. However, the off-line formulation will be extensively used for the theoretical analysis of the chane-point scheme.

5.3.5 Bayesian Versus Non-Bayesian Formulations

Depending on how researchers treat the change-point time, we could find two formulations for the change-point detection problem: the Bayesian and the non-Bayesian set-up. In the former set-up, the change-point time is assumed to follow some a-priori given density (c.f., [Bat62], [Gar69], [Smi75], and [Shi78]). Shiryaev's work on Bayesian change-point detection will be briefly presented in the next section. A common probability model for the change-point τ is given by

$$P(\tau = 0) = p_0, \qquad P(\tau > n | \tau \geq 1) = \exp(-\lambda n), \tag{5.4}$$

for some known p_0 and λ constants.

In the non-Bayesian set-up, the change-point $\tau = \tau^*$ is assumed to be totally unknown. We would follow the non-Bayesian approach since in most applications it is not realistic and rather impossible to consider an a-priori distribution for the change-point time.

5.3.6 Change-Point Detection and Optimality

An important issue in change-point detection problems is to provide solutions which will exhibit some form of optimality. The first proper notion of optimality in an on-line framework is found in the pioneer work of Shirayev (c.f., [Shi61], [Shi63], and [Shi78]). Due to its importance and impact on the field of change-point detection we shall provide a short summary of the mentioned papers.

The articles address the on-line change-point detection problem for a sequence of i.i.d. random variables whose distributions are known before and after the change,

and where the change-point has an a-priori geometrical distribution. The aim is to find a stopping time which will consequently help solve the change-point problem in the quickest possible way.

Consider then the probability space $(\Omega, \mathcal{F}, \mathrm{P})$ and a filtration $\mathcal{F}_N, N \geq 0$. Define the stopping times T as the class of functionals Γ such that for each T we have $\{T < N\} \in \mathcal{F}_N \subset \mathcal{F}$. Let τ be a random variable defined in $(\Omega, \mathcal{F}, \mathrm{P})$ which represents the a-priori distribution of the change-point. In this framework we write $\tau^* = \tau(\omega)$. (The distribution of τ used by Shirayev was (5.4).)

Now define the following risk function

$$\rho(T) = \mathrm{P}(T < \tau) + c\,\mathbb{E}(T - \tau | T \geq \tau)\mathrm{P}(T \geq \tau), \qquad (5.5)$$

where $c > 0$ is some given constant.

Note that the risk function $\rho(T)$ is given in terms of two conflicting terms

$$T_d = \mathbb{E}(T - \tau^* | T \geq \tau^*) \qquad \text{and} \qquad R_f = \mathrm{P}(T < \tau^*).$$

These are known respectively as the probability of false alarms and the detection delay.

Due to the necessary tradeoff between T_d and R_f, Shiryaev solved the optimization problem among the subclass of stopping times $\Gamma_\alpha \subset \Gamma$ such that

$$\mathrm{P}(T < \tau^*) = \alpha,$$

that is, Γ_α is the set of stopping times with an a-priori fixed probability of false alarm. The optimal stopping rule, T^* can then be obtained by solving

$$T^* = \arg\min_{T \in \Gamma_\alpha} T_d.$$

Shiryaev then showed that the optimal detection rule for the case of the change in drift of a process (y) generated by

$$y_N = a\mathrm{I}_{\{N - \tau^*\}} + e_N, \qquad (5.6)$$

where a is a constant and (e_N) is a Gaussian process with independent increments, and $\mathrm{I}\{A\}$ the indicator function of the set A, is

$$T^* = \min\{N; P(\tau^* \leq N|\mathcal{F}_N) > (1-\alpha)(1-P(\tau^*=0))\},$$

which is the conditional probability distribution of the change-point time giving the observations up to the present time until they surpass the level $(1-\alpha)(1-P(\tau^*=0))$.

The definitions of T_d and R_f introduced by Shirayev are not the only possible ones. However others found in the literature differ only slightly. For example, when τ^* is totally unknown but fixed, the delay time is frequently defined as

$$T_d = \max_{\tau \geq 1} \mathbb{E}(T-\tau|T \geq \tau).$$

In the case of two fixed models M_{θ_0} and M_{θ_1} describing the dynamics before and after the change respectively, T_d and R_f represent, in a sense, the behavior of the change-point detection criterion with respect to the parametric values θ_0 and θ_1. Now, let us define

$$L(\theta) = \mathbb{E}T(\theta), \qquad \theta \in D_0,$$

where $L(\theta)$ is known in the literature as the average run length (ARL) function. The purpose of introducing $L(\theta)$ is to be able to cover other, in principle, possible values of the parametric models. It is easy to see that $L(\theta_0)$ and $L(\theta_1)$ are directly related to R_f and T_d. Therefore since $L(\theta)$ captures for θ_0 and θ_1 the properties of the change-point detection algorithm as defined by Shirayev, $L(\theta)$ generalizes those properties for all the possible $\theta \in D_0$. The function $L(\theta)$ is the direct counterpart of the power function in hypothesis testing (c.f., [Nik91]).

In an off-line set-up, the notion of optimality of a change-point problem is generally formulated in the context of classical hypothesis testing. Let $S(y^N)$ be a given statistic and h a level to be chosen optimally as follows: Under the constraint

$$P(S(y^N) > h|\mathrm{H}_0) \leq \alpha$$

known as the level (representing the probability of opting for a change when a change-point did not occur) maximize the power of the test, that is

$$P(S(y^N) > h|\mathrm{H}_1),$$

(the probability of deciding on a change when a change did occur). Note that since H_1 is actually a multiple hypothesis, a Uniformly Most Powerful (UMP) test is looked for. However as shown in [DP86], if the change-point is totally unknown then no UMP test exists in a non asymptotic framework even under known models before and after the change-point. Nevertheless, UMP tests can be recaptured in certain asymptotic formulations. For example, for fixed models M_0 and M_1, and under the assumption that

$$\lim_{n\to\infty} \tau^*/n = \gamma, \qquad 0 < \gamma < 1$$

the likelihood ratio test is found to be optimal for a level and a power of exponential type.

There seems to be a slight confusion in the literature about the meaning of optimality of a change-point detection procedure. Shirayev defined it as the detection rule or functional, among a general class of functionals, which minimizes a well-defined risk function. On the other hand, many researchers have employed the term optimality as the optimization of an a-priori given sufficient statistic or criteria (i.e., likelihood ratio) with respect to the design parameters of the problem. Clearly, Shirayev's approach is much more involved, and for complex change-point detection situations it might be almost impossible to implement. That is why the other notion of optimality has been more frequently used. However, researchers have failed to point-out this difference.

The optimality notion to be used in this thesis is based on minimizing a statistic characterized by the stochastic complexity of the data with respect to tentative models

over a threshold and a fixed gain related to the estimation of those models. Let us then state the general optimality definition for a change-point detection scheme in line with our future use. Let us mention that this is the most widely used definition.

Definition 5.3.1 An on-line change-point test or detection rule is called *optimal with respect to the statistic chosen to solve the problem*, if and only if for $R_f = \alpha$, α an a-priori given constant, the delay time T_d is minimized over all possible detection rules.

Other possible definitions of optimality are certainly used. For example one could fix a value for T_d and minimize R_f.

In an on-line framework [Lor71] proved the optimality of the Page-Hinkley's test for an i.i.d. sequence of random variables with known distribution before and after the change. More precisely, the smallest possible delay time T_d was established for the Page-Hinkley's test stopping time given in (5.3) when the rate of false alarms $R_f \to \infty$, and the threshold $h = R_d$. Also, the following interesting asymptotic relation was obtained in the mentioned paper

$$\lim_{R_f \to \infty} T_d = O(\log(R_f)/I(\theta)),$$

where $I(\theta)$ is Kullback's information measure.

The work of [DP86] provides the asymptotic distribution of the test statistic and the change-point time in a GLR framework.

The exact formulas relating R_f and T_d with the threshold h for the sequential detection of a change in mean for Bernoulli random walks and brownian motions were established by [Bas81]. For a brownian motion whose drift changes from $\mu_0 \neq 0$ to μ_1 and fixed dispersion σ,

$$R_f = \frac{1}{\mu_0}\left(\frac{\sigma^2}{2\mu_0}\left(\exp\left(2\frac{\mu_0}{\sigma^2}h\right) - 1\right) - h\right),$$

and we refer the reader to [Bas81] for the expression T_d since it is too cumbersome to be included here.

Note that the computation of the delay time T_d and the probability of false alarms R_d are closely linked to the distribution of the detection rule, or stopping time T. The exact or approximate computation of such distributions has been the prevailing focus of the mathematical statistical literature that deals with the change-point problem. Some of the first results in this direction were the asymptotic distributions of the change-point estimate and its associated Page-Hinkley test statistics (c.f., [Hin70]) for an i.i.d. sequence with a simple change in mean. Even for this simple case the asymptotic distribution is not given in an explicit form but in terms of double Laplace transforms.

One of the most complete works for exact distributions of test statistics for change-point changes is [JJS87]. The authors' results are valid for the detection of a change in an i.i.d. normally distributed sequence with known or unknown variance and whose constant mean could experience only a single change. The intricate nature of the distribution of the test statistics for this very simple case shows the difficulty of finding them in more involved situations.

5.3.7 Change-Point Detection with Unknown Model Parameters

When the parametric dimension k of the model class $\mathcal{M}(\theta)$ is known but the parametric-values θ_0 and θ_1 are not, the change-point detection problem becomes substantially more intricate. Note that in this case the likelihood ratio test, commonly known as the GLR (generalized likelihood ratio) is given by

$$\min_{\theta_0} \max_{\theta_1} \max_{\tau} \mathcal{L}_{\mathrm{H}_0/\mathrm{H}_1}(\tau) \geq h > 0. \tag{5.7}$$

Evidently, solving inequality (5.7) instead of (5.1) is a much more difficult task since the search is no longer finite. One possible simplification for this problem is obtained by using some of the classical statistical estimation methods, for example the least square or the maximum likelihood estimation schemes. In most change-point detection problems, θ_0 could be properly estimated in advance by one of these estimation methods, the reason being that the assumption of having sufficient data before the change-point, can in general be made. Therefore, for theoretical purposes, one can assume that θ_0 is given. As a result one does not usually have to consider the minimization in inequality (5.7). Moreover, let us mention that in some simple cases, like when using model (5.6), the maximization can be reduced to a single one (c.f., [Bas88]).

In view of the above observation, the actual challenge arises when θ_1 is not known, which is the most typical case found in applications. The difficulty in solving this problem stems from the fact that model and change-point estimation have to be performed concurrently. This represents one of the main challenges in change-point detection problems. One attempt at resolving this issue (c.f., [BEG81]) involves the assumption of a lower bound, say $\Delta\theta$, for the jump magnitude

$$\|\theta_0 - \theta_1\| \geq \Delta\theta \qquad \forall\, \theta_0, \theta_1 \in D^k$$

and designing the test under the worst-case scenario, that is, considering a jump of magnitude $\Delta\theta$. A related approach along those lines is the so-called local asymptotic method (e.g., [DP86]). In [DP86] the assumptions used is

$$\lim_{n\to\infty} \|\theta_0(n) - \theta_1(n)\| = 0$$

with $\tau^* \to \infty$. The authors claim that this assumption is useful when little is known about the models after the change but one is nonetheless interested in som sort of worst case optimize solution.

In what are called the local approaches to the change-point detection problem, the central idea is to construct the statistic for change-point detection from the dominant terms of the asymptotic expansion of the likelihood ratio. In many important situations, like Gaussian AR or ARMA models, the expansion is possible and one can then show that the random variable

$$\rho_\tau = \frac{\partial}{\partial\theta}\Lambda_{H_0/H_1}(\tau)$$

is asymptotically ($\tau \to \infty$) distributed, for small $\|\theta_1 - \theta_0\|$, according to the following laws

$$\begin{aligned} \mathcal{N}(0, I(\theta_0)) \qquad & \text{for} \quad \tau < \tau^*, \\ \mathcal{N}(I(\theta_1)(\theta_1 - \theta_0), I(\theta_1)) \qquad & \text{for} \quad \tau \geq \tau^* \end{aligned}$$

where $I(\theta)$ is the Fisher information matrix.

Another approach which deals with the case of unknown θ_1 is based on the assumption that this parameter could take only one value out of a finite set of a-priori given values, that is to say

$$\theta_1 = \{\theta_1^1, \theta_1^2, \ldots, \theta_1^m; m < \infty\}$$

Methods based on this assumptions can be accomplished as simple extension to previous approaches since filters could be run in parallel for each of the fixed θ_1 values. A similar type of solution can also be applied if working under the assumption that the model M_{θ_1} could only be among a finite number of completely defined models' structures and values, not necessarily sharing all the same dimension for example. All these fault detection problem formulations fall into what is known as MM (multiple model) approaches. These were extensively studied by [WJ76] among others. A successful implementation was reported by [WES+80] in which the MM method was applied to the detection of incidents on freeways.

Recently, some investigations were carried out in which the only knowledge assumed about θ_1 is that it is an unknown constant, i.e. corresponding to a time-invariant model (c.f., [Bas88] and [Nik91]). The method has been called the two-model approach. However, it is mainly based on heuristic arguments and unfortunately no theoretical backing has yet been provided. The method that we shall propose in Section 5.5 partially resembles the two-model approach. Thus, our proposed scheme could be viewed as a first step towards a solid foundation of the two-model scheme.

5.3.8 Change-Point Detection in Very Complex Situations

In some applications, the change-point τ^*, instead of being modeled by a jump, is better represented by a slowly time-variant change. An illustration could be given by the depth of sleep monitoring applied to patients undergoing surgery. It also finds application in equipment maintenance by providing fast detection of worn down components (c.f., [PFC89]). For this particular case the time variation is very slow so these types of faults are known, in the literature, as incipient faults.

A possible modeling strategy for this case is to use time variant parametric models $M_{\theta_1(n)}$, where n represents time. This formulation of the change-point problem has not, to the best of our knowledge, been previously investigated. In this thesis we shall propose a solution in which the parameters $\theta_1(n)$ are assumed to be unknown.

We now arrive at the case where the parametric dimension k of the chosen model class $\mathcal{M}(\theta)$ is unknown. This sort of formulation has never been tackled by researchers in the field, and in this dissertation we shall provide some promising directions for solving this problem. More precisely, we will look at the issue of underparameterization when detecting change-points.

The most difficult change-point detection situation is obtained when the model class $\mathcal{M}(\theta)$ is itself unknown. However, there can be no hope of solving the detection

problem unless a model class, containing models which could satisfactorily describe the main features of data after the change-point, is given a-priori. This is because, in general, change-point detection algorithms would have to be implemented on-line. This constitutes a fundamental constraint in the design of change-point detection algorithms since, as a consequence of an on-line implementation, there would not be enough time to make any extensive model searches. Nevertheless, to search among model classes of excessive complexity might be fruitless since the data set after a change-point is usually very small due to the frequent desire for promptness of detection. For this reason, model classes that are too complex would do, in general, poorer jobs than simple classes. Therefore, if at least some a-priori knowledge about the dynamics after the change-point is available, it seems reasonable to presume that a non-necessarily restrictive model class is given for the description of the data after the change-point. Therefore we will henceforth assume, that a model class will be given for the implementation of the detection algorithm.

5.4 Failure Detection for Dynamical Systems

In this section we shall mainly deal with the deterministic approaches that have been proposed in the literature to solve fault detection problems. Since the basically simple methods under this framework were dealt with in the introduction, we shall concentrate our efforts on the class of elaborate FDI methods which were introduced in the early 70's. The origin of this class of techniques was the dissertations of [Bea71] and [Jon73]. The main feature of theses methods, as opposed to the previously more prominent ones (see Section 5.2), is the use of an explicit mathematical model of the system or subsystems to be monitored. There are now a number of similar and improved methods based on those original dissertations and these will be described

in what follows.

The basic idea of those schemes is a simple one. Let us assume the availability of a "good" model M of, say, an SISO system P, and moreover let us assume that the system's input (u) is available.

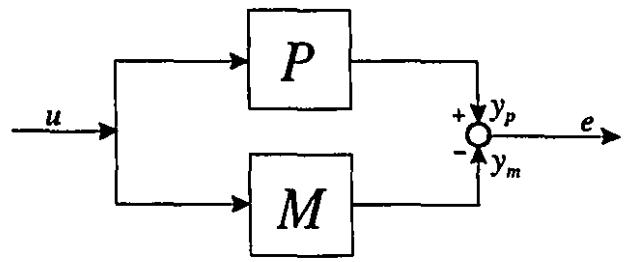


Figure 5.1: **Model based FDI method.**

Then the difference output model process (y_p) and the measured output signal (y_m) (see Figure 5.1) generates the residual process (e_n) which if properly designed should give an indication of a fault when it exceeds some given threshold. Based on this notion a variety of more elaborate and efficient methods were developed. These procedures are usually referred to as analytical redundancy methods, since they compare true measurements with artificial ones provided by the model. (Recall that earlier methods were based on comparing signals given by redundant physical components.)

The most well known methods of FDI for dynamical systems are: the parity space, the detection filter, and the Kalman filter based approach. We will now discuss the basic ideas of each one of them. We will try to take the most simple scenario to illustrate each approach since our goal is simply to convey the general idea of the methods, and not to give the full in depth design steps needed in more general situations. References will be provided that cover many of the more complex cases.

Before presenting the different FDI methods let us introduce some of the basic aspects involved in the steps taken when modeling failures in dynamical systems.

5.4.1 Modeling of Failures for Dynamical Systems

A very general and compact description of dynamic systems that are subjected to changes in time is found in the survey paper of Frank [Fra90]. Here, we will follow a similar formulation for the description of the models.

The class of stochastic linear dynamical models given by

$$x_{n+1} = Ax_n + Bu_n + Ew_n \tag{5.8}$$

$$y_n = Cx_n + Du_n + Fv_n. \tag{5.9}$$

provides a very general model representation for the normal operations of systems. As it is well known, A models the dynamics of the system, B and E the way the actuator signal (u_n) and system noise (w_n) enter the system respectively, C the way measurements of the system are taken, and D and F the way the reference input (u_n) and noise (v_n) affects the measurements (y_n) respectively (c.f., [Kai80]).

Since listing all of the model dimensions in (5.8)–(5.9) would be cumbersome and would not add to the understanding of the main ideas, let us just say that as usual uppercase letters denote matrices and lowercase letters denote vectors of appropriate dimension. Let us in general denote the i-th column of a matrix M by m^i, and the i-th component of a vector v by v^i.

Let faulty system configurations be described by the class of models given by

$$x_{n+1} = Ax_n + Bu_n + Ew_n + Kf_n \tag{5.10}$$

$$y_n = Cx_n + Du_n + Fv_n + Gg_n. \tag{5.11}$$

The normal and faulty model configurations described by (5.8–5.9) and (5.14–5.15) respectively can represent a wide variety of non-faulty and faulty situations. Let us illustrate this through some simple examples.

In a deterministic set-up the non-faulty model is frequently given by the state

space equations

$$x_{n+1} = Ax_n + Bu_n \tag{5.12}$$

$$y_n = Cx_n, \tag{5.13}$$

Typical faulty configurations are given as follows: For a model with actuator and sensor faults, we have

$$x_{n+1} = Ax_n + Bu_n + Kf_n \tag{5.14}$$

$$y_n = Cx_n + Gg_n. \tag{5.15}$$

For instance, if the matrix $K = b^i$ and f_n^i is a step function, we get a typical actuator bias failure; if $f_n^i = -u_n^i$ a the total breakdown of an actuator; if $G = c^i$ and g_n^i a step function we get a sensor bias; and if $g_n^i = -x_n^i$ a dead sensor. Model (5.14)–(5.15) can also incorporate changes in its free dynamics. For example, if $f_n = x_n$, the transition matrix equals $A + K$.

We have discussed only a few types of fault situations. However, with these simple examples, the reader should be able to model, in a similar fashion, numerous fault situations.

Finally let us add an example in a stochastic framework. For instance, a nominal model can be given by

$$x_{n+1} = Ax_n + Bu_n + w_n \tag{5.16}$$

$$y_n = Cx_n + v_n \tag{5.17}$$

where w_n and v_n are the process and sensor noise respectively. Some particular examples of faulty system operation can be given by

$$x_{n+1} = Ax_n + Bu_n + w_n + f_n \tag{5.18}$$

$$y_n = Cx_n + v_n + g_n \tag{5.19}$$

where f_n and g_n are two noise processes which give in this case increased process noise and added sensor noise, respectively.

In general, we can say that the term Ew_n is used to model unknown inputs to the open-loop system and to the actuators, whereas Kf_n is used to model faults in the plant and in the actuator dynamics. The term Fv_n basically models unknown inputs to the sensors, whereas Gg_n models faults in the sensors.

A terminology employed frequently in the literature is to refer to the fault terms like E, K, F, and G as signatures, and to the time functions w_n, f_n, v_n, and g_n as modes. Note that in general, the modes will be functions of $\mathrm{I}_{n>\tau}$ but we did not write it explicitly for brevity of notation.

In the design of fault detection algorithms, one is interested in maximizing the sensitivity of the detector to sensor malfunctions while assuring that remains insensitive to disturbances and noise (c.f., [PFC89]).

5.4.2 The Parity Space Approach

Let us assume that the true system to be monitored is represented, in its non-faulty operation, by the standard deterministic state space equations:

$$x_{n+1} = Ax_n + Bu_n, \qquad x_0 = 0 \tag{5.20}$$

$$y_n = Cx_n \tag{5.21}$$

From s-pairs of input-output data $\{(y_i, u_i),\ i \in [n-s, n]\}$ and by simply recursively solving equations (5.20) and (5.21) starting from time $n-s$, we can obtain the following input-output-state relations

$$\begin{bmatrix} y_{n-s} \\ \vdots \\ y_n \end{bmatrix} - M \begin{bmatrix} u_{n-s} \\ \vdots \\ u_n \end{bmatrix} = \mathcal{O}x(n-s), \tag{5.22}$$

where matrix M is a Hankel matrix formed with the first s Markov parameters of system (5.20)–(5.21)

$$M = \begin{bmatrix} 0 & & & \\ CB & 0 & & 0 \\ CAB & CB & 0 & \\ \vdots & \ddots & \ddots & \ddots \\ CA^{s-1}B & \dots & CAB & CB & 0 \end{bmatrix},$$

and the vector $\mathcal{O}$ is

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix},$$

which coincides with the observability matrix of system (5.20)–(5.21) if $s = \dim x_n$ (c.f., [Kai80]). Note that (5.23) can be used to check for consistency of the data set $\{(y_i, u_i),\ i \in [n-s, n]\}$ with respect to the mathematical model (5.20–5.21). This is so since under any deviation from the ideal situation—presence of disturbances, modeling errors, noise, failures, etc.—(5.23) will cease to hold. Note that in the ideal case any of the scalar equations, which are extracted from the dynamic equations and are contained in (5.23), will suffice to check for consistency. However, the redundant scalar equations in (5.23) can be exploited for FDI purposes when they deviate from the ideal situation. Let us see how:

Define the parity space of order s as

$$\mathcal{P} = \left\{v; v^T \cdot \mathcal{O} = 0\right\}.$$

Then one can define the residuals

$$r_n = v^T \cdot \left(\begin{bmatrix} y_{n-s} \\ \vdots \\ y_n \end{bmatrix} - M \begin{bmatrix} u_{n-s} \\ \vdots \\ u_n \end{bmatrix} \right), \tag{5.23}$$

and get the following situations:

$$r_n \begin{cases} = 0 & \text{when system is operating under (5.20)–(5.21);} \\ \neq 0 & \text{at the presence of a fault.} \end{cases}$$

Clearly, in a real situation, r_n should exceed a certain level before signaling an alarm. The space $\mathcal{P}$ can be then interpreted as the invariant unreachable manifold of vectors of input-output data generated by the left hand side of (5.23). The freedom thus obtained for choosing the vectors v can be used to try to make each signal (r_n) sensitive to a specific type of failure.

In practice, modeling errors, disturbances, and noise have to be taken into account. However, in general, there is not enough freedom to satisfy all of the numerous specifications. Thus some optimal solution based on the minimization of a certain appropriate cost function must be employed.

5.4.3 The Kalman Filter Based Approach

Let us assume that the true system to be monitored is now represented by

$$x_{n+1} = Ax_n + Bu_n + w_n \tag{5.24}$$

$$y_n = Cx_n + v_n \tag{5.25}$$

The stochastic processes w_n and v_n are second order stationary and uncorrelated with $\mathbb{E}\, w_n = \mathbb{E}\, v_n = 0$, $\mathbb{E}\, w_k w_j^T = R\delta_{kj}$, and $\mathbb{E}\, v_k v_j^T = S\delta_{kj}$, where δ_{kj} is the Kronecker delta operator.

The most simple FDI design, under this category of methods, is based on the construction of one Kalman filter. Let $\widehat{x}_{n|n-1} = (x_n|\mathbf{H}_y^{n-1})$, that is, the orthogonal projection of the state x_n onto the Hilbert space $\mathbf{H}_y^{n-1}$, which is the space generated by the random variables $y_0, \ldots, y_{n-1}$. Then the Kalman filter corresponding to the non-faulty model is given by

$$\widehat{x}_{n+1|n} = A\widehat{x}_{n|n-1} + K_k(y_k - C\widehat{x}_{n|n-1}), \qquad \widehat{x}_{0|-1} = 0, \tag{5.26}$$

where K_n is the Kalman filter gain calculated by solving the well-known Riccati equation (c.f., [Cai88]).

Now one can compute the difference between the measured output and the estimated output to generate the residual process

$$\epsilon_n = y_n - C\hat{x}_{n|n-1},$$

and use it as a basis for fault detection. Under the non-fault operation the residual corresponds to the innovation process which is Gaussian white noise with known variance. If a fault occurs these convenient properties will be lost, and thus statistical tests for a change in the probability distribution of the residuals can be used. For example, in some particular situations a change in mean test will suffice to detect a change from the normal operation (c.f., [MP71]).

A well-known approach, which is based on the use of a single Kalman filter, is to apply the GLR test to this particular case. (See for example the article of Willsky in [BB86].) It is also based on computing the innovations based on a Kalman filter designed under a non-faulty operation. Since different types of faults will be reflected in a different manner onto the innovations, the GLR actually computes the likelihood of events by calculating the correlations of the residuals with certain abrupt change signatures which are related to the dynamic profile of each change.

A drawback of the methods based on one Kalman filter is that parametric changes cannot be accounted for. In order to generalize these kinds of methods to more complex fault situations, a bank of Kalman filters is used.

Suppose that in the time interval $[1, N]$, an unknown number of change-points occurred, and that every change-point is represented by a jump to a model M_i belonging to a class of models $\mathcal{M} \triangleq \{\mathrm{M}_i; i \in [1, l], l < \infty\}$. A method has been proposed to solve this multiple change-point problem using a bank of Kalman filters. In order to account for any possible structural changes at each point in time, a growing bank of

Kalman filters are employed. That is, at each time n, l Kalman filters are started in parallel in order to assess each possible type of change. The method is based on recursively computing the conditional probability densities

$$p(x_n|y_1,\ldots,y_n;\mathrm{M}_i), \tag{5.27}$$

for each possible model $\mathrm{M}_i \in \mathcal{M}$. The densities (5.27) can be computed via the innovation processes of an exponentially growing bank of Kalman filters. Under the assumption that x_0, (w_n), and (v_n) are jointly Gaussian with $\mathbb{E}\,w_n = \mathbb{E}\,v_n = 0$, the Kalman filters give the conditional densities

$$\mathcal{N}(\widehat{x}_{n|n-1}, V_k), \qquad V_k = \mathbb{E}\,(x_k - \widehat{x}_{n|n-1})(x_k - \widehat{x}_{n|n-1})^T$$

of the state process (x_n) for each hypothetical model M_i (c.f., [Cai88]). It can be easily seen that the conditional densities given by the Kalman filters form the bulk of the computations of the conditional probabilities (5.27).

Under the assumption that only one change-point needs to be detected—meaning that in practice the change-points occur at ample respective distances in time—the bank of Kalman filters will grow linearly in time. This bank is related to all possible blended models constructed as follows: a non-faulty model for $n \in [0, k)$, and a faulty model for $n \in [k, N]$. Since the faulty model could be any of the possible l fault-models, one gets that the total number of possible blended models are $lN - l + 1$. Therefore, under the assumption of a single change-point, the method is simplified considerably.

One of the main drawbacks of the above procedures is that the classical Kalman filter has infinite memory, and thus will respond too slowly to any abrupt change in the dynamics of the system if it happens much beyond the time constant of the filter. Numerous modifications are possible to overcome this problem. For example we could mention the exponential age-weighting of data, and limited memory filters, some of

which fix the filter gain. However, to the best of our knowledge, all the techniques for improving the response of Kalman filters which are applied to the detection of abrupt changes, are ad-hoc, in that they are derived only from practical experience. Unfortunately, no theoretical analysis for the possible increase of FDI performance has been done.

Note that in all these methods Kalman filters are used to generate state estimates from incorrect models, models which might deviate considerably from the hypothetical model for which the Kalman filter was originally designed. No theoretical work has been performed to analyse possible filter instability and degraded performance. An informal discussion of this issue can be found in the paper of Willsky [BB86].

Another important point is that the Kalman filters are constructed on the basis of an exact knowledge of system models. Again no robustness analysis accounting for model uncertainty has yet been incorporated in this class of FDI methods.

5.4.4 The Detection Filter Approach

The detection filter approach has normally been carried out for a continuous-time LTI. In order to maintain the homogeneity of this survey we will present this method for discrete-time LTI systems.

Let us assume that the true system to be monitored is again given by (5.20–5.21). Then for FDI purposes we design the full-order observer

$$\widehat{x}_{n+1} = (A - HC)\widehat{x}_n + Bu_n, \qquad x_0 = 0 \tag{5.28}$$

$$\widehat{y}_n = C\widehat{x}_n. \tag{5.29}$$

Now, define the residuals as

$$r_n = y_n - C\widehat{x}_n.$$

Then the idea is to choose the matrix H such that, under the nominal model, the

residual r_n decays to zero, while under the faulty model, r_n increases in such a way as to reflect the fault in a unique and appreciable way.

Now, let us define the error process as

$$\epsilon_n = \widehat{x}_n - x_n.$$

Then combining (5.20) and (5.28)), we get the following difference equation

$$\epsilon_{n+1} = (A - HC)\epsilon_n - Lf_{n+1}. \tag{5.30}$$

It is well-known that the solution of (5.30) is given by

$$\epsilon_n = (A - HC)^n e_0 + L\sum_{k=1}^{n}(A - HC)^{n-i} f_{k+1}. \tag{5.31}$$

We see from (5.31) that the first requirement needed to be imposed on the matrix H is that $A - HC$ must be stable. This in turn will satisfy the first requirement for r_n, which is $\lim_{n=\infty} r_n \rightarrow 0$ under the nominal model.

Now assume that n is large enough so that the first term in (5.30) can be neglected, and moreover assume that only the i^{th} actuator fault is affecting the operation of the system. Then (5.30) can be simplified as

$$\epsilon_n = l_i \sum_{k=1}^{n}(A - HC)^{n-i} f_{k+1}^i \tag{5.32}$$

Now assume that one can find a matrix H such that the subspaces

$$S_i = C\ \mathcal{R}[l_i \quad l_i(A - HC) \quad \ldots \quad l_i(A - HC)^{n-1}],$$

are mutually independent. (Here $\mathcal{R}[D]$ denotes the range of the matrix D.)

If a matrix H satisfying the above two conditions exists, then the FDI problem is solved by monitoring the signals

$$\nu_n^i = (r_n | S_i),$$

where $(\cdot|S)$ denotes projection onto the subspace S, that is

$$\nu_n^i \begin{cases} = 0 & \text{when system is operating under (5.20)-(5.21) ;} \\ \neq 0 & \text{at the presence of a fault.} \end{cases}$$

In practical situations a threshold has to be used before setting an alarm signal.

Note that the effect of the fault on the residuals ν_n^i depends not on the time functions f_n^i but on the signatures l_i. Therefore this method has the advantage of only needing knowledge of the direction, that is the signature, in which the fault affects the system. We stress that this is the main advantage of this method as opposed to other FDI techniques.

The main drawback is that up to now no robustness analysis has been performed. Therefore the method requires precise modeling if satisfactory results are to be expected.

5.5 Predictive Stochastic Complexity Applied to Change-Point Detection for ARMA Systems

In this section, we shall present a change-point detection method for ARMA systems under the assumption that they have a slow and non-decaying drift after the change occurs. Also the abrupt jump parameter case, and change-point detection with undermodeling will be considered. The general detection scheme to be developed is inspired by the stochastic complexity theory. A salient feature is that the resulting change-point detection algorithm will ultimately be expressed in terms of fairly simple recursive equations. Some results on the analysis of the scheme are obtained, showing that the method is amenable to theoretical analysis. Moreover, simulations show that the approach exhibits surprisingly good detection capabilities. Some of these results can be found in [BG90], [GB91], and [BG92a].

5.5.1 The Mathematical Model

We shall now specify the modeling conditions to be imposed on the change-point detection problem, which is divided into two parts. The first part corresponds to the time before the change-point, where the system is considered to be time invariant. The second part is valid for the time after the change-point, where the system is considered time varying. In contrast to the time invariant system, we will see that the description of the time variant system is not a standard one.

We now describe the dynamics before the change-point. Let (y_n) for $0 \leq n < \tau^* < \infty$, be the output of an ARMA(p^*, q^*) system generated by the equation

$$A^* y = C^* e, \tag{5.33}$$

where (e_n) is the input process. The values of e_n and y_n for $n \leq 0$ are assumed to be 0. (Recall that τ^* is the actual location of the change-point.)

The time-invariant ARMA(p^*, q^*) system given by (5.33) satisfies Conditions 3.1.1 and 3.1.2.

The dynamics after the change-point is described by the time varying ARMA(p^*, q^*) system

$$(A_n^* y)_n = (C_n^* e)_n, \qquad \tau^* \leq n \leq N \leq \infty \tag{5.34}$$

The interpretation of the left hand side of (5.34) is as follows: the difference operator A_n^* acts on the process (y) and the evaluation is done at time n to get y_n. The right hand side is interpreted similarly.

We impose on the so-called "frozen time system" (frozen at time n) described by

$$A_n^* y = C_n^* e, \tag{5.35}$$

the following condition.

Condition 5.5.1 *The frozen time system given by (5.35) satisfies Condition 3.1.1 for each n, $\tau^* \leq n \leq N$.*

Let θ_n^* denote the $k = p+q$-dimensional vector composed of the coefficients of the polynomials A_n^* and C_n^*.

Condition 5.5.2 *We have* $\sup_{n \geq \tau^*} |\theta_{n+1}^* - \theta_n^*| = \dot{S} < \infty$, *where* $\dot{S}$ *is an upper bound for the rate of change of the time varying* $ARMA(p^*, q^*)$, *and where* $|\cdot|$ *denotes the Euclidean norm.*

We denote by $\mathcal{M}_{\tau^*}$ the model class described by appending in time the models given by (5.33) and (5.34).

We say that a system is slowly time varying if Condition 5.5.2 holds. (A more general definition and a theory of slowly time varying systems is given in [ZW91].) This definition of time varying systems is particularly useful for identification purposes. For example, it is well-known that if $\dot{S}$ is sufficiently small then the ARMA system described by (5.35) is stable (also inverse stable).

We would like to point out a certain drawback with the slowly time varying definition expressed by Condition 5.5.2. The drawback is that this definition does not seem to capture the "true rate of change" of the system in all cases. In the following three simple examples we will come across systems which are rapidly changing according to our definition but very slowly changing according to the definition given in [ZW91].

The first example is given by considering the case where $\theta_n^* = \theta^*$ for all n except for $n = \tau$, and where $\dot{S} = \theta_\tau^* - \theta^*$ is large. The next illustration is obtained by the time varying ARMA system described by the equation

$$y_n = a_n y_{n-1} + e_n, \tag{5.36}$$

where the parameter a_n typically alternates between only two values, say α_1 and α_2, and $\dot{S} = |\alpha_1 - \alpha_2|$ is large. The drawback of our slowly time variant definition is even more apparent in continuous-time, as we shall see by the final example which follows.

Suppose we have a continuous-time process modeled by a parametric model whose true parameter is given by

$$\theta_t^* = \theta^* + \varepsilon \sin(\omega t), \tag{5.37}$$

where $\varepsilon\omega$ is large but ε is small. Then $\dot{S} = \varepsilon\omega$ is large, but there is no reason to expect that the time variant estimation method will track θ_t^*.

The difficulty in defining the proper rate of change did not emerge in earlier change-point detection publications because it was generally assumed that the changes in the parameters were instantaneous, i.e, the jump parameter case. In spite of these shortcomings, our definition has the advantage that the standard identification procedures applied to this type of time varying ARMA model are theoretically tractable. A fairly complete analysis of this kind of time varying system is available in (c.f., [Ger89f]).

5.5.2 The Encoding Procedure

In this section we will encode the data process (y_n) by means of a predictive encoding procedure. This procedure makes extensive use of the off-line and on-line prediction error estimation methods introduced in Sections 3.1, and 3.2 respectively.

Let us describe the predictive encoding procedure. Let $\widehat{y}_n(\theta)$ denote the one-step ahead prediction of y_n, using $\theta \in D$ as the system parameter vector. It is easy to see that the prediction error is $\epsilon_n(\theta)$. Now, let τ, with $0 < \tau \leq N$, represent a possible location for the change-point. In order to get "good" prediction, it is clear that we should use the estimators $\widehat{\theta}^{\,0}_{n-1}$ for $0 \leq n < \tau$, and $\widehat{\theta}^{\,\lambda}_{n-1}$ for $\tau \leq n \leq N$, at time n. It then follows that the optimal predictive code length for y_n with respect to the model $\mathcal{M}_\tau$, τ fixed, with the off-line time invariant and small gain prediction error method is given by

$$\overline{C}(y_n, \tau) = \begin{cases} \left(\epsilon_n(\widehat{\theta}^{\,0}_{n-1})\right)^2, & \text{if } n < \tau; \\ \left(\epsilon_n(\widehat{\theta}^{\,\lambda}_{n-1})\right)^2, & \text{if } n \geq \tau. \end{cases} \tag{5.38}$$

Thus we associate to the observation y_n the code length $\overline{C}(y_n, \tau)$. Note that the prediction errors $\epsilon_n(\hat{\theta}^{0}_{n-1})$ and $\epsilon_n(\hat{\theta}^{\lambda}_{n-1})$ are, in the terminology of [Ris86], "honest", i.e. to predict y_n only data preceding the moment n is used.

A weak point about the codelengths $\overline{C}(y_n, \tau)$ is that they are obtained using a computationally intensive procedure. Namely, in order to obtain the prediction errors $\epsilon_n(\hat{\theta}^{0}_{n-1})$, and $\epsilon_n(\hat{\theta}^{\lambda}_{n-1})$ we have to start the computation at time 0 for each time n. Moreover, we also have to compute at time $n-1$ prediction error processes from time 0 until time $n-1$, for each iteration along the search for the estimators $\hat{\theta}^{0}_{n-1}$, and $\hat{\theta}^{\lambda}_{n-1}$. Since the ultimate objective is to obtain a change-point detection method that will be computable in real time, we need to modify the encoding procedure so as to make it on-line. It can be obtained by looking at recursive estimation methods. The off-line encoding procedure will nonetheless prove very useful in the analysis of the change-point detection method.

Let us now state the on-line predictive encoding procedure for the data process (y_n). As in the off-line case, the optimal predictive code length for y_n with respect to the model $\mathcal{M}_\tau$, τ fixed, with respect to the on-line time invariant and small gain prediction error method is given

$$C(y_n, \tau) = \begin{cases} (\epsilon^0_n)^2, & \text{if } n < \tau; \\ (\epsilon^\lambda_n)^2, & \text{if } n \geq \tau. \end{cases} \tag{5.39}$$

The codelengths for the process (y_n) could be computed by running two recursive prediction error algorithms in parallel for the whole time interval $[1, N]$.

5.5.3 Change-Point Detection as Model Selection

According to the previous section, for each possible τ, where $1 \leq \tau \leq N$, we have a model class $\mathcal{M}_\tau$ with the help of which the sequence (y_n) is encoded. Let us define

the associated total codelength for $\mathcal{M}_\tau$ by

$$S_N(\tau) = \sum_{n=1}^{N} C(y_n, \tau) = \sum_{n=1}^{\tau-1} (\epsilon_n^0)^2 + \sum_{n=\tau}^{N} (\epsilon_n^\lambda)^2.$$

The use of the word codelength is justified since in the Gaussian case, (i.e., when (e_n) is Gaussian white noise), $S_N(\tau)$ is identical to what Rissanen defined as predictive stochastic complexity.

Observe that $S_N(\tau)$ serves as a basis for comparison between different model classes, i.e. different $\mathcal{M}_\tau$'s. According to the stochastic complexity theory, the best model class description of the data process (y_n), is the one whose associated total codelength is minimum. This model class implicitly gives an estimate for the change-point, showing that the change-point detection method has been reduced to a model selection problem.

Set

$$m_N = \min_{1 \le \tau \le N} S_N(\tau),$$

then the estimator of τ^* is defined by

$$\widehat{\tau} = \left\{\tau; S_N(\tau) = m_N\right\}.$$

Let us denote the increments of $S_N(\tau)$ with respect to τ by

$$u_\tau = S_N(\tau + 1) - S_N(\tau).$$

It is straightforward to see that

$$u_\tau = (\epsilon_\tau^0)^2 - (\epsilon_\tau^\lambda)^2,$$

and hence the increments of $S_N(\tau)$ with respect to τ are independent of N. If we now rewrite $S_N(\tau)$ as

$$S_N(\tau) = S_N(1) + \sum_{k=1}^{\tau-1} (S_N(k+1) - S_N(k)) = S_N(1) + \sum_{k=1}^{\tau-1} u_k, \qquad (5.40)$$

then the minimization of $S_N(\tau)$ with respect to τ is equivalent to the minimization of

$$S'(\tau) = \sum_{k=1}^{\tau-1} u_k.$$

since $S_N(1)$ is constant. With this observation, a formal correspondence between cumulative sum methods and our stochastic complexity based method is established.

Now let N represent the present time and say that we wish to signal the presence of a change-point as quick as possible as data becomes available to us sequentially. It is important to observe that there is an intrinsic difference between finding the minimum of $S_N(\tau)$ when all the data sequence is given and signaling this minimum on-line. For the on-line alarm signal of the change-point we use the so-called Page-Hinkley test (c.f., [Hin70]). Let

$$m'_N = \min_{1 \le \tau \le N} S'(\tau),$$

and

$$d(N) = S'(N) - m'_N. \tag{5.41}$$

Then define the stopping time, or alarm time, as

$$T = \min\left\{N > 0; d(N) > h > 0\right\}, \tag{5.42}$$

where h is some constant level. The need to use a threshold h is, in a sense, what makes the off-line and the on-line change-point detection problems intrinsically different. Observe that at present time N, only two prediction errors have to be computed in order to know whether or not we have an alarm. Note that for

$$\hat{\hat{\tau}} = \max\left\{\tau; S'_T(\tau) = m'_T\right\}.$$

we have $\hat{\tau} = \hat{\hat{\tau}}$.

5.5.4 Analysis of the Change-Point Detection Method

In the pioneering work of [Shi63], the first statements about desirable properties of on-line change-point detection methods for i.i.d. random variables were given. Those were the minimization of the rate of false alarms and the delay time, i.e. the time elapsed between the change-point and the alarm signal. Since these are conflicting requirements, and thus cannot be minimized simultaneously, he proposed to fix the probability of false alarms and find, among a very general class of statistics, the one that minimizes the delay time. In our case the statistic for solving the change-point detection problem for ARMA systems is at first obtained without consideration for these requirements. What we do instead is to construct the change-point statistic based primarily on stochastic complexity ideas. Then we analyze the effect of parameters, such as the threshold h and the fixed gain λ, on the false alarm rate and the delay time. Although a complete analysis of the change-point detection method still needs further research, we nevertheless have some very encouraging results.

As a first step in the analysis we need to replace the recursive prediction error process (ϵ_n^λ) with its off-line version $\left(\epsilon_n(\widehat{\theta}_{n-1}^{\,\lambda})\right)$, and consider the off-line associated total codelength $\overline{S}_N(\tau) \triangleq \sum_{n=1}^{N} \overline{C}(y_n, \tau)$. Now, define the increments of $\overline{S}_N(\tau)$ with respect to τ by

$$\overline{u}_\tau = \overline{S}_N(\tau+1) - \overline{S}_N(\tau).$$

Then clearly $\overline{u}_\tau = \left(\epsilon_\tau(\widehat{\theta}_{\tau-1}^{\,0})\right)^2 - \left(\epsilon_\tau(\widehat{\theta}_{\tau-1}^{\,\lambda})\right)^2$, and hence the increments of $\overline{S}_N(\tau)$ are independent of N. Similarly to the on-line case, we get that the minimization of $\overline{S}_N(\tau)$ is equivalent to minimizing

$$\overline{S}'_N(\tau) = \sum_{k=1}^{\tau} \overline{u}_k.$$

Theorem 5.5.1 ([Ger91a]) *Under Conditions 3.1.1, 3.1.2, and under the assumption of no-change, i.e. $\tau < \tau^*$, for any $\lambda > 0$ the process $\overline{u}_\tau$ is L-mixing, and moreover*

$$\mathbb{E}\,\overline{u}_\tau = -\lambda\frac{p+q}{2} + O(\lambda^{3/2}) + O(c_0^\tau),$$

for all τ such that $\tau \leq \tau^*$, and with some $0 < c_0 < 1$.

Theorem 5.5.2 ([Ger91b]) *Under suitable conditions, and under the assumption of change, i.e. $\tau > \tau^*$, we have*

$$\left|(\epsilon_\tau^\lambda)^2 - e_\tau^2\right| \leq \delta_{1\tau} + \delta_{2\tau} + o(1), \tag{5.43}$$

where $(\delta_{1\tau})$ is L-mixing and such that $\delta_{1\tau} = O_M(\lambda^{1/2})$, and $(\delta_{2\tau})$ is a deterministic process such that $\delta_{2\tau} = O_M(\dot{S}/\lambda)$. Moreover, letting $\lambda = \dot{S}^{2/3}$ we get

$$\mathbb{E}\left|(\epsilon_\tau^\lambda)^2 - e_\tau^2\right| \leq O(\dot{S}^{1/3}) + o(1).$$

REMARK. With this choice of $\lambda = \dot{S}^{2/3}$ the order of magnitude of the upper bound of the tracking error in inequality (5.43) is minimized. We denote this choice of λ by λ_{opt}.

Theorem 5.5.3 *Under the conditions of Theorem 5.5.1, and the assumption that the change-point is a jump, i.e.*

$$\theta_\tau^* = \begin{cases} \theta_1^*, & \text{if } \tau < \tau^*; \\ \theta_2^*, & \text{if } \tau \geq \tau^*. \end{cases},$$

we have

$$\mathbb{E}\,u_\tau = (\theta_2 - \theta_1)^T W_{00}(\theta_2^*, \theta_2^*)(\theta_2 - \theta_1) + O(\lambda^{1/2}) + o(1).$$

PROOF. First note that

$$u_\tau = \left((\epsilon_\tau^0)^2 - e_\tau^2\right) - \left((\epsilon_\tau^\lambda)^2 - e_\tau^2\right).$$

A particular case of Theorem 5.5.2 is obtained by setting $\dot{S} = 0$, getting

$$\mathbb{E}\left((\epsilon_\tau^\lambda)^2 - e_\tau^2\right) = O(\lambda^{1/2}) + o(1). \tag{5.44}$$

Let us now compute $\mathbb{E}\left([\epsilon_\tau^0(\theta_2^*, \theta_1^*)]^2 - (e_\tau)^2\right)$. Based on [Ger88a] we can write

$$\begin{aligned}(\epsilon_\tau^0(\theta_2^*, \theta_1^*))^2 &= (\epsilon_\tau^0(\theta_2^*, \theta_2^*))^2 + 2\epsilon_{\theta\tau}^0(\theta_2^*, \theta_2^*)\epsilon_\tau^0(\theta_2^*, \theta_2^*)(\theta_2^* - \theta_1^*) + \\ &\quad (\theta_2^* - \theta_1^*)^T \left(\epsilon_{\theta\theta\tau}^0(\theta_2^*, \theta_2^*)e_\tau + \epsilon_{\theta\tau}^0(\theta_2^*, \theta_2^*)\left(\epsilon_{\theta\tau}^0(\theta_2^*, \theta_2^*)\right)^T\right)(\theta_2^* - \theta_1^*),\end{aligned}$$

to get

$$\begin{aligned}\mathbb{E}\left((\epsilon_\tau^0(\theta_2^*, \theta_1^*))^2 - (e_\tau)^2\right) &= (\theta_2^* - \theta_1^*)^T \mathbb{E}\,\epsilon_{\theta\tau}^0(\theta_2^*, \theta_2^*)\left(\epsilon_{\theta\tau}^0(\theta_2^*, \theta_2^*)\right)^T(\theta_2^* - \theta_1^*), \\ &= (\theta_2^* - \theta_1^*)^T W_{\theta\theta}(\theta_2^*, \theta_2^*)(\theta_2^* - \theta_1^*). \end{aligned} \tag{5.45}$$

Combining equations (5.44) and (5.45) we get

$$\mathbb{E}\,u_\tau = (\theta_2 - \theta_1)^T W_{\theta\theta}(\theta_2^*, \theta_2^*)(\theta_2 - \theta_1) + O(\lambda^{1/2}) + o(1).$$

which proves the claim. ▮

Using the previous theorems, the next corollary gives a rigorous theoretical justification for the change-point detection method in the case of an abrupt change-point.

Corollary 5.5.1 *If λ is small enough, and the effect of the "nonstationary initial conditions" are neglected, then*

(a) *Under the conditions of Theorem 5.5.1 we get*

$$\mathbb{E}(\overline{u}_\tau) = -\alpha_1 < 0; \qquad \tau < \tau^*. \tag{5.46}$$

(b) *Under the conditions of Theorem 5.5.3 we get*

$$\mathbb{E}(u_\tau) = \alpha_2 > 0; \qquad \tau > \tau^*. \tag{5.47}$$

Now let us consider the problem of false alarms. For this purpose we shall work under the assumption of no-change, i.e. $N < \tau^*$. Let us rewrite the stopping time T given by (5.42), in its off-line version and in a form more suitable to analysis. That is, let

$$\overline{T} = \min\left\{N > 0; \max_{1\leq m\leq N} \sum_{k=m}^{N} \overline{u}_k > h > 0\right\}.$$

Letting $\overline{U}_k = \overline{u}_k - \mathbb{E}(\overline{u}_k)$, we get $\overline{u}_k = \overline{U}_k - \alpha_1$. Now define

$$\overline{U}_N^*(\alpha_1) = \max_{1\leq m\leq N} \sum_{k=m}^{N} \overline{U}_k - \alpha_1. \tag{5.48}$$

Then, to arrive at an expression for the false alarm rate, we observe that the frequency of the event $\left\{\overline{U}_N^*(\alpha_1) > h\right\}$, is, say

$$F_1 = \frac{1}{N}\sum_{n=1}^{N} \mathrm{I}_{\left\{\overline{U}_N^*(\alpha_1)>h\right\}}$$

where I_B is the indicator function of the set B, and F_1 represents an upper bound for the frequency of false alarms. Since $\overline{U}_N^*(\alpha_1)$ is an L-mixing process in a restricted sense, we have by the law of large numbers

$$\overline{\lim}\frac{1}{N}\sum_{n=1}^{N} \mathrm{I}_{\left\{\overline{U}_N^*(\alpha_1)>h\right\}} \leq M_1\left(\mathrm{I}_{\left\{\overline{U}_N^*(\alpha_1)>h\right\}}\right) = P\left(\overline{U}_N^*(\alpha_1) > h\right)$$

The next theorem shows that an upper bound for the probability of the set $\left\{\overline{U}_N^*(\alpha_1) > h\right\}$ can actually be computed.

Theorem 5.5.4 *If $\overline{U}_N$ is a zero-mean L-mixing process such that $M_\infty\left(\overline{U}\right) < \infty$ and $\Gamma_\infty\left(\overline{U}\right) < \infty$, then setting $\beta = \alpha_1/\left(2M_\infty(\overline{U})\Gamma_\infty(\overline{U})\right)$, and $\gamma = \alpha_1\beta/2$ we have*

$$P\left(\overline{U}_N^*(\alpha_1) > h\right) < c_1 e^{-\beta h}$$

where $c_1 = e^{-\gamma}(1-e^{-\gamma})^{-1}$.

Remark: Theorem 5.5.4 shows that with h large enough, the rate of false alarms F_1 can be made as small as desired.

PROOF. (Theorem 5.5.4) Using (5.48) we get that

$$P\left(\overline{U}_N^*(\alpha_1) > h\right) \leq \sum_{m=1}^{N} P\left(\sum_{k=m}^{N}\left(\overline{U}_k - \alpha_1\right) > h\right). \tag{5.49}$$

Since $\left(\overline{U}_k\right)$ is a bounded, zero-mean L-mixing process we have by Theorem 2.4.5 with $u_k \equiv \overline{U}_k$ and $f_k \equiv \beta$, that

$$\mathbb{E}\exp\left(\beta\left(\sum_{k=m}^{N}\overline{U}_k\right) - \beta^2\kappa(N-m+1)\right) \leq 1$$

with $\kappa = 2M_\infty(\overline{U})\Gamma_\infty(\overline{U})$, for which

$$\mathbb{E}\exp\left(\beta\sum_{k=m}^{N}\left(\overline{U}_k - \alpha_1\right)\right) \leq e^{(\beta^2\kappa - \beta\alpha_1)(N-m+1)}. \tag{5.50}$$

Choosing $\beta = \alpha_1/2\kappa$ the right hand-side of (5.50) becomes $\exp\left(-\frac{\alpha^2}{4\kappa}(N-m+1)\right)$. Now for the m-th term in the left hand side of 5.49 we get

$$\begin{aligned} P\left(\sum_{k=m}^{N}\left(\overline{U}_k - \alpha_1\right) > h\right) &= P\left(\exp\left[\beta\sum_{k=m}^{N}\left(\overline{U}_k - \alpha_1\right) > \exp\beta h\right]\right) \\ &\leq \exp\left(-\frac{\alpha^2}{4\kappa}(N-m+1)\right)/e^{\beta h} \end{aligned}$$

by Markov's inequality. Let $\gamma = \alpha_1\beta/2$, then summation over m from 1 to N gives

$$\begin{aligned} P\left(\overline{U}_N^*(\alpha_1) > h\right) &\leq \sum_{m=1}^{N} e^{-\gamma(N-m+1)}/e^{\beta h} \\ &< e^{-\gamma}(1-e^{-\gamma})^{-1}e^{-\beta h} \end{aligned}$$

Setting $c_1 = e^{-\gamma}(1-e^{-\gamma})^{-1}$ we get the claim of the theorem. ∎

The present form of the analysis is thus far not very practical since the process $\left(\bar{\epsilon}_\tau(\hat{\theta}^{\lambda}_{\tau-1}, \theta^*)\right)^2$ is obtained through a computationally intensive procedure as was pointed out in the previous section. A similar deficiency was overcome in [Ger89a] by using a strong approximation result which relates off-line and on-line estimators. It is conjectured that a similar result holds for fixed gain estimators. For the time being, however, we must be satisfied with the above results.

The next aspect to be analyzed is the performance of our change-point detection method as measured by the so called detection delay. That is, the time elapsed between the change-point and the alarm time. More precisely, we would like to analyze the probability

$$F_2 = P\{T - \tau^* < \delta t > 0\}.$$

For this matter, we need to understand the nature of the stochastic process $T - \tau^*$, or equivalently the nature of the process $\sum_{k=\tau^*}^{T} u_k$. A first step in the right direction is provided by Theorem 5.5.2

What is actually left by the analysis is a lower bound for the tracking error $(\epsilon_N^{\lambda})^2 - e_N^2$ in terms of $\dot{S}$. As was illustrated by the various examples given previously, our definition for slowly time variant systems does not seem to capture the "true rate of change". It is therefore very difficult to obtain a lower bound for the tracking error. Nevertheless, Theorem 5.5.2 seems to be an important step towards obtaining an expression for the delay time.

5.5.5 Change-Point Detection and Undermodeling

One of the main goals of real-time change-point detection algorithms is to detect change-points as quickly as possible. Hence, these algorithms should use the minimum number of data samples after the change-point. As was shown in previous sections, change-point detection problems are particular types of model selection problems. This connection implies that the issue of quickest detection translates naturally to the question of how to choose "good" models for systems when only few data are available. It is intuitively clear that knowledge of only the parametric structure of a "good" model of a complex system might be of little help when only a short data sequence coming from this system is available. This is simply because of the fact that the effect of uncertainty about the parametric values of a complex structure might

be much more undesirable than the uncertainty due to the undermodeling of this system. Note that in change-point detection problems which occur in practice the parametric values of the system after the change (and sometimes even the structure) are unknown. This makes it reasonable to investigate whether undermodeling could improve the performance of change-point detection algorithms. To the very best of our knowledge, this issue has not been previously investigated. In the present section we shall give a partial theoretical justification that undermodeling could in principle improve the effectiveness of change-point detection algorithms. We will also support this claim via simulations in Section 5.6.

The recent results obtained in [Ger92a] provide us with important guidelines on how to theoretically tackle the issue of undermodeling in change-point detection. We shall consider here the specific problem of undermodeling of ARMA processes by using the simpler AR model structure. Thus, we shall explore the possibility of using AR models classes as the tentative descriptions of the data after the change-point, instead of the more complex ARMA models—even though the data is actually generated by an ARMA system.

Let (y) be the ARMA(p,q) process given by

$$A^* y = C^* e, \tag{5.51}$$

satisfying Conditions 3.1.1 and 3.1.2. Now consider AR(k) model classes which fit the data produced by (5.51). Let A^k denote any stable k-th order polynomial with constant term 1, and let the k-th order predictor error process be defined as

$$\epsilon^k(A^k) = A^k y.$$

Then the optimal k-th order predictor error process is given by

$$e^k = \mathring{A}^k y, \qquad \mathring{A} = \min_{A^k} \mathbb{E}\,|\epsilon^k(A^k)|^2,$$

and the effect of model uncertainty is quantified as the excess of mean-square prediction error

$$\rho(k) = \mathbb{E}\,(e_N^k)^2 - \mathbb{E}\,(e_N)^2.$$

Since $\mathring{A}^k$ is generally not known in practice, let us apply the time invariant recursive least square method to obtain the estimator sequence $(\hat{\mathring{A}}_N^{0,k})$ of $\mathring{A}^k$. Then the effect of parameter uncertainty is given by

$$m(A^k) = \mathbb{E}\,(\epsilon_N^{0,k})^2 - \mathbb{E}\,(e_N^k)^2, \qquad \epsilon_N^{0,k} = \left(\hat{\mathring{A}}_{N-1}^{0,k} y\right)_N.$$

Note that $\rho(k) + m(A^k) = \mathbb{E}\left((\epsilon_N^{0,k})^2 - (e_N)^2\right)$ is the excess of predictive stochastic complexity between the model class AR(k) and the ideal codelength that would be obtained if the parameters of the ARMA(p,q) model were known without uncertainty. In [Ger92a] the following result was established

$$\mathbb{E}\left((\epsilon_N^{0,k})^2 - (e_N)^2\right) = \left(\sigma^2 \frac{k}{N} + \rho(k)\right)(1 + o(1)) \tag{5.52}$$

under some suitable conditions on $k = k(N)$.

For change-point detection purposes, we would like to compare the difference of predictive stochastic complexities which are obtained with the help of AR(k) and ARMA(p,q) model classes. Let us then apply the time invariant PEM as described in Section 3.2 to get the prediction error

$$\left(\hat{\mathring{C}}_{N-1}^{0} \epsilon^{0,p+q}\right)_N = \left(\hat{\mathring{A}}_{N-1}^{0} y\right)_N.$$

Then in [Ger89d] the following result was derived:

$$\mathbb{E}\left((\epsilon_N^{0,p+q})^2 - (e_N)^2\right) = \left(\frac{\sigma^2}{N}(p+q)\right)(1 + o(1)). \tag{5.53}$$

Now combining (5.52) and (5.53) we get

$$\mathbb{E}\left((\epsilon_N^{0,p+q})^2 - (\epsilon_N^{0,k})^2\right) = \left(\frac{\sigma^2}{N}(p+q-k) - \rho(k)\right)(1 + o(1)). \tag{5.54}$$

This says that in principle it is possible to achieve a lower predictive stochastic complexity with an AR(k) model class than with the original ARMA(p, q) model class for small sample sets.

It is conjectured that a similar result also holds if we compute the predictive stochastic complexities relative to the recursive algorithms with small gain λ. Namely, that

$$\mathbb{E}\left((\epsilon_N^{\lambda,p+q})^2 - (\epsilon_N^{\lambda,k})^2\right) = \left(\sigma^2\frac{\lambda}{2}(p+q-k) - \rho(k)\right)(1+O(\lambda^{1/2})). \tag{5.55}$$

Therefore, for example, if $p+q > k$ and the contribution of the model uncertainty $\rho(k)$ is small enough so that $\sigma^2\frac{\lambda}{2}(p+q-k) - \rho(k) > 0$, then the encoding using the AR(k) model class gives shorter codelength than the one obtained if we instead use an ARMA(p, q) model class. In this case, we should then expect a decrease in the probability of false alarms.

In the coming simulation we shall show how the change-point detection performance is actually improved using the just described undermodeling ideas.

5.6 Change-Point Detection Simulations

What follows are a sequence of simulations to illustrate the change-point detection problem by means of the stochastic complexity approach which was presented in previous sections. In Section 5.6.1 an example of the detection of a slowly time variant change-point is presented, followed in Section 5.6.2 by illustrations of the effect that the fixed gain λ has on the performance of the change-point detection algorithm. In Section 5.6.4 the issue of undermodeling in change-point detection will be investigated. Lastly, in Section 5.6.5, the stochastic complexity based change-point

detection method will be compared to what could be thought of as a naive change-point detection procedure obtained by simply monitoring the time variant parameter estimates.

5.6.1 Slowly Time Variant Change-Point Simulation

Let data $\mathbf{y}^N$ be generated by a computer program that simulates a time invariant ARMA(p^*, q^*) system until a chosen change-point τ^*, and a slowly time varying ARMA(p^*, q^*) system after and including time τ^*. More precisely, we simulate a model in the model class $\mathcal{M}_{\tau^*}$. (This model class is described in Section 5.5.1.) Only a computer realization $\mathbf{y}^N$ of the process y^N generated by $\mathcal{M}_{\tau^*}$, plus the orders p^* and q^* of the ARMA systems, are assumed to be given a-priori to the user for the implementation of the change-point detection algorithm. In Section 5.6.4 we will illustrate that the correct knowledge of the model orders after the change-point is not crucial and that undermodeling actually can improve the performance of the change-point detection scheme.

The present simulation is run until $N = N_f = 1000$, and the change-point is chosen at $N = \tau^* = 500$. The input process (e) is Gaussian white noise with mean 0 and variance 1. Let $p^* = 2$ and $q^* = 1$, and consider the time invariant ARMA(2,1) system as described in (3.15). Then (3.15) generates the process (y) until the change-point τ^*.

From $N = \tau^*$ until $N = N_f$, the process (y) is generated by the slowly time varying ARMA(2,1) system described in (3.17). Note that the poles of this ARMA system move linearly from an initial location $.35 \pm .82i$ at $N = \tau^*$ to a final location $.35 \pm .28i$ at $N = N_f$, as illustrated in Figure 5.2.

In Figure 5.3 the realization of the process (y) used in the simulation is shown. Note that the change in the dynamics of the data process $\mathbf{y}^N$ is hardly noticeable.

As the data $\mathbf{y}^N$ becomes available, we run two recursive prediction error algorithms

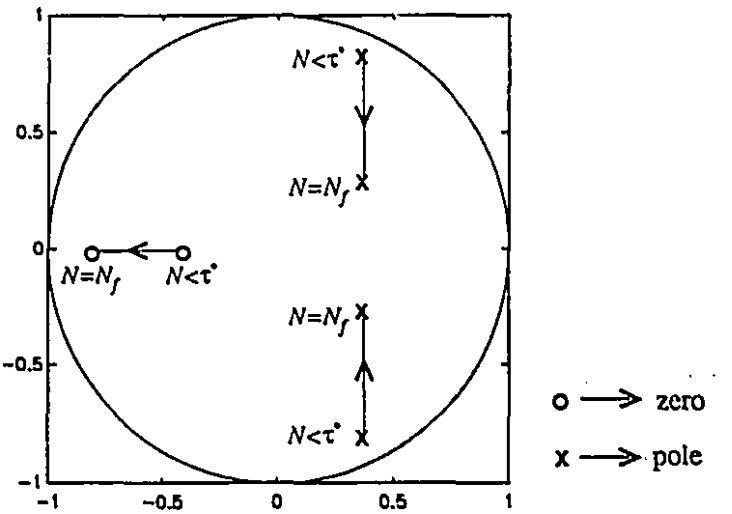


Figure 5.2: The time history of the pole-zero locations of the slowly time variant model $\mathcal{M}_{\tau^*}$.

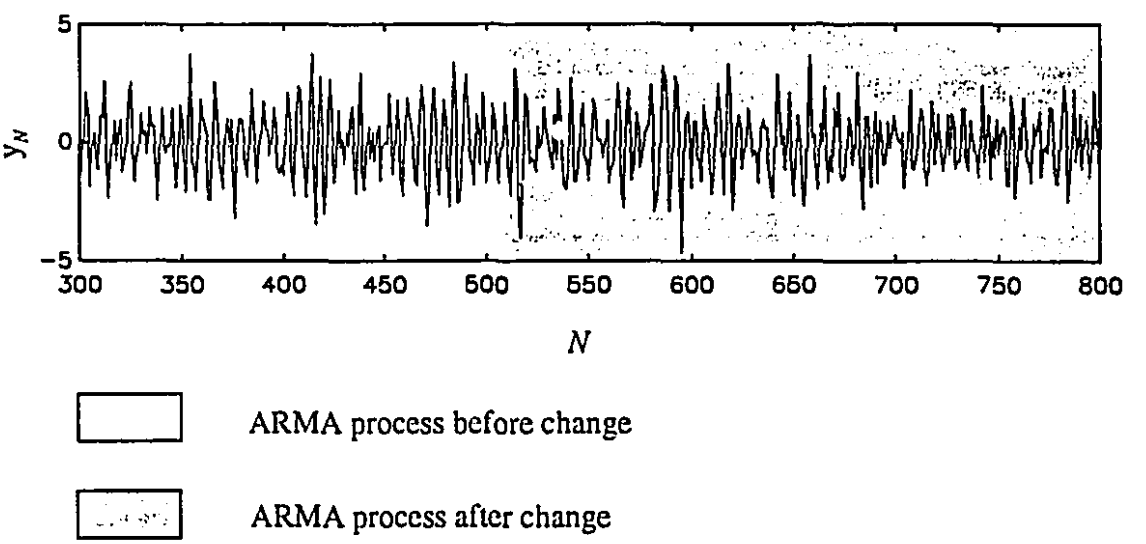


Figure 5.3: The data process $\mathbf{y}^N$ generated by model $\mathcal{M}_{\tau^*}$ which has a slowly time variant change-point at $\tau^* = 500$.

in parallel for the on-line computation of the prediction errors ϵ_N^0, and ϵ_N^λ. The fixed gain λ of the time varying prediction error algorithm is set to its $\lambda = \lambda_{opt}$ value given by $\lambda_{opt} = \dot{S}^{2/3} = .0113$. (Refer to the remark on Theorem 5.5.2.) The parameter estimates $\widehat{a}_{1,N}^\lambda$, $\widehat{a}_{2,N}^\lambda$, and $\widehat{c}_{1,N}^\lambda$ of $a_{1,N}^*$, $a_{2,N}^*$, and $c_{1,N}^*$ are shown in Figures 3.2-3.4, respectively.

The computations ϵ_N^0, and ϵ_N^λ allow us to compute the detector $d(N)$ given in (5.41). (See Figure 5.4b.) This detector is used to turn an alarm on when it exceeds some given threshold h. The prediction errors also allow us to compute the stochastic complexity $\mathcal{M}_\tau$. $S_{N_f}(\tau)$ for each model class $\mathcal{M}_\tau = 1, \ldots, 1000$ (as defined in (5.40)) once all the observations y^N have been obtained, that is when $N = N_f$. (See Figure 5.4a.)

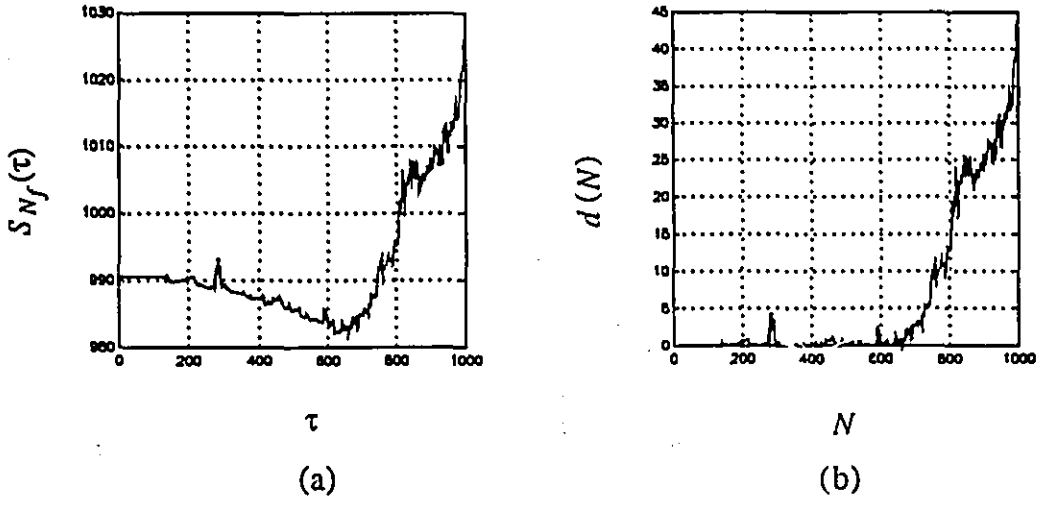


Figure 5.4: a) Predictive stochastic complexity with respect to model classes $\mathcal{M}_\tau$; b) The on-line detector.

The off-line estimate of the change point τ^* is $\widehat{\tau} = 628$, which corresponds to the value of τ which minimizes $S_{N_f}(\tau)$ in the interval $(0, N_f]$. The estimation error of the change-point is then $\widehat{\tau} - \tau^* = 128$.

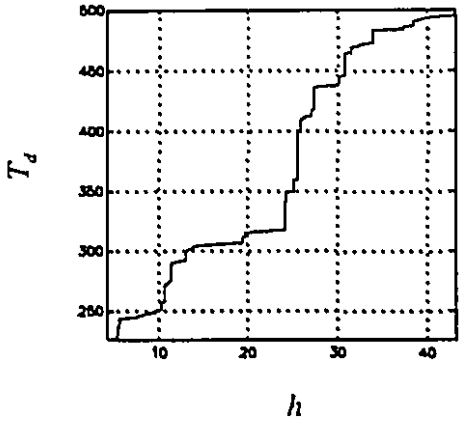


Figure 5.5: **The detection delay $T_d = T - \tau^*$ as a function of the threshold h.**

The behaviour of the detection delay T_d with respect to the threshold h is shown in Figure 5.5. The values of h range from its minimum value which allows no false alarms to occur in $[0, \tau^*)$, to its maximum value which acheives detection in the time interval $[\tau^*, N_f]$.

We shall see that the delay time for the abrupt change-point case to be presented in Section 5.6.3 is much smaller than the delay time of the slowly time variant change-point just described for fixed h. This shows that the delay time, $T_d = T - \tau^*$, is greatly affected by the rate of change of the system $\dot{S}$. That is, we should expect to obtain smaller T_d with greater $\dot{S}$. Therefore, the delay time and also the off-line change-point detection estimate must be viewed relative to the change in magnitude of the parameter vector in the time interval $[\tau^*, T]$. (Refer to Figure 5.22 of Section 5.6.5.)

5.6.2 Performance of the Change-Point Detection Algorithm with Respect to the Fixed Gain λ

We shall now repeat the simulation of Section 5.6.1 for different values of the fixed gain λ. The purpose is to study the effect fixed gain on change-point detection performance. Since we plan to use values of λ as small as $\lambda_{opt}/10$, we need to have a larger value for the final time N_f. This is so since for $\lambda_{opt}/10$ the time invariant and time variant recursive prediction error algorithms described in (3.13–3.14) will coincide until time $N = 10/\lambda_{opt} = 886$. Thus, both algorithms would produce the same prediction errors until this particular time N. Therefore let us set $N_f = 4000$ and $\tau^* = 3500$, with the rest of the parameters remaining at their previously assigned values. In Figure 5.6 the realization of the process (y) is presented.

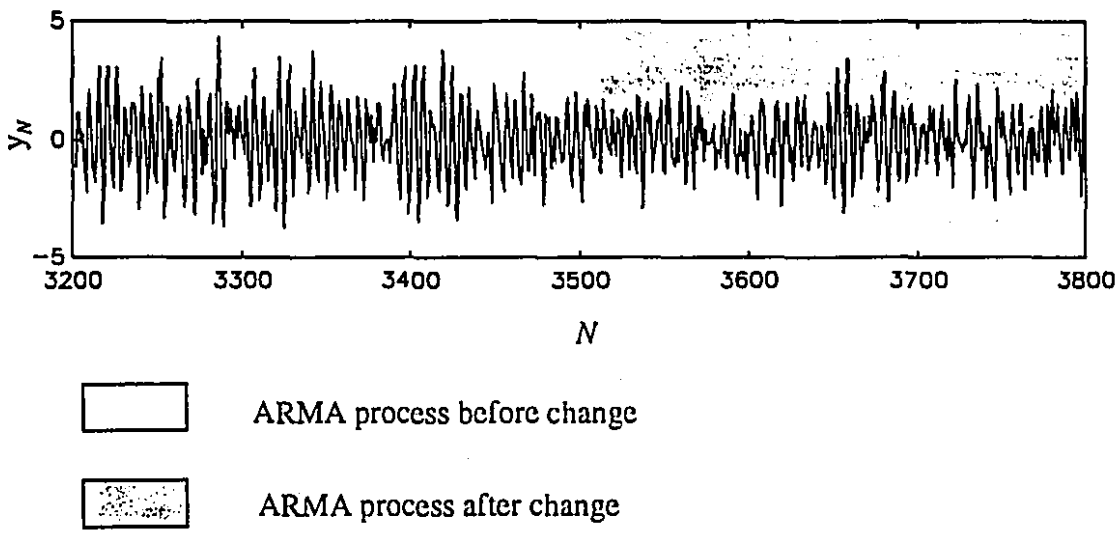


Figure 5.6: The data process y^N generated by model $\mathcal{M}_{\tau^*}$ which has a slowly time variant change-point at $\tau^* = 3500$.

Figure 5.7 shows the behaviour of the stochastic complexities and the detector $d(N)$. The change-point estimate is $\hat{\tau} = 3567$, and the estimation error $\hat{\tau} - \tau^* = 67$.

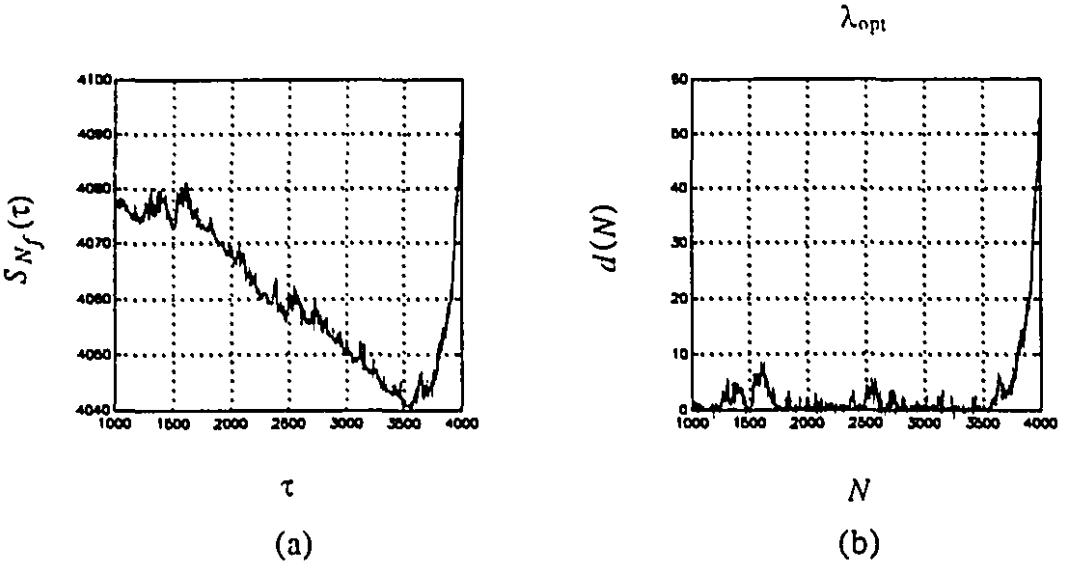
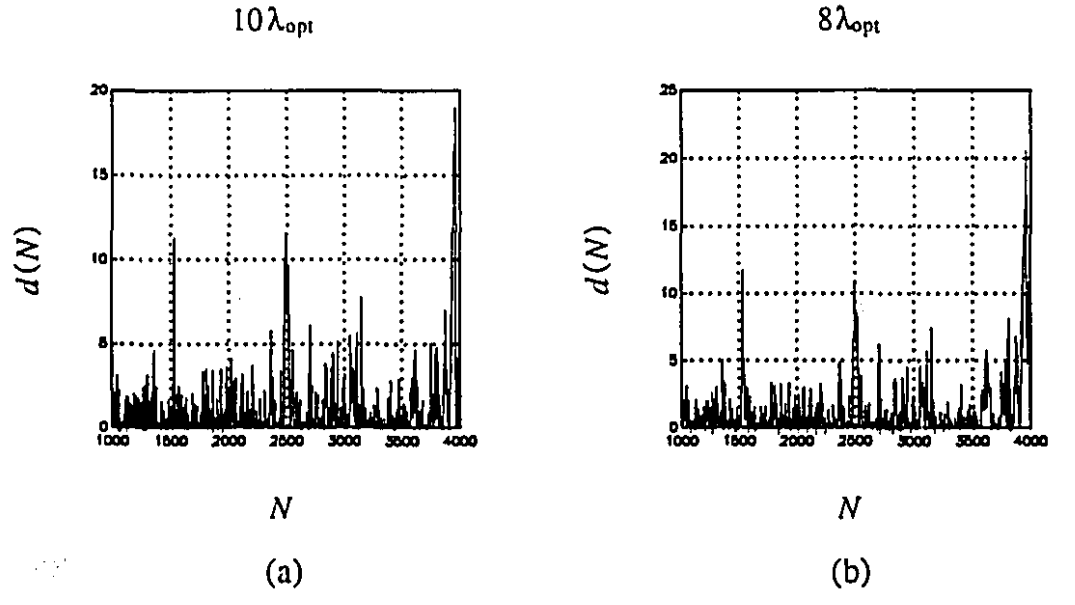
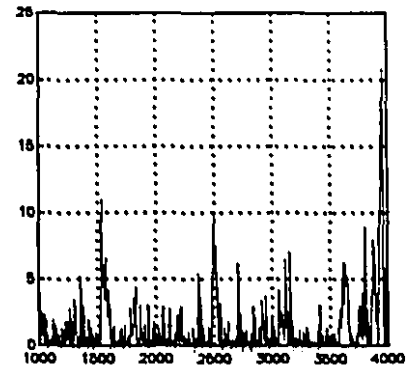


Figure 5.7: a) Predictive stochastic complexity with respect to model classes $\mathcal{M}_\tau$ b) The on-line detector.

In Figure 5.8 we display the detector $d(N)$ for values of the fixed gain λ ranging from $10\lambda_{opt}$ to $\lambda_{opt}/10$.

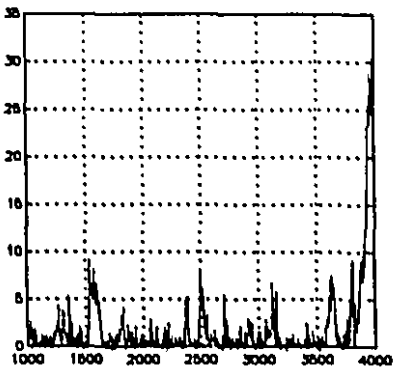


$6\lambda_{opt}$



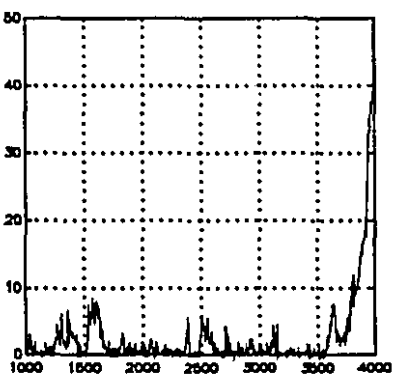
(c)

$4\lambda_{opt}$



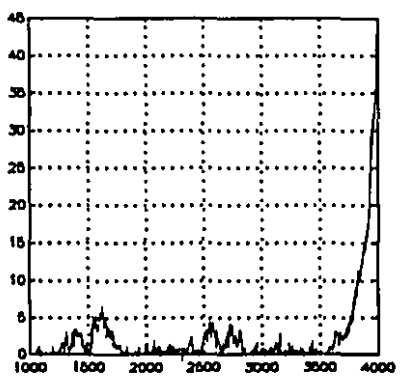
(d)

$2\lambda_{opt}$



(e)

$\frac{1}{2}\lambda_{opt}$



(f)

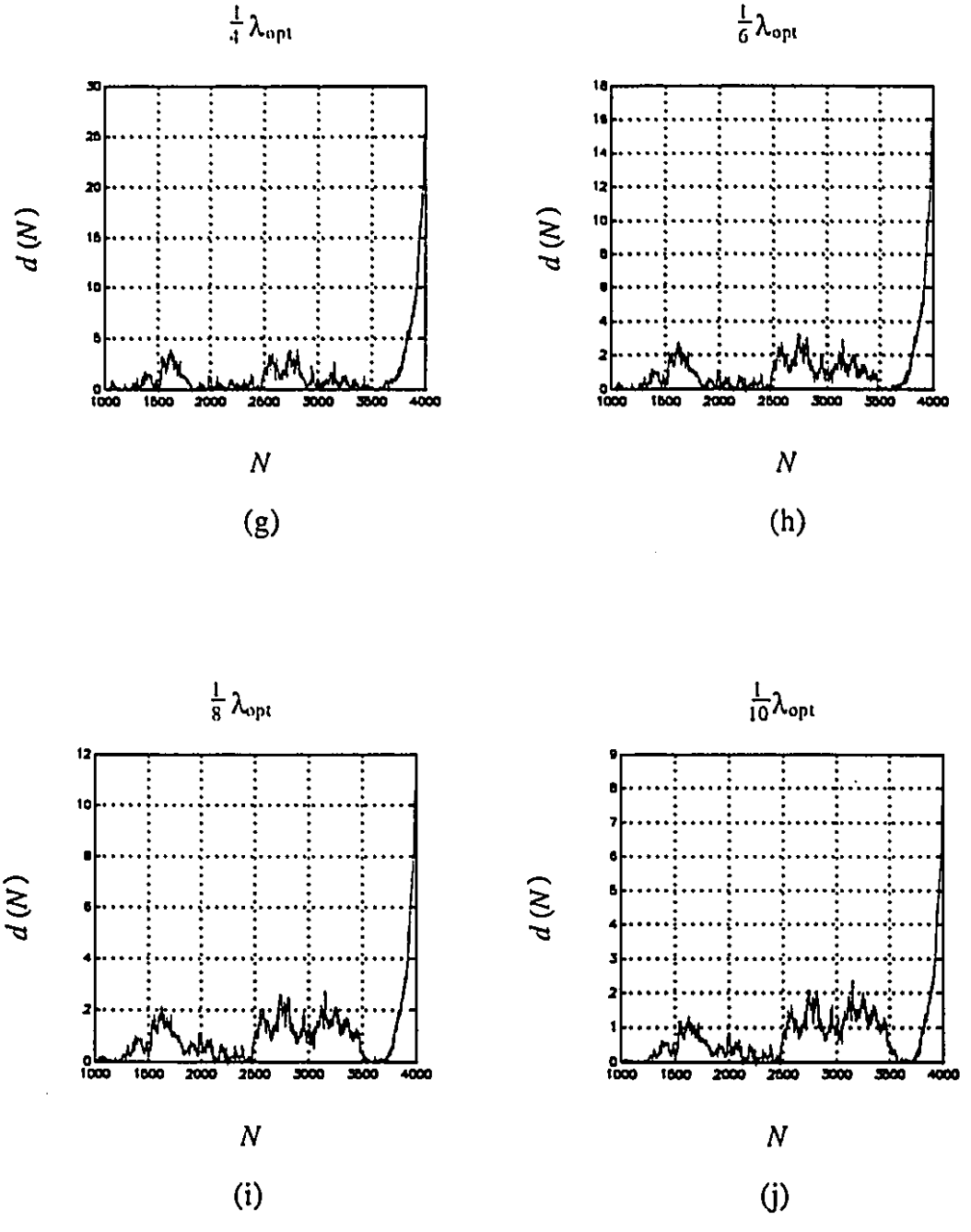


Figure 5.8: The on-line detector $d(N)$ with different values of the fixed gain λ.

In order to make the comparison of the behaviour of the detectors $d(N)$ for different values of λ more clear, we plot them together after the change-point τ^* in Figure 5.9 and Figure 5.10.

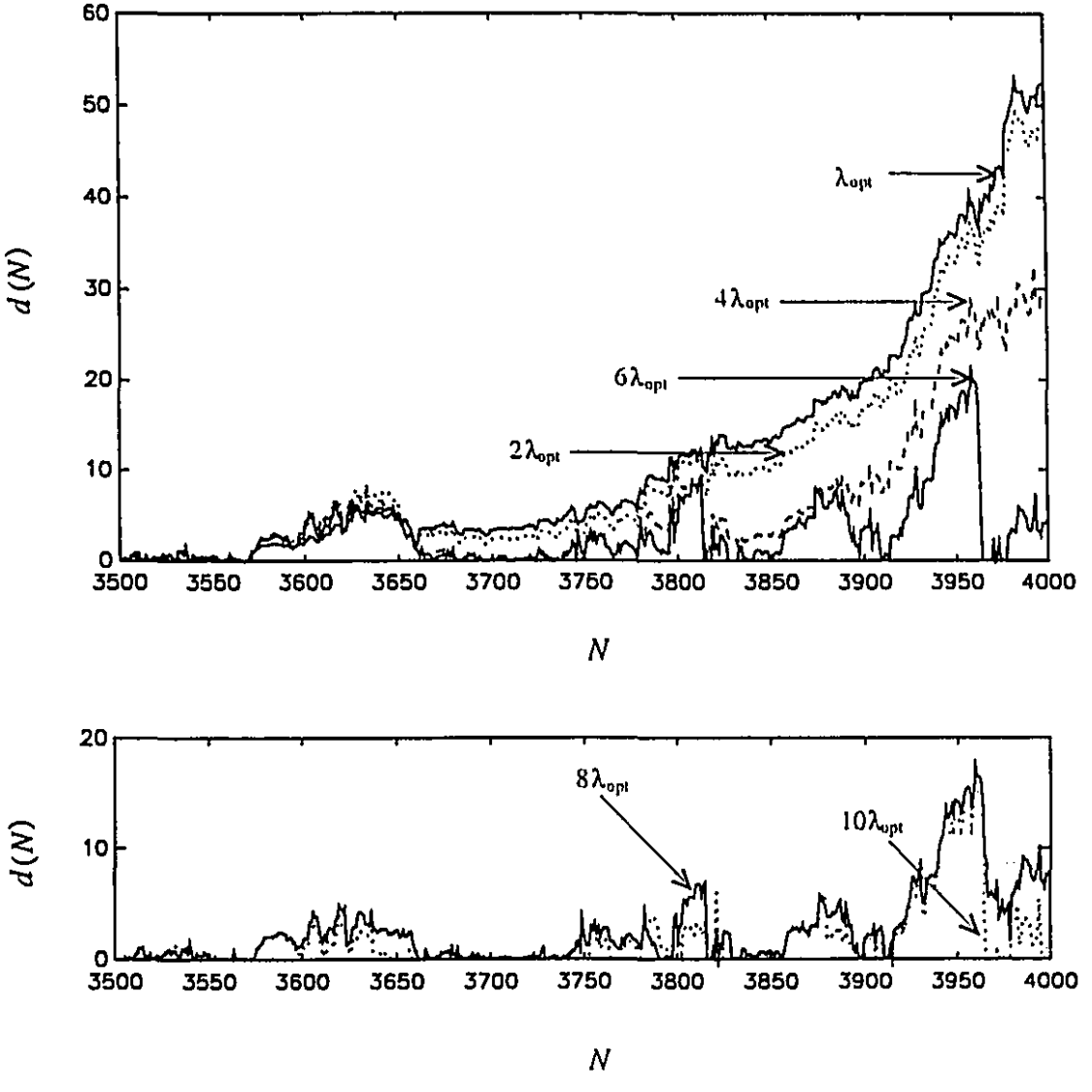


Figure 5.9: The on-line detector $d(N)$ for decreasing values of the fixed gain λ.

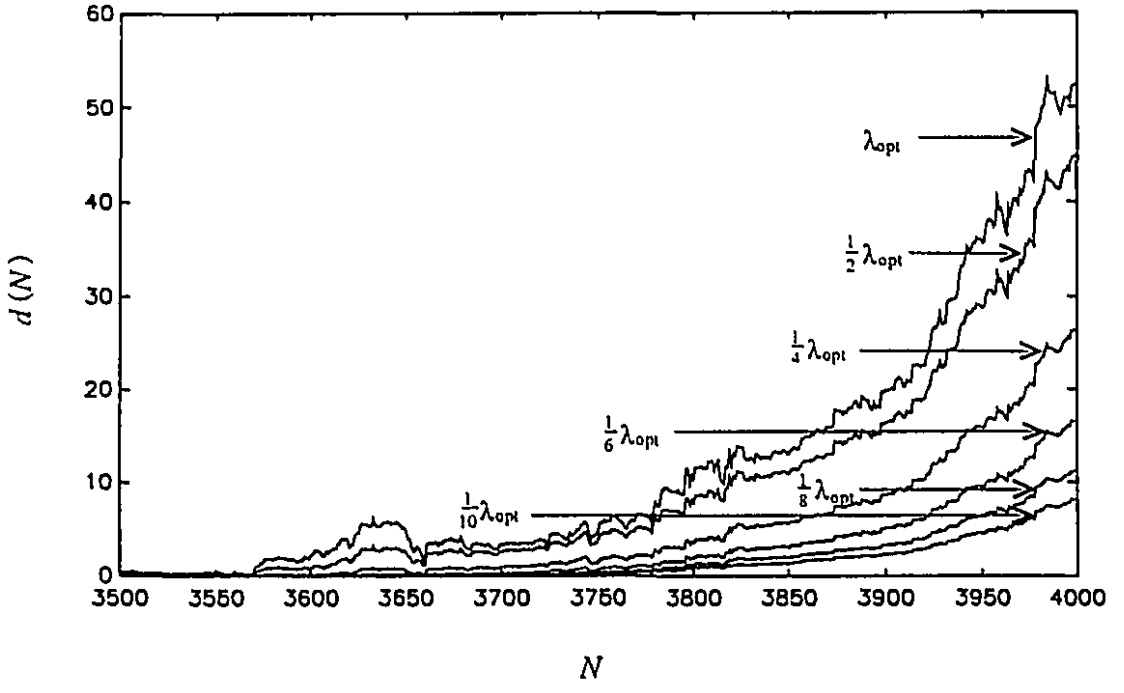


Figure 5.10: **The on-line detector $d(N)$ for increasing values of the fixed gain λ.**

In Figures 5.11 and 5.12 the detection delay T_d is plotted with respect to the threshold h for the different values of λ that are being considered. This is done so as to best appreciate the varied performance of the change-point detection method with respect to λ. Note that the best performance is obtained for $\lambda = \lambda_{\text{opt}}$, which shows that the a-priori chosen value of λ was indeed a good choice. The beginning of each of these plots marks the minimum threshold h under which no false alarms are obtained for the particular realization y^N of y^N that we are looking at. Hence, for values of h less than this minimum the change-point is not detected since for these values of h we have $T < \tau^*$.

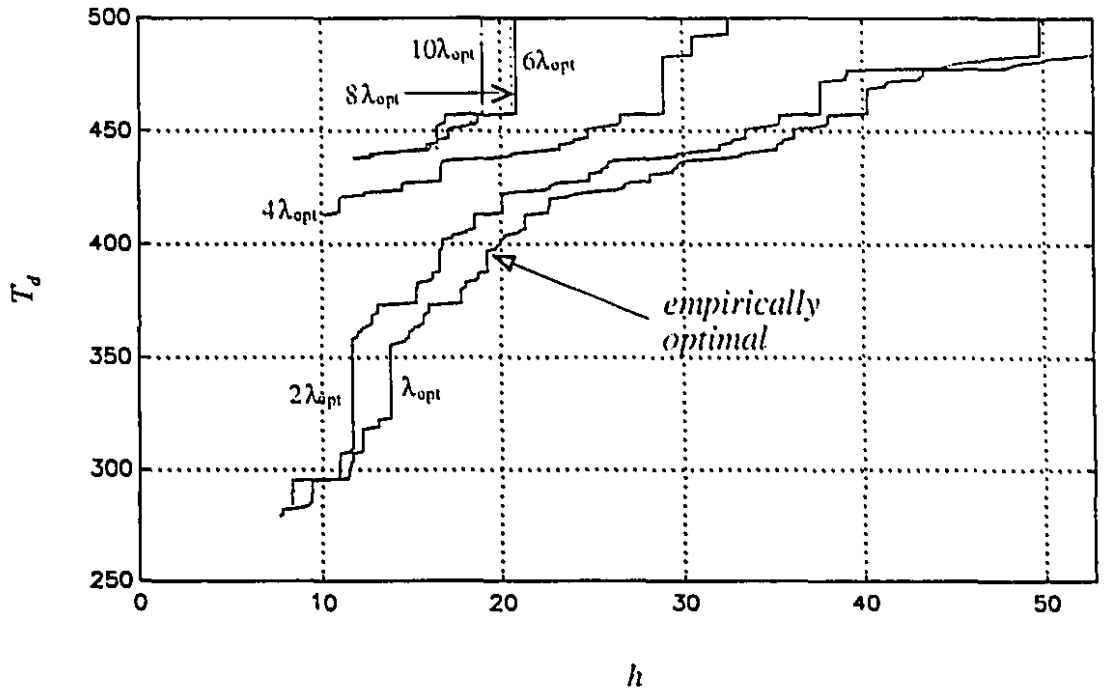


Figure 5.11: The detection delay $T_d = T - \tau^*$ as a function of the threshold h for decreasing values of the fixed gain λ.

Another observation that can be drawn from Figures 5.11 and 5.12 is that the delay time T_d is not drastically affected when values of λ which differ from λ_{opt} by about 100% are used. Moreover, for the values of λ being considered which differ by more than 100% from λ_{opt}, the change-point detection algorithm still provides reasonable performance with the exception of $\lambda < \lambda_{opt}/4$. This illustrates the robustness of the change-point detection method with respect to the fixed gain λ.

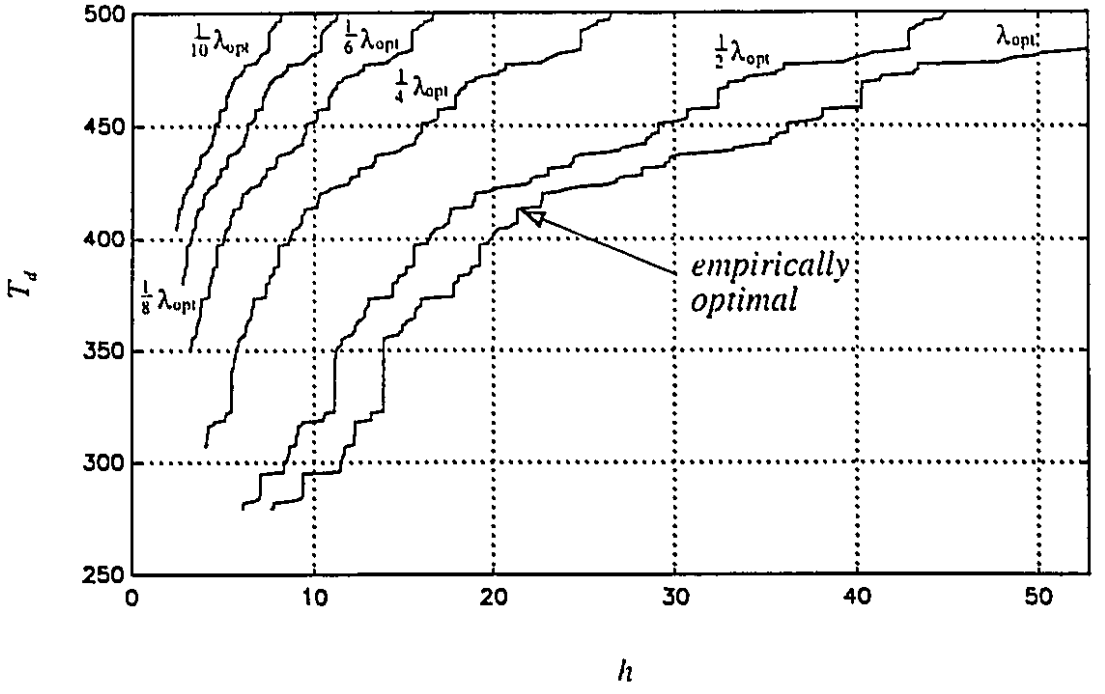


Figure 5.12: The detection delay $T_d = T - \tau^*$ as a function of the threshold h for increasing values of the fixed gain λ.

5.6.3 Jump Change-Point Detection Simulation

The simulation in this section will differ from the one introduced in Section 5.6.1 in that the change-point, instead of being slowly time variant will be a jump. Therefore the poles and the zero of the ARMA(2,1) system will jump from their initial condition to their final condition as exhibited in Figure 5.13. The only other difference is that the gain λ is set to $\lambda = .02$. The parameter estimates $\widehat{\widehat{a}}^{\lambda}_{1,N}$, $\widehat{\widehat{a}}^{\lambda}_{2,N}$, and $\widehat{\widehat{c}}^{\lambda}_{1,N}$ of $a^*_{1,N}$, $a^*_{2,N}$, and $c^*_{1,N}$ are shown in Figures 3.5–3.7, respectively.

In Figure 5.14 the realization of the process (y) with the abrupt change-point is presented. Again note that the change in the dynamics of the data process y^N is hardly noticeable.

Figure 5.15 shows the behaviour of the predictive stochasitic complexities $S_{N_j}(\tau)$ and the detector $d(N)$. The off-line estimate of the change point τ^* is $\widehat{\tau} = 503$, and

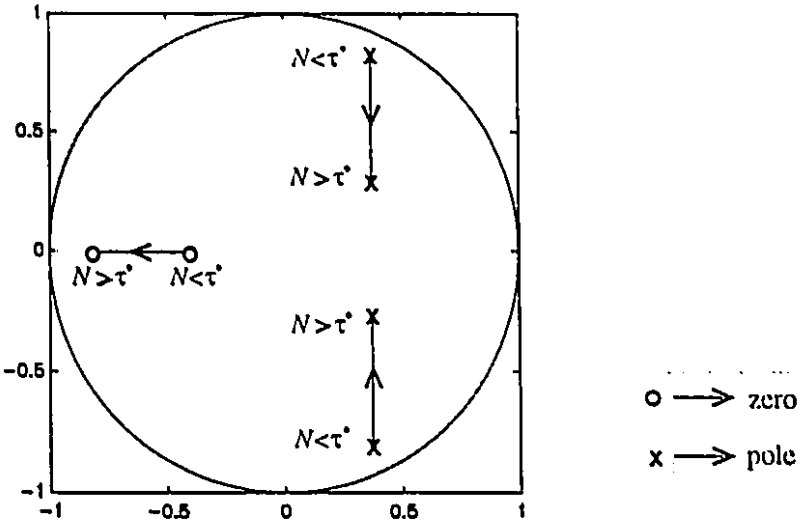


Figure 5.13: Jump case: The initial and final pole-zero locations of the time variant ARMA(2,1) system.

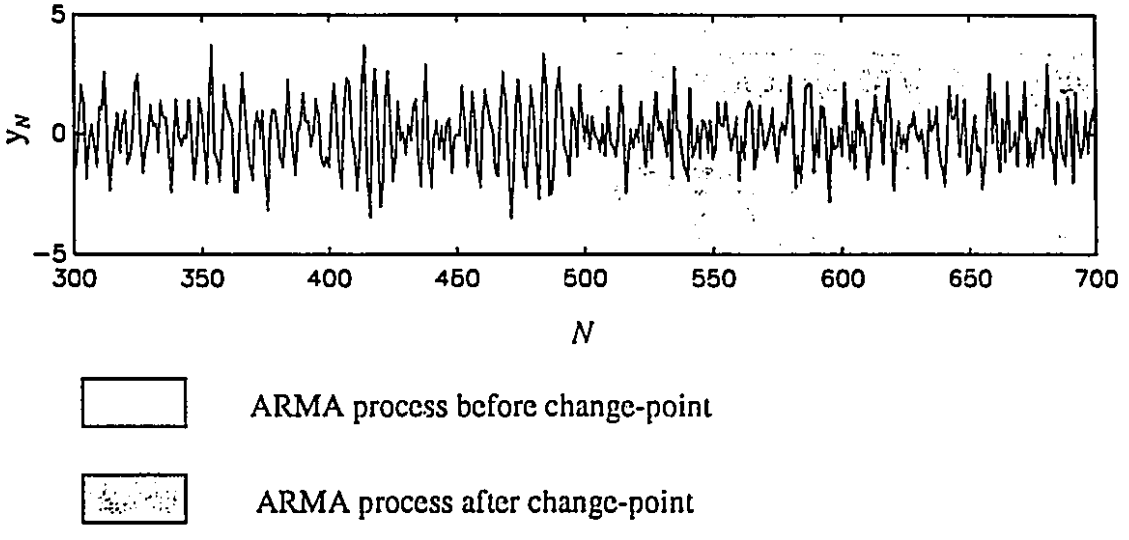


Figure 5.14: The data process y^N generated by model $\mathcal{M}_{\tau^*}$ which has an abrupt change-point at $\tau^* = 500$.

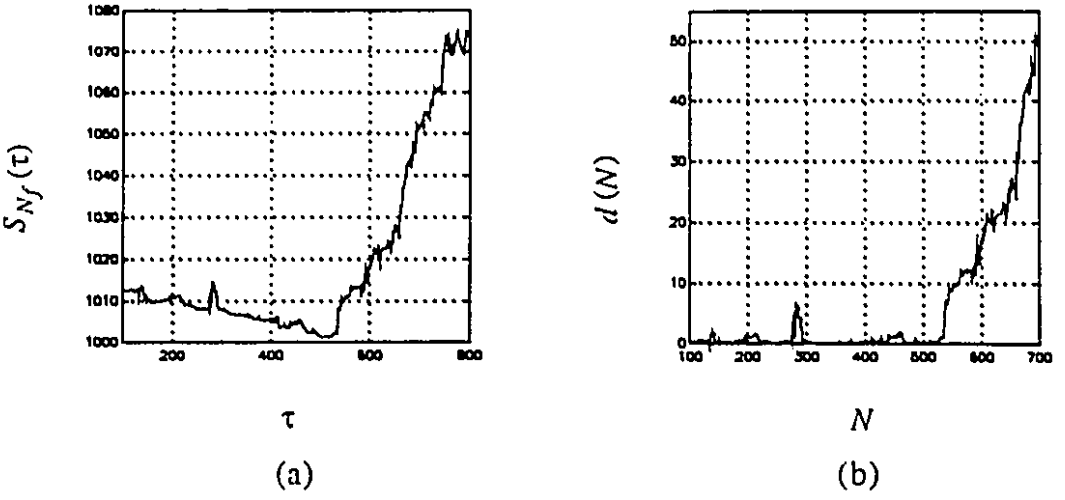


Figure 5.15: a) Predictive stochastic complexity with respect to model classes $\mathcal{M}_\tau$; b) The on-line detector.

thus the estimation error of the change-point is only $\hat{\tau} - \tau^* = 3$.

The relationship between those thresholds h for which no false alarms occur, and the detection delay T_d is depicted in Figure 5.16.

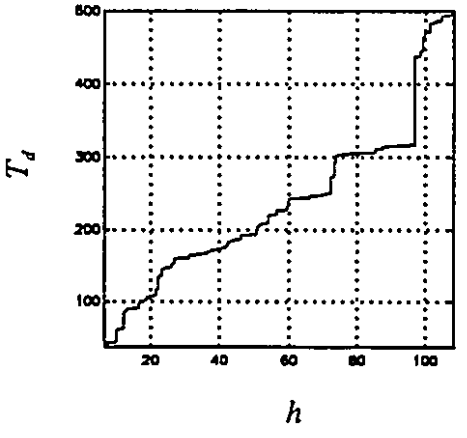


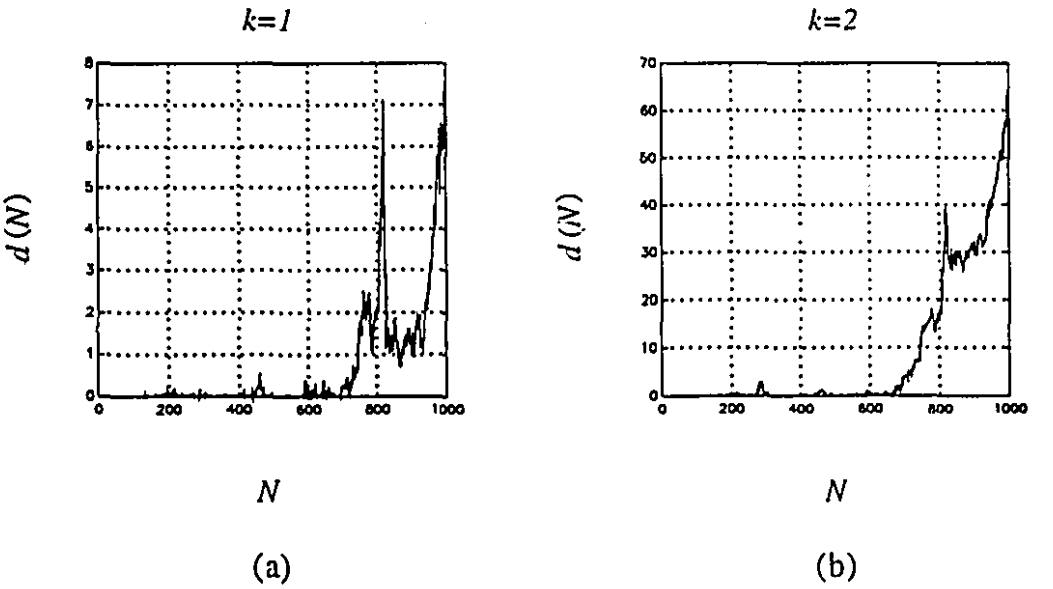
Figure 5.16: The detection delay $T_d = T - \tau^*$ as a function of the threshold h.

5.6.4 Change-Point Detection and Undermodeling

Here, the issue of undermodeling in change-point detection will be considered via simulations. The idea will be to detect the slowly time variant change-point introduced in Section 5.6.1 and the abrupt change-point of Section 5.6.3 using AR models instead of ARMA models to generate the prediction errors ϵ_N^{λ}. Thus the model classes $\mathcal{M}_\tau$ will correspond to time-invariant ARMA models until time τ and time-variant AR models after τ. We will see that an increase in change-point detection performance is obtained for both types of change-points being considered.

Let us start with the slowly time variant change-point case. In Figure 5.17 the detectors $d(N)$, when AR(k), $k = 1, \ldots, 4$, model classes are used, are displayed.

To help discern the increase of performance of the change-point detection algorithm when using an AR(2) model instead of the more complex ARMA(2, 1) model employed in Section 5.6.1, the detectors $d(N)$ which result in each of these cases are plotted in Figure 5.18.



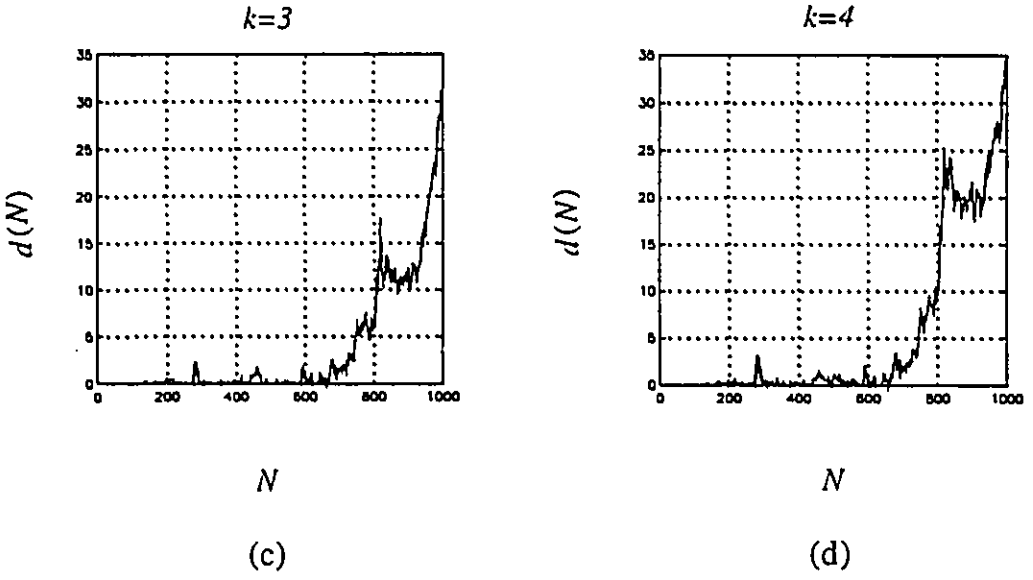


Figure 5.17: The on-line slowly time variant change-point detector $d(N)$ with model classes: a) AR(1); b) AR(2); c) AR(3); d) AR(4).

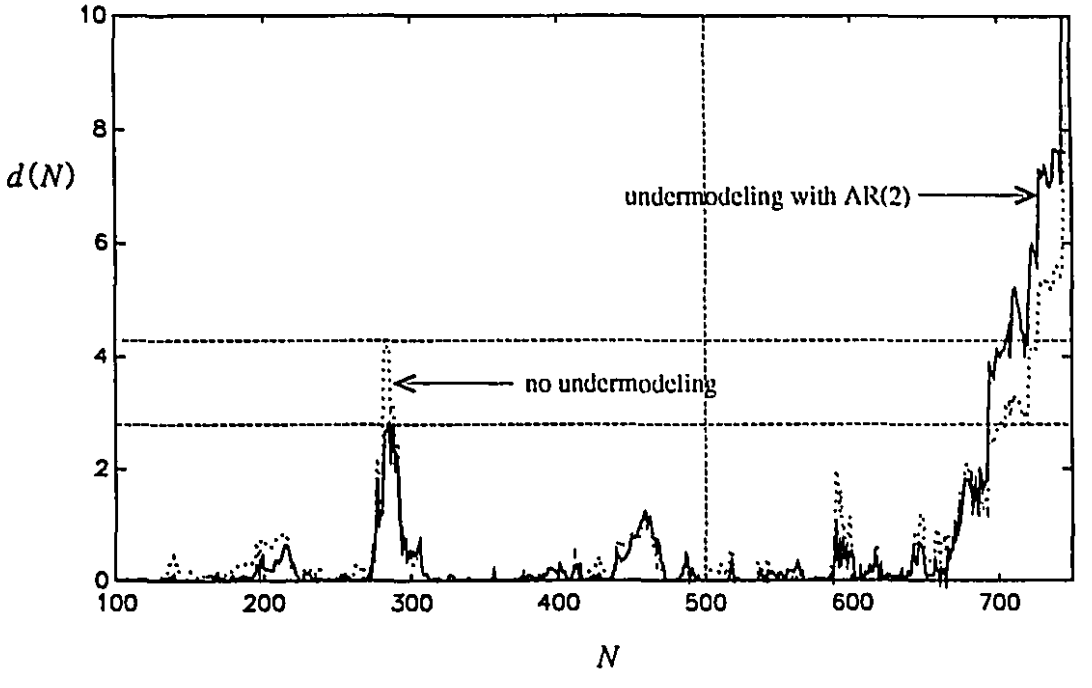


Figure 5.18: Comparison of the on-line slowly time variant change-point detectors $d(N)$ obtained when not using undermodeling (that is when employing ARMA(2,1) models) and when using undermodeling with an AR(2).

Let us now move to the case of the abrupt change-point when using undermodeling. As with the slowly time variant case, AR(k), $k = 1, \ldots, 4$, model classes are applied to construct the detectors $d(N)$. These are shown in Figure 5.19.

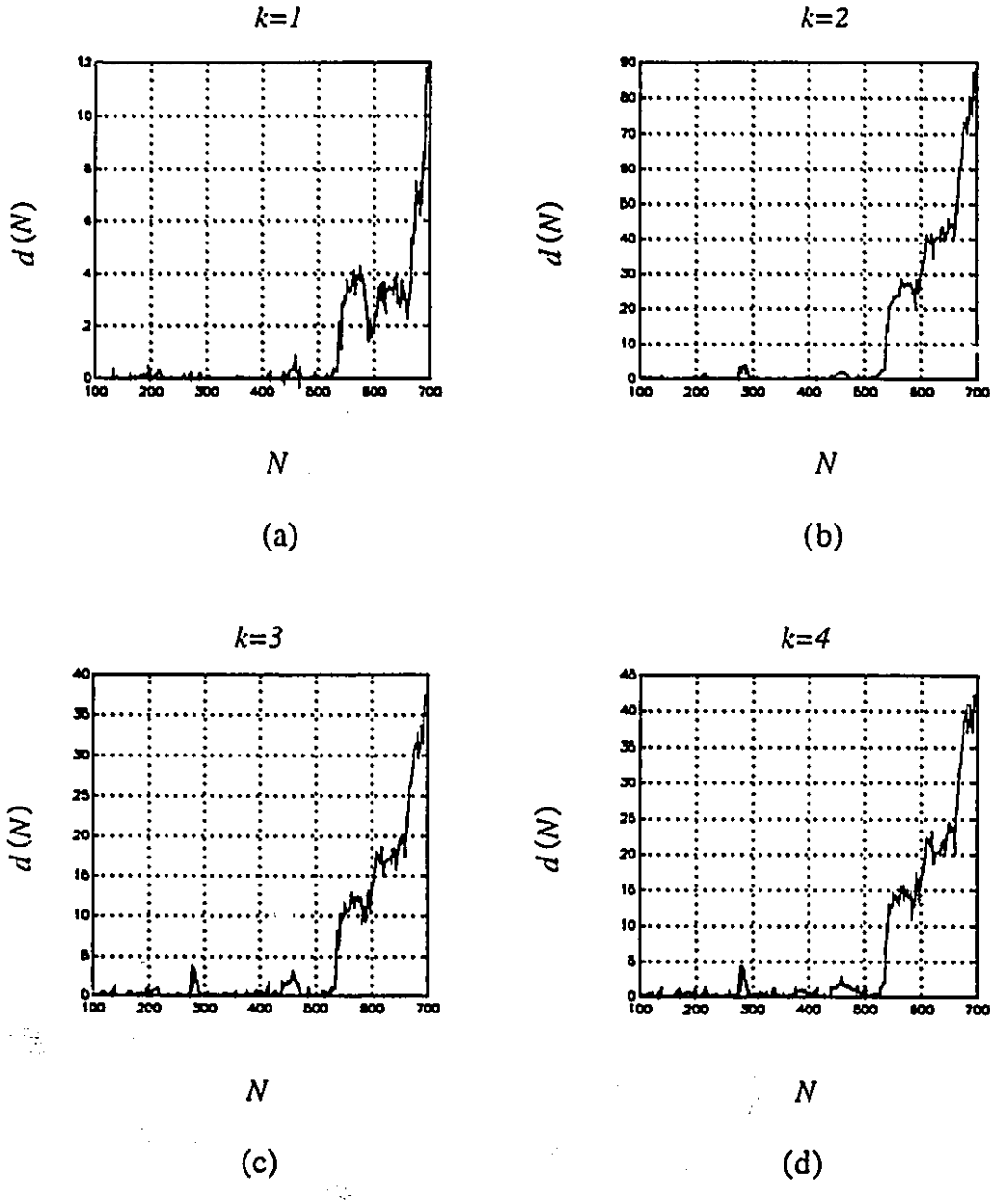


Figure 5.19: The on-line abrupt change-point detector $d(N)$ with model classes: a) AR(1); b) AR(2); c) AR(3); d) AR(4).

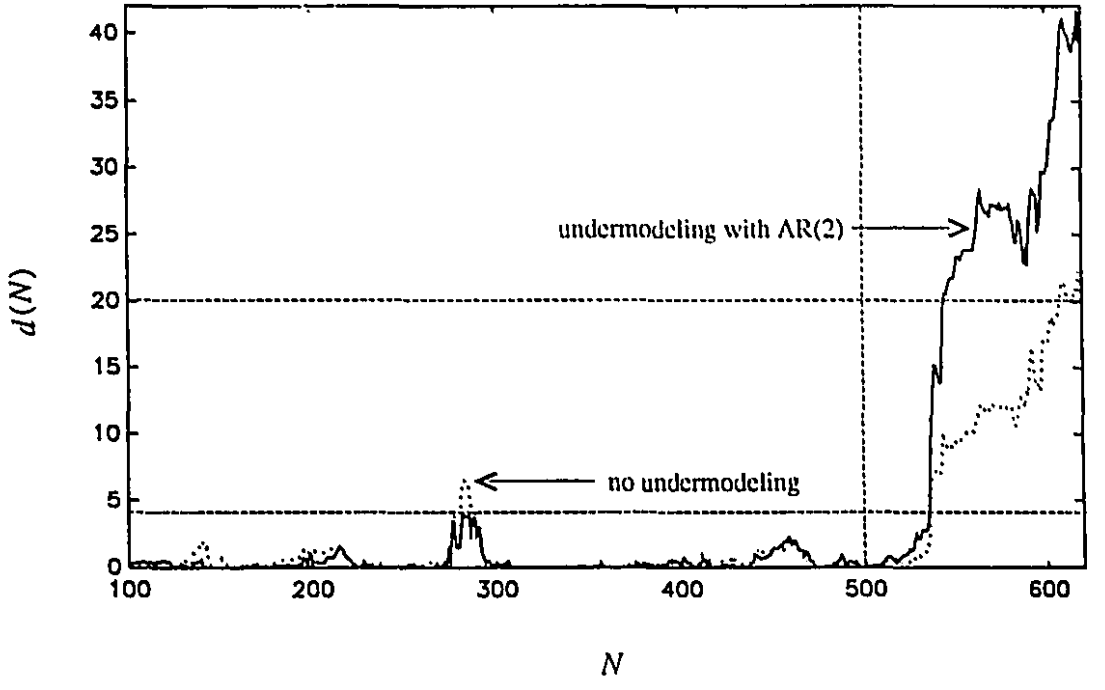


Figure 5.20: **Comparison of the on-line abrupt change-point detectors $d(N)$ obtained when not using undermodeling (that is when employing ARMA(2,1) models) and when using undermodeling with an AR(2).**

Again, the performance enhancement of the change-point detection algorithm, when using an AR(2) model (note that this is the model that performs best among the AR models being considered) instead of the ARMA$(2,1)$ model, can be clearly observed when the detectors $d(N)$ for these two cases are plotted together as in Figure 5.20.

5.6.5 The Detector $d(N)$ Versus a "naive" Detector Based on Monitoring Parameter Estimates

Since the recursive prediction error algorithm with fixed gain has the ability to track time variant parameters, one might think of using the parameter estimates of the system, without further processing, as a change-point detection scheme. We shall

illustrate here that this "naive" change-point detection technique is out-performed by the stochastic complexity based change-point detection method.

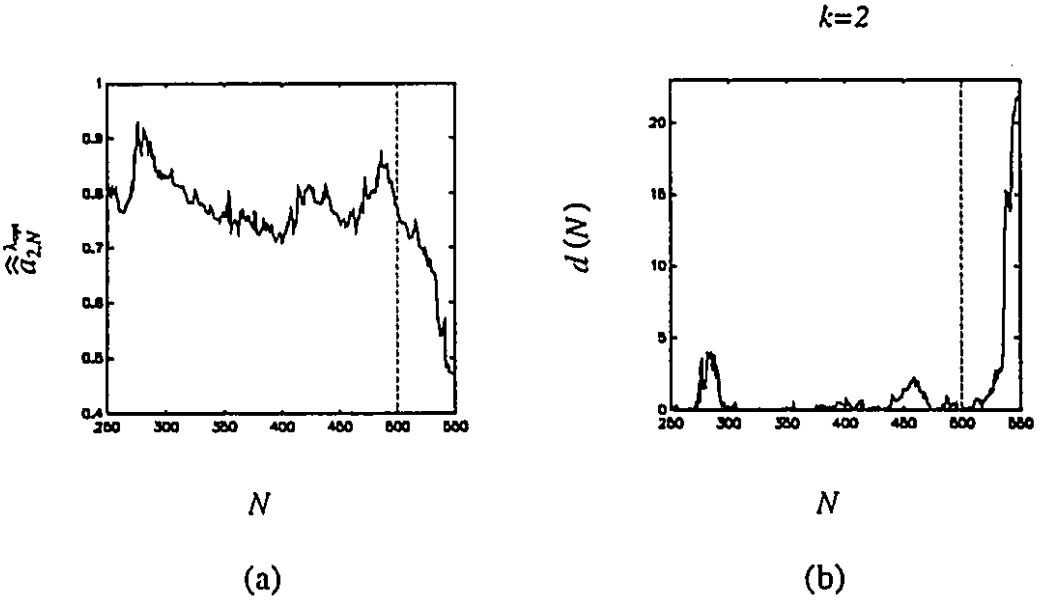


Figure 5.21: **The abrupt change-point case: a) The "naive" detector; b) The stochastic complexity based detector.**

From Figures 3.5–3.7, we can observe that the estimator $\widehat{\widehat{a}}^{\lambda}_{2,N}$ of $a^{*}_{2,N}$ best tracks its corresponding "true" time variant parameter if it is compared to the tracking performance of the estimates $\widehat{\widehat{a}}^{\lambda}_{1,N}$ and $\widehat{\widehat{c}}^{\lambda}_{1,N}$. Therefore, in Figure 5.21, the estimator $\widehat{\widehat{a}}^{\lambda}_{2,n}$ and the detector $d(N)$ (for the abrupt change-point of Section 5.6.1) are plotted side by side so as to appreciate the improvement obtained by using $d(N)$ as opposed to $\widehat{\widehat{a}}^{\lambda}_{2,N}$ for the alarm signal.

In Figure 5.22 the same comparison is represented (as that displayed in Figure 5.21) but in this case for the slowly time variant change-point of Section 5.6.3. Once again a similar conclusion is drawn: the stochastic complexity based change-point scheme using undermodeling outperforms the "naive" change-point method based on monitoring the parameter estimates.

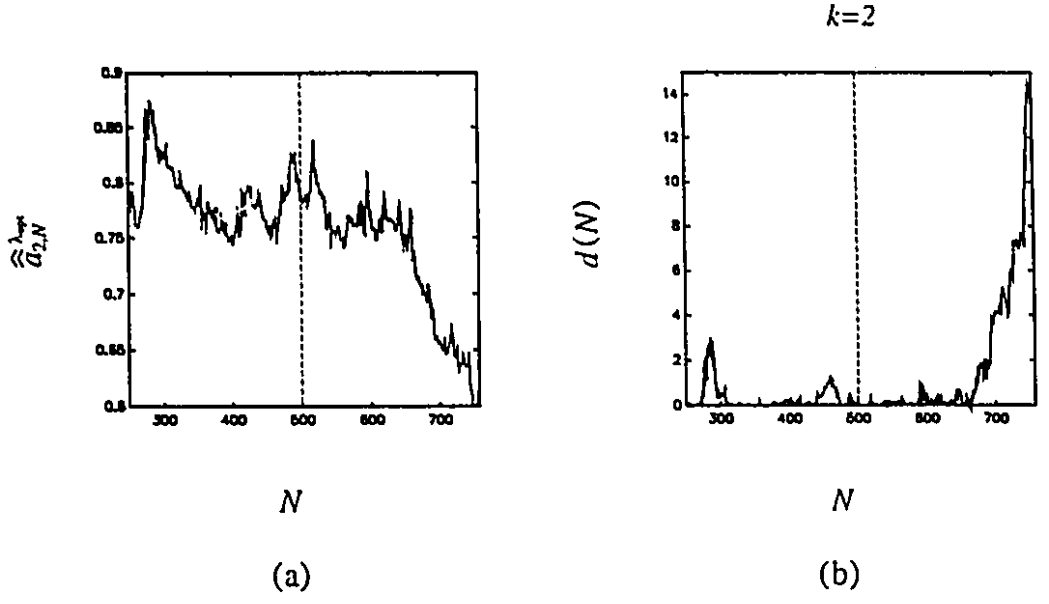


Figure 5.22: **The slowly time variant change-point case: a) The "naive" detector; b) The stochastic complexity based detector.**

Chapter 6

Adaptive Control of an LSS

Even though the use of a properly designed fixed feedback controller would reduce the effect of plant uncertainty on closed loop system performance, it would not be well suited to the control of physical systems with little a-priori knowledge of their dynamics. Since the beginning of the 50's a new area of control, known as adaptive control, emerged. It was triggered by the need to tackle the control of physical systems whose dynamics experiences major alterations. For example, we can mention the control of aircrafts and ships whose dynamics is greatly affected by the different conditions under which these physical systems ought to operate. Another situation in which adaptive control is deemed necessary is when experimentation with the physical system is not possible in advance. For instance, this applies to many control problems found in process control, in particular in chemical engineering, and in economics. (For applications in the area of adaptive control the reader is referred to [NM80] and [Åst83].)

The theory of adaptive control has at present time gained certain level of maturity; results on stability, both local and global, as well as performance of some adaptive schemes are available. Recent books on adaptive control amongst the now extensive literature are: [IK83], [GS84], [KV86] [Cai88], [AW89], [SB89], and [CG91].

The main current approaches of adaptive control are: i) the model reference adaptive systems (c.f., e.g., [Par66] for an earlier work and [Lan79] for a book dedicated to this approach); ii) self-tunning controllers (c.f., e.g., [Kal58] for an earlier work and [Alo87] for a summary of results when using ARMAX models). The first approach is based on updating the parameters of the controller directly from information of the error generated between measured and model outputs. The second approach—which is generally formulated in a stochastic framework—estimates the parameters of the plant and uses these new estimates to recompute the control law. This approach is much more involved than the simple model matching approach. Moreover, the model reference approach can be viewed as a particular case of the latter method (c.f., [SB89]). In this chapter, the adaptive controller to be presented falls in the category of self-tunning regulators.

In the work of [Ger90a], an adaptive control problem for finite dimensional time invariant linear stochastic systems was considered. The structure of the adaptive controller was based on the certainty equivalence principle which consists of using the latest estimates given by an identification scheme to the design of an appropriate controller, as if the parameter estimates were the true parameters of the system. A very important feature of the identification scheme is that since it is formulated for the closed loop system, the parameter estimates are the ones giving optimal performance for the controlled system.

In the cited work a link was established between open loop and closed loop identifiability. More precisely, closed loop parametric identifiability of a system driven by a suboptimal adaptive controller was proven under the assumption of open loop identifiability of the corresponding open loop system driven by a persistently exciting open loop control signal.

As usual the identification scheme was formulated as that of finding the root

of the gradient of an appropriate cost function. It was shown that this problem could be solved successfully via Ljung's scheme, arriving at an on-line computable adaptive controller. When implementing the adaptive controller, one of the main computationally expensive features was its dependence on the directional derivatives of the adaptive feedback transfer function gain. In this chapter, we will prove that in fact there is no such dependence, reducing considerably the computational complexity of the algorithm.

We will also extensively illustrate the adaptive control methodology for an ARX system, showing the stability and tracking capability of the adaptive controller. Moreover, we will show the effect of the dither process (c.f., [Cai88]) on the closed loop performance, a process which is embedded in the controller to guarantee identifiability of the closed loop system.

6.1 Closed Loop Identifiability from Open Loop Identifiability

In this section a self contained summary of [Ger90a] will be presented. Consider the parametrized sets of transfer functions

$$H^w(\theta) = \left(H^w(\theta, e^{-i\lambda})\right) \qquad \text{and} \qquad H^u(\theta) = \left(H^u(\theta, e^{-i\lambda})\right),$$

which are $m \times m$ and $m \times r$ matrices respectively, defined over $\theta \in D_\theta \subset \mathbb{R}^k$, where D_θ is a compact domain. Let $\theta^* \in \text{int}D_\theta$, and consider the output processes $y(\theta^*) = (y_n(\theta^*))$, defined by the discrete-time linear stochastic system

$$y(\theta^*) = H^w(\theta^*)w + H^u(\theta^*)u, \tag{6.1}$$

where $w = (w_n)$ is an m-dimensional noise process, and $u = (u_n)$ an r-dimensional input process, satisfying the following condition:

Condition 6.1.1 *The process (u, w) is defined over some probability space $(\Omega, \mathcal{F}, P)$ and is jointly second-order stationary with zero mean. Let $(\mathcal{F}_n)$ be a monotone increasing σ-algebra of $\mathcal{F}$, then (u_n, w_n) is $\mathcal{F}_n$ adapted. The noise process (w_n) is itself an orthogonal process with $\mathbb{E}\, w_n w_n^T = \Lambda > 0$, for all n. The input process can be decomposed as $u = u^{\perp} + u^{-}$ such that $(w, u^{\perp}, u^{-})$ is wide sense stationary, $u^{\perp}$ is orthogonal to w, and u^{-} is predictable with respect to w (i.e. $u_n^{-} \in Sp\{w_i; i \leq n-1\}$).*

Further conditions imposed on the stochastic system (6.1) are as follows:

Condition 6.1.2 *$H^w(\theta, e^{-i\lambda})$, $\left(H^w(\theta, e^{-i\lambda})\right)^{-1}$, and $H^u(\theta, e^{-i\lambda})$, are boundary functions of H^∞-functions on the unit disc D.*

Let the analytic extension onto the unit disc D of a transfer function $H(\theta, e^{-i\lambda})$ be denoted by $H(\theta, z)$.

Condition 6.1.3 *The matrix functions $H^w(\theta, z)$, and $H^u(\theta, z)$, are smooth with respect to θ in the strong topology of $H^\infty(\theta)$. Moreover, $H^w(\theta, 0) = I$, where I denotes the $m \times m$ identity matrix.*

The parameter θ^* of system (6.1) is identified as follows. First, choose a trial parameter $\theta \in D_\theta$, and compute the residuals

$$\epsilon(\theta, \theta^*) = (H^w(\theta))^{-1} \left(y(\theta^*) - H^u(\theta)u\right),$$

where $\epsilon(\theta, \theta^*) = (\epsilon_n(\theta, \theta^*))$ is often called the prediction error process. Define the cost function

$$W(\theta, \theta^*) = \lim_{n \to \infty} \tfrac{1}{2}\, \mathbb{E}\, |\epsilon_n(\theta, \theta^*)|^2.$$

Then, a necessary condition to identify system (6.1) is to assure $W(\theta, \theta^*)$ a local minimum at $\theta = \theta^*$. However, this condition is not sufficient since there could exist a manifold of global minima. To avoid this situation, let the following definitions be introduced:

Definition 6.1.1 A second-order stationary process x is said to be persistently exciting if its spectral density matrix $\phi(\cdot)$ for some constant $c > 0$ satisfies $\phi(e^{-i\lambda}) > cI$, for all $-\pi \leq \lambda \leq \pi$.

Definition 6.1.2 System (6.1) is said to be *locally identifiable under persistent excitation* if for all persistently exciting input processes (w, u) satisfying Condition 6.1.1 the Hessian matrix

$$\frac{\partial^2}{\partial \theta^2} W(\theta, \theta^*) \Big|_{\theta=\theta^*}$$

is positive definite.

System (6.1) can then be said to be identifiable if it is locally identifiable under persistent excitation, and its associated cost function $W(\theta, \theta^*)$ has a root at $\theta = \theta^*$.

For example, it is easy to show that in the case of an open loop system,

$$y^o(\theta^*) = H^w(\theta^*)w + H^u(\theta^*)u^o. \tag{6.2}$$

where (u^o) is an open loop input such that $(u^o)^- \equiv 0$, the associated cost function denoted $W^o(\theta, \theta^*)$ has a global minimum at $\theta = \theta^*$. Therefore, if the open loop system (6.2) is locally identifiable under persistent excitation the system's parameter θ^* can be identified.

Let us introduce the closed loop system associated with the open loop system (6.2). Observe that condition 6.1.1 allows closed loop control inputs $u = u^c$, formed as combinations of a causal feedback and an external dither. Therefore, define the closed loop system associated with the open loop system (6.2) as

$$y^c(\theta, \theta^*) = H^w(\theta^*)w + H^u u^c(\theta, \theta^*), \tag{6.3}$$

with the closed loop control input given by

$$u^c(\theta, \theta^*) = K(\theta)(-y^c(\theta, \theta^*) + u^o) + v, \tag{6.4}$$

where $K(\theta)$ is a designed $p \times m$ feedback transfer function gain, and $v = (v_n)$ is a dither. Note that the dependence of the feedback gain is with respect to θ and not θ^* since the latter corresponds to the true system's parameter which is assumed to be unknown. The use of an external dither process will prove essential to achieving closed loop identifiability. This type of controller is sometimes referred to as continuously disturbed control (c.f., [Cai88]). The following conditions are imposed on the closed loop system (6.4):

Condition 6.1.4 *The transfer functions $K(\theta)$, and $(I+H^u(\theta^*)K(\theta))^{-1}$ are in $H^\infty(D)$, for all $\theta \in D_\theta$. Moreover, $K(\theta)$ is smooth with respect to θ in the strong topology of $H^\infty(D)$.*

Condition 6.1.5 *The dither v is a second-order stationary persistently exciting process orthogonal to the noise process w.*

The prediction-error process for the closed loop case is given by

$$\epsilon^c(\theta,\theta^*) = (H^w(\theta))^{-1}\,(y^c(\theta,\theta^*) - H^u(\theta)u^c(\theta,\theta^*)) \tag{6.5}$$

and the associated cost function by

$$W^c(\theta,\theta^*) = \lim_{n\to\infty} \tfrac{1}{2}\,\mathbb{E}\,|\epsilon_n^c(\theta,\theta^*)|^2.$$

In [Ger90a] closed loop identifiability was achieved under the assumption of open loop identifiability.

Theorem 6.1.1 *Under Conditions 6.1.1–6.1.5, and the assumption that the open-loop system (6.2) is locally identifiable under persistent excitation, the closed loop system (6.3–6.4) is also locally identifiable under persistent excitation, i.e.*

$$\frac{\partial^2}{\partial\theta^2} W^c(\theta,\theta^*)\,\Big|_{\theta=\theta^*},$$

is positive definite.

Using Theorem 6.1.1 we can show that the cost-function $W^c(\theta, \theta^*)$ has a local minimum at $\theta = \theta^*$. Indeed,

$$\begin{aligned} \frac{\partial}{\partial \theta} W^c(\theta, \theta^*) \Big|_{\theta=\theta^*} &= \lim_{n\to\infty} \mathbb{E} \left(\frac{\partial}{\partial \theta} \epsilon_n^c(\theta^*, \theta^*) \right)^T \cdot \epsilon_n^c(\theta^*, \theta^*) \\ &= \lim_{n\to\infty} \mathbb{E} \left(\frac{\partial}{\partial \theta} \epsilon_n^c(\theta^*, \theta^*) \right)^T \cdot w_n, \end{aligned} \tag{6.6}$$

and since it is easy to show that $\partial/\partial\theta \epsilon^c(\theta^*, \theta^*)$ is $\mathcal{F}_{n-1}$ we get the claim. Therefore, the parameter θ^* of the closed loop system (6.3–6.4) can be in principle identified. However as shown in the next section there are some limitations to this solution which will have to be overcome.

6.2 Closed Loop Identification via Ljung's Scheme

A process x is said to be computable if it can be obtained by a known transformation of a known monitored process. This monitored process will usually be the output of an unknown physical system driven by a partially or completely unknown input. For example $\epsilon_n^c(\theta, \theta^*)$ is computable. In this case $y^c(\theta, \theta^*)$ is the monitored process, and (6.5) describes its known transformation.

The identification of θ^* is based on finding the roots of the equation

$$\frac{\partial}{\partial \theta} W^c(\theta, \theta^*) = 0. \tag{6.7}$$

It is important to note that (6.7) actually represents the simultaneous identification and control of system (6.3–6.4)

An important drawback of (6.7) is that it is not computable, and thus cannot be solved in practice. To prove this claim, note that

$$\frac{\partial}{\partial \theta} W^c(\theta, \theta^*) = \mathbb{E} \left(\frac{\partial}{\partial \theta} \epsilon_n^c(\theta, \theta^*) \right)^T \cdot \epsilon_n^c(\theta, \theta^*), \tag{6.8}$$

requires the computation of process $\partial/\partial\theta\epsilon^c(\theta,\theta^*)$ which in turn, by (6.5), requires the computation of process $\partial/\partial\theta y^c(\theta,\theta^*)$. To now show that $\partial/\partial\theta y^c(\theta,\theta^*)$ is not computable, first combine (6.3) and (6.4) to get

$$y^c(\theta,\theta^*) = H^w(\theta^*)w + H^u(\theta^*)K(\theta)(-y^c(\theta,\theta^*)+u^o) + H^u(\theta^*)v, \qquad (6.9)$$

and differentiate (6.9) in the direction $(\nu,0)$, where $\nu\in\mathbb{R}^k$, $\nu\neq 0$, to obtain

$$y^c_{\nu,0}(\theta,\theta^*) = [I+H^u(\theta^*)K(\theta)]^{-1}H^u(\theta^*)K_\nu(\theta)(-y^c(\theta,\theta^*)+u^o). \qquad (6.10)$$

Since in (6.10) $H^u(\theta^*)$ is unknown, then $y^c_{\nu,0}(\theta,\theta^*)$ is not computable and thus cannot be generated by the user.

As noted in [Ger90a], in practice, a stochastic approximation scheme--performed via Ljung's scheme—is used to identify θ^*. However, Ljung's scheme cannot be applied directly to the solution of (6.7) since $W^c(\theta,\theta^*)$ is not computable. Thus, we need first to find a computable approximation, say $U(\theta,\theta^*)$, of the cost function $W^c(\theta,\theta^*)$, which would not destroy the desirable properties of $W^c(\theta,\theta^*)$.

In [Ger90a] the computable approximation process $U(\theta,\theta^*)$ was given in terms of the derivative of the feedback gain $K(\theta)$. This dependence is at first hand obvious since the directional derivatives $(\nu,0)$ of the process $y^c(\theta,\theta^*)$ are expressed in terms of $K_\nu(\theta)$ as shown by (6.10). This dependence dramatically increases the computational complexity of the adaptive control algorithm. In fact, it is of common practice to design the feedback gain $K(\theta)$ by solving an optimal control problem, assuming that the trial parameter θ is the true parameter. It is well-known that this optimal controller is obtained via the solution of a Ricatti equation. Thus, the computation of $K_\nu(\theta)$ involves, in this case, the differentiation of the Ricatti equation. This process has to be repeated at each time interval since new estimates of θ^* are computed. Fortunately, as it will be shown, there is actually no dependence of the adaptive controller presented in [Ger90a] on the term $K_\nu(\theta)$, thus avoiding the need of differentiating Ricatti equations.

Recall that in order to estimate θ^*, (6.8) must be solved, and since $\partial/\partial\theta\epsilon^c(\theta,\theta^*)$ is not computable, a computable approximation of $\partial/\partial\theta\epsilon^c(\theta,\theta^*)$ is sought, which will be denoted by $\partial/\partial\theta\psi^c(\theta,\theta^*)$.

Lemma 6.2.1 *A computable function $U(\theta,\theta^*)$ which approximates $W(\theta,\theta^*)$ is given by*

$$U(\theta,\theta^*) = \mathbb{E}\,\partial/\partial\theta\psi_n^c(\theta,\theta^*)\epsilon_n^c(\theta,\theta^*),$$

where the directional derivative of $\psi^c(\theta,\theta^*)$ in the direction $(\nu,0)$ is

$$\psi_{\nu,0}(\theta,\theta^*) = (H^w(\theta))^{-1}\left[H_\nu^u(\theta)u^c(\theta,\theta^*) - H_\nu^w(\theta)\epsilon^c(\theta,\theta^*)\right]. \tag{6.11}$$

REMARK. Note that $\psi_{\nu,0}$ is independent of $K_\nu(\theta)$.

PROOF. Rewrite (6.5) as

$$H^w(\theta)\epsilon^c(\theta,\theta^*) = y^c(\theta,\theta^*) - H^u(\theta)[-K(\theta)y^c(\theta,\theta^*) + K(\theta)u^o + v] \tag{6.12}$$

Then, the process $\epsilon_{\nu,0}^c(\theta,\theta^*)$ is simply obtained by differentiating (6.12) and thus we get

$$\begin{aligned} H_\nu^w(\theta)\epsilon^c(\theta,\theta^*) + H^w(\theta)\epsilon_{\nu,0}^c(\theta,\theta^*) \;=\; & y_{\nu,0}^c(\theta,\theta^*) - H_\nu^u(\theta)\left(K(\theta)(-y^c(\theta,\theta^*) + u^o) + v\right) \\ & -H^u(\theta)[-K_\nu(\theta)y^c(\theta,\theta^*) - K(\theta)y_{\nu,0}^c(\theta,\theta^*) \\ & +K_\nu(\theta)u^o] \end{aligned} \tag{6.13}$$

Recall that the process $y_{\nu,0}^c(\theta,\theta^*)$ was not computable since the term $H^u(\theta^*)$ in (6.10) is unknown. Let us rewrite (6.10) as

$$(I + H^u(\theta^*)K(\theta))y_{\nu,0}^c(\theta,\theta^*) = -H^u(\theta^*)K_\nu(\theta)(y^c(\theta,\theta^*) - u^o). \tag{6.14}$$

The first approximation step is to substitute $H^u(\theta^*)$ by $H^u(\theta)$ in (6.14) to get the approximation process $z_{\nu,0}^c(\theta,\theta^*)$ of $y_{\nu,0}^c(\theta,\theta^*)$ computed by

$$(I + H^u(\theta)K(\theta))\,z_{\nu,0}^c(\theta,\theta^*) + H^u(\theta)K_\nu(\theta)(y^c(\theta,\theta^*) - u^o) = 0. \tag{6.15}$$

Substituting $y^c_{\nu,0}(\theta,\theta^*)$ by $z^c_{\nu,0}(\theta,\theta^*)$ in (6.13) the computable approximation process $\psi^c_{\nu,0}(\theta,\theta^*)$ of $\epsilon^c_{\nu,0}(\theta,\theta^*)$ is obtained. Thus we get

$$\begin{aligned} H^w_\nu(\theta)\epsilon^c(\theta,\theta^*) + H^w(\theta)\psi^c_{\nu,0}(\theta,\theta^*) &= [I + H^u(\theta)K(\theta)]\, z^c_{\nu,0}(\theta,\theta^*) - \\ & H^u_\nu(\theta)\,[-K(\theta)y^c(\theta,\theta^*) + K(\theta)u^o + v] + \\ & H^u(\theta)K_\nu(\theta)(y^c(\theta,\theta^*) - u^o). \end{aligned}$$

Observe that since (6.15) holds, the first and third terms of the second part of (6.16) cancel each other. Therefore,

$$H^w_\nu(\theta)\epsilon^c(\theta,\theta^*) + H^w(\theta)\psi^c_{\nu,0}(\theta,\theta^*) = H^u_\nu(\theta)(-K(\theta)y^c(\theta,\theta^*) + K(\theta)u^o + v) \quad (6.16)$$

which finally gives

$$\psi_{\nu,0}(\theta,\theta^*) = [H^w(\theta)]^{-1}\,[H^u_\nu(\theta)(-K(\theta)y^c(\theta,\theta^*) + K(\theta)u^o + v) - H^w_\nu(\theta)\epsilon^c(\theta,\theta^*)] \quad \blacksquare$$

Corollary 6.2.1 *We have*

$$\psi_{\nu,0}(\theta^*,\theta^*) = \epsilon^c_{\nu,0}(\theta^*,\theta^*). \quad (6.17)$$

PROOF. If $\theta = \theta^*$, equation (6.14) can be used directly to cancel the first and last two terms of equation (6.13) leading to the stated equality. ∎

Based on the above corollary the following theorem is established:

Theorem 6.2.1 *The computable function $U(\theta,\theta^*)$ satisfies the following properties*

$$U(\theta,\theta^*)\,\Big|_{\theta=\theta^*} = 0, \quad (6.18)$$

and

$$\frac{\partial}{\partial\theta}U(\theta,\theta^*)\,\Big|_{\theta=\theta^*}$$

is positive definite.

PROOF. Simply note that from Corollary (6.2.1) we have

$$U(\theta^*, \theta^*) = W^c(\theta^*, \theta^*). \quad \blacksquare$$

Now the computable equation (6.18) can be solved via Ljung's scheme under certain additional conditions imposed on the input noise process w (c.f., [Ger88b]). Note that the crucial stability condition needed when applying Ljung's scheme is satisfied. Indeed,

$$\dot{\theta}(t) = -U(\theta(t), \theta^*),$$

which is the associated ODE to equation (6.18), is asymptotically exponentially stable at $\theta = \theta^*$, since the gradient of $U(\theta^*, \theta^*)$ is positive definite.

Let us finish with some remarks: Since the addition of a dither makes the controller suboptimal, the variance of the dither becomes an important design parameter. In this respect some promising directions of research are opened by the use of predictive stochastic complexity in the optimization of the performance of continually perturbed adaptive controllers. More precisely, when using fixed gain, the size of the dither could be optimized so as to minimize the variance of the estimation error. The result of Theorem 10.8 in [Ger91c] can be taken as the first steps on the study of the effect of parameter uncertainty on the closed loop performance.

6.3 An Application: Adaptive Control of an ARX System

In this section, the adaptive control methodology introduced for finite dimensional time-invariant linear stochastic systems will be illustrated by an autoregressive system with an exogenous input, or ARX system. Extensive simulations of this particular

system will be provided which demonstrate the stability and tracking capability of the adaptive controller. We will also show the effect of the dither on closed loop performance.

Consider the following open loop ARX system

$$x^o_{N+1}(\theta^*) = a^* x^o_N(\theta^*) + b^* u^o_N + e_{N+1}, \tag{6.19}$$

where $\theta^* \in \text{int}D_\theta, D_\theta \subset \mathbb{R}^2$ denotes the vector composed of the coefficients a^* and b^*, the process (e_N) is the input noise process, and the process (u^o_N) is a deterministic reference input.

Let us pick a $\theta \in D_\theta$ and associate to the open loop system (6.19) the closed loop system

$$x^c_{N+1}(\theta^*) = a^* x^c_N(\theta^*) + b^* u^c_N + e_{N+1}, \tag{6.20}$$

with feedback law

$$u^c_N(\theta, \theta^*) = -k(\theta) x^c_N(\theta, \theta^*) + u^o_N + v_N, \tag{6.21}$$

where v_N is a dither. The gain $k(\theta)$ is designed as the optimal gain found by solving the optimal stochastic quadratic control problem, taking θ as the "true" parameter of the system. The cost function to be minimized is

$$\mathbb{E}\left(m[x^c_N(\theta, \theta^*)]^2 + l[u^c_N(\theta, \theta^*)]^2\right), \tag{6.22}$$

for some $m > 0$ and $l > 0$. As is well-known $k(\theta)$ is found by solving the Ricatti equation

$$P + a^2 P - aPb(l + b^2 P)^{-1} bPa + m = 0,$$

and computing $k(\theta) = [l + b^2 P]^{-1} bPa$.

We assume that Conditions 6.1.1–6.1.5 hold for system (6.20). Therefore, using Theorem 6.2.1, the adaptive control problem can be solved via Ljung's scheme by finding the root of the following stochastic equation

$$\mathbb{E}\partial/\partial\theta \psi^c_N(\theta, \theta^*)\epsilon^c_N(\theta, \theta^*) = 0.$$

In this case, the prediction error process associated with system (6.20) is

$$\epsilon^c_{N+1}(\theta,\theta^*) = x^c_{N+1}(\theta,\theta^*) - (a - bk(\theta))x^c_N(\theta,\theta^*) - b(u^o_N + v_N)$$

Applying (6.11), the approximating process for the derivative of the prediction error process is given by

$$\psi^c_N(\theta,\theta^*) = \begin{bmatrix} -x^c_N(\theta,\theta^*) \\ k(\theta)x^c_N(\theta,\theta^*) - u^o_N - v_N \end{bmatrix}.$$

Now we are ready to apply Ljung's scheme, which in this case coincides with the recursive least square algorithm, given by the following equations

$$\hat{\hat{\theta}}_N = \hat{\hat{\theta}}_{N-1} - \frac{1}{N}\left(\hat{\hat{R}}_{N-1}\right)^{-1}\psi^c_N \cdot \epsilon_N \tag{6.23}$$

$$\hat{\hat{R}}_N = \hat{\hat{R}}_{N-1} + \frac{1}{N}\left(\psi^c_N \cdot (\psi^c_N)^T - \hat{\hat{R}}_{N-1}\right), \tag{6.24}$$

with initial conditions $\hat{\hat{\theta}}_0$ and $\hat{\hat{R}}_0$.

6.3.1 Simulation of the Adaptive Control of an ARX System

Let us now perform a simulation of the adaptive control problem introduced in Section 6.3. The parameter values of the ARX system (6.20) are

$$a^* = a^*_N = .98, \qquad \text{and} \qquad b^* = b^*_N = .01.$$

The input process (e_N) is Gaussian white noise with mean 0 variance 1. Set the reference input $u^o_N = 3\sin([\pi/50]N)$, and $m = 10$, $l = .01$ for the parameters of the cost function 6.22. The dither (v_N) is also Gaussian white noise with mean 0 and variance $\sigma^2_v = .04$ uncorrelated to the input noise (e).

We ran the simulation for 200 iterations, with initial conditions $\hat{\hat{\theta}}_0 = [.70]$, $\hat{\hat{R}}_0 = .01 \times I(2)$, where $I(2)$ is the identity matrix of dimension 2, and $y^o_0 = y^c_0 = -4$.

The initial conditions of all other variables are set to 0. The correlation coefficient between the process (e_N) and (v_N), $N = 1, \ldots, 200$, is .05.

In Figure 6.1 the performance of the adaptive controller is illustrated by simultaneously plotting the reference input u_N^o, and the open and close loop outputs y_N^o, and y_N^c.

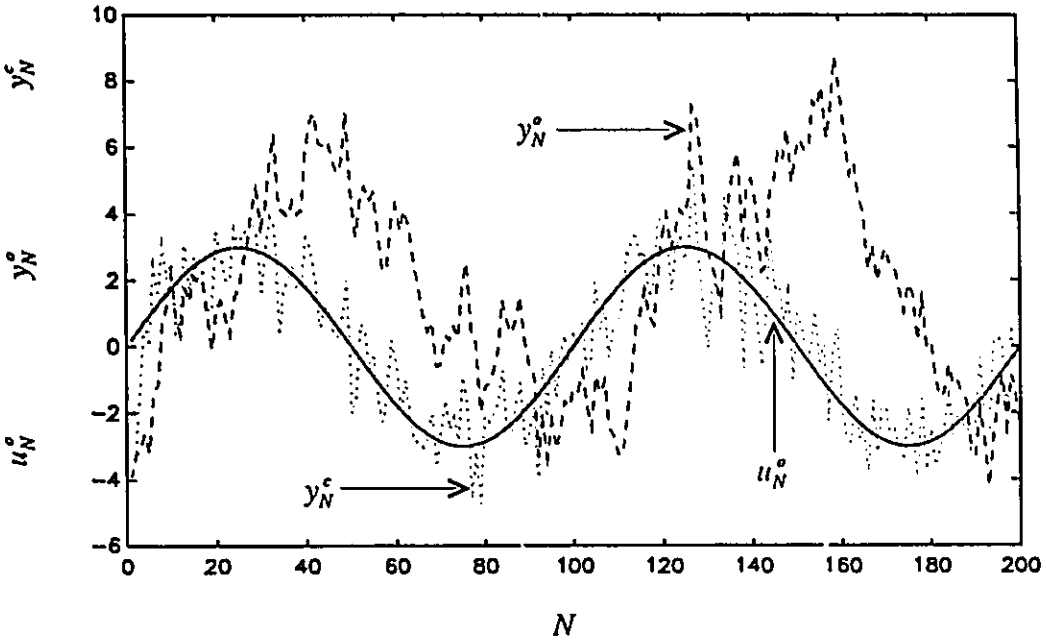
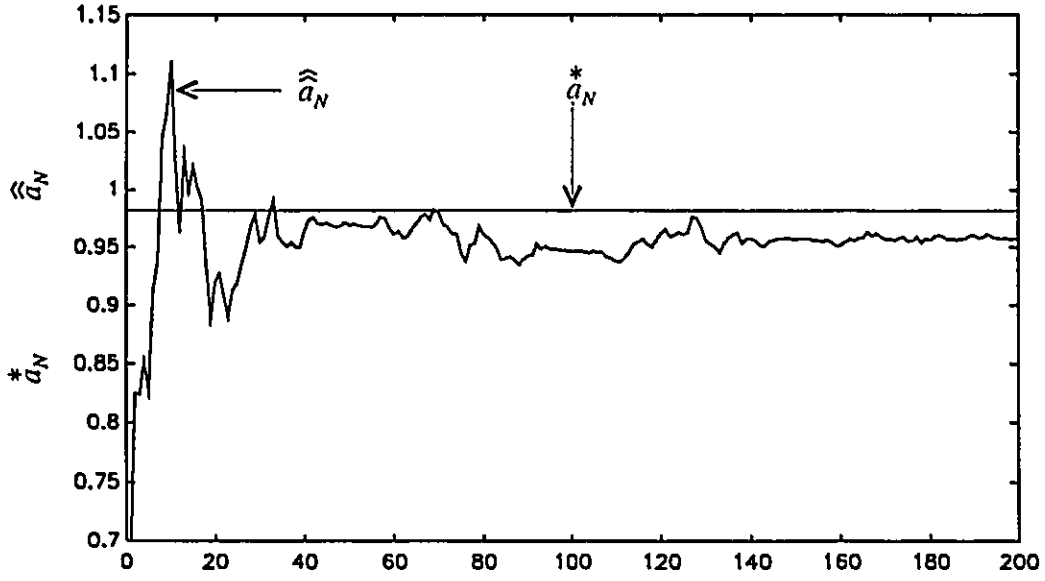


Figure 6.1: **The reference input u_N^o, and the open loop and close loop outputs y_N^o, and y_N^c respectively.**

In Figure 6.2 one can see that parameter estimates $\hat{\hat{a}}_N$ and $\hat{\hat{b}}_N$ give consistent estimates of a_N^*, and b_N^*, respectively.



(a)

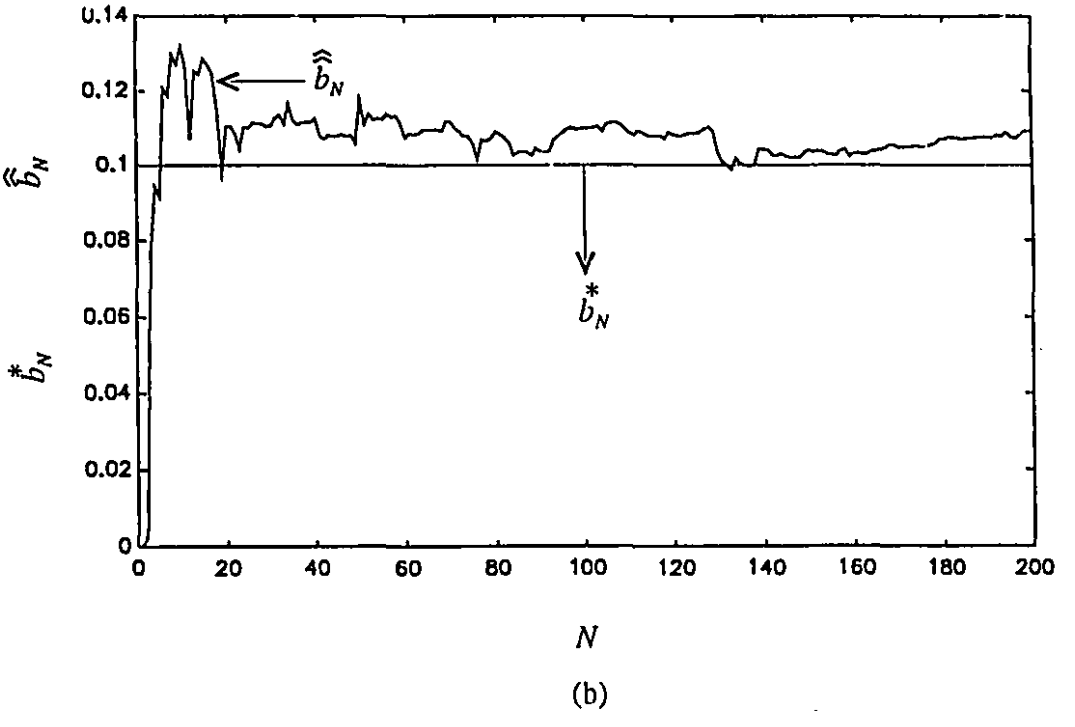


Figure 6.2: The true parameters and the parameter estimates of the ARX system 6.20: a) The autoregressive parameter a_N^*, and $\hat{\hat{a}}_N$; b) The exogenous parameter b_N^*, and $\hat{\hat{b}}_N$.

The gain of the adaptive control algorithm $k_N(\hat{\hat{\theta}}_{N-1})$ is shown in Figure 6.3.

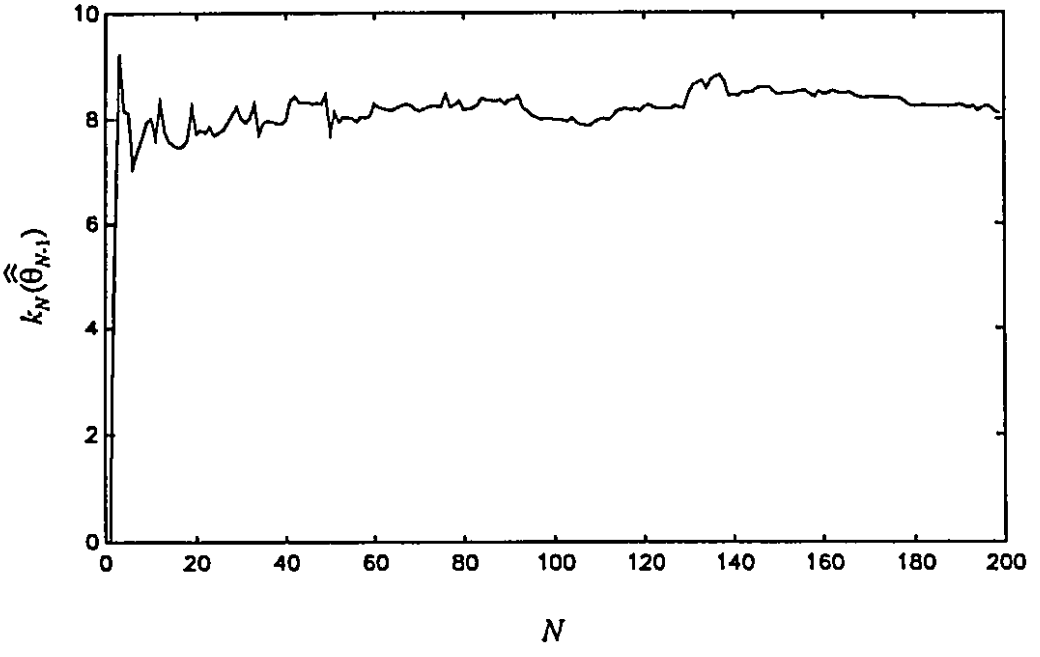


Figure 6.3: The gain of the adaptive controller.

6.3.2 The Effect of the Dither

In this section we will study, by means of a simulation, the effect of the dither on the overall adaptive control methodology. Here we make a slight modification to the simulation given in Section 6.3.1 by setting the reference input $u_N^c \equiv 0$ and the initial condition $y_0^o = y_0^c$.

The importance of the dither v_N in this case becomes very clear. Indeed, if the dither is extracted from the control law given in (6.21) then

$$x_{N+1}^c(\theta^*) = (a^* - b^* k(\theta))x_N^c(\theta^*) + e_{N+1},$$

which shows that the parameters a^* and b^* cannot be identified simultaneously.

In order to test the effect of the covariance of the dither, we ran three simulations with the dither covariance values: $\sigma_v^2 = 1 \times 10^{-4}$, $\sigma_v^2 = 9 \times 10^{-4}$, and $\sigma_v^2 = 25 \times 10^{-4}$.

In Figure 6.4 the reference input u_N^o, and the open loop and close loop outputs y_N^o, and y_N^c are plotted together for the value of $\sigma_v^2 = 25 \times 10^{-4}$.

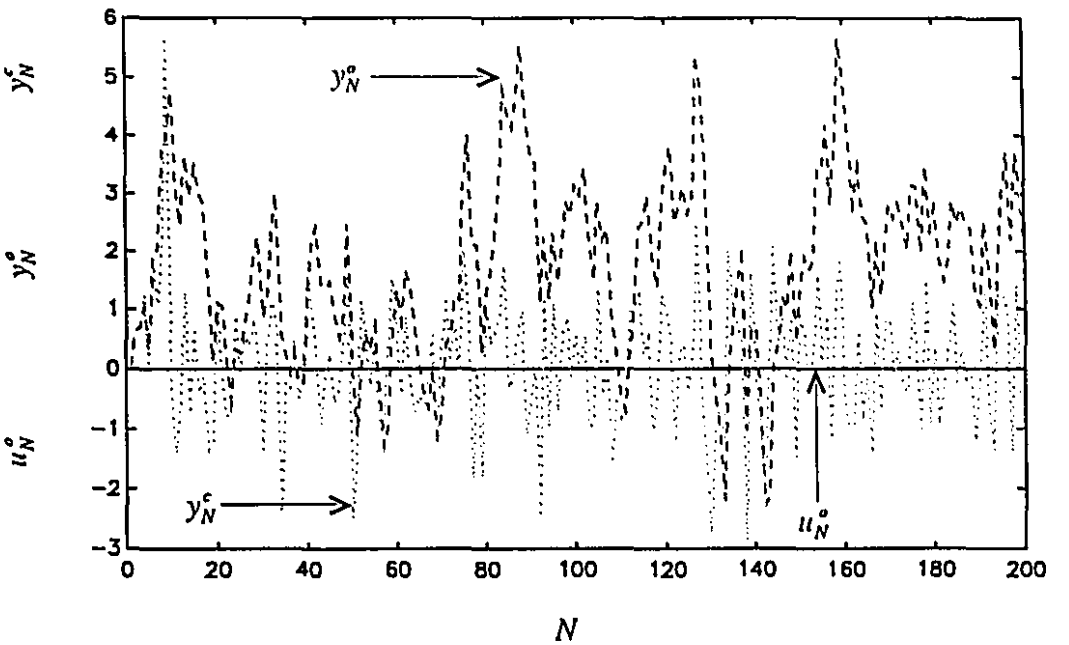
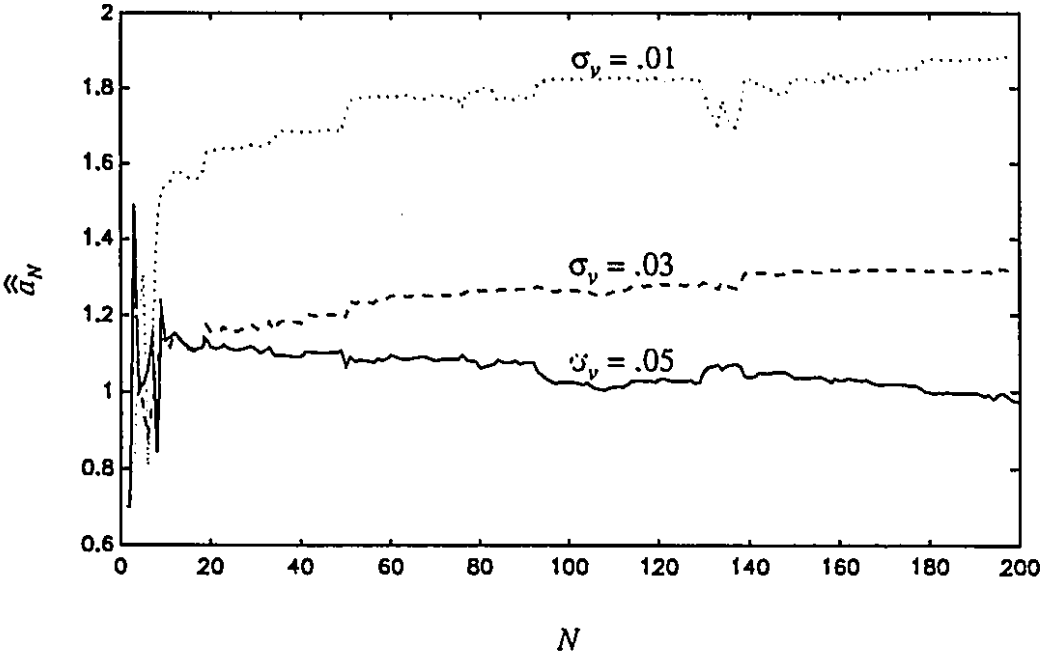


Figure 6.4: **The reference input u_N^o, and the open loop and close loop outputs y_N^o, and y_N^c for $\sigma_v^2 = 25 \times 10^{-4}$..**

Figure 6.5a and Figure 6.5b characterize the effect of the dither on the estimation of the parameters in (6.20). Observe the crucial role of the dither in consistently estimating the parameters of the system.


2
1.8
1.6
1.4
1.2
1
0.8
0.6
0
20
40
60
80
100
120
140
160
180
200
$\sigma_\nu = .01$
$\sigma_\nu = .03$
$\sigma_\nu = .05$
$\hat{a}_N$
N

(a)

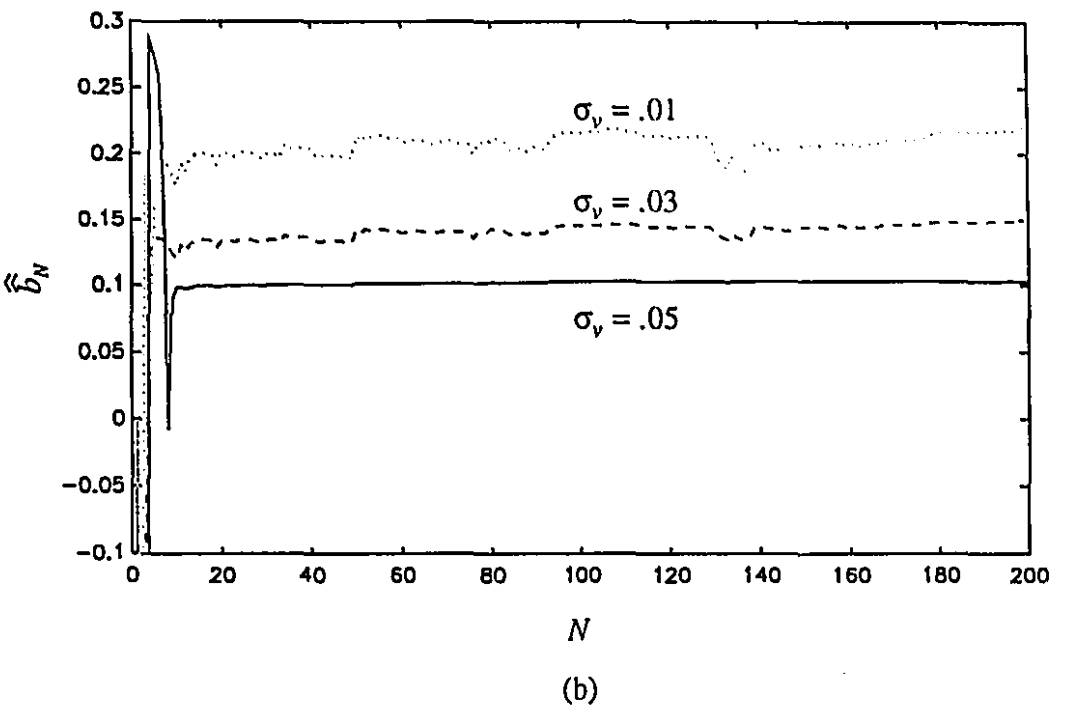


Figure 6.5: The true parameters and the parameter estimates of the ARX system (6.20): a) The autoregressive parameter a_N^*, and $\widehat{\widehat{a}}_N$; b) The exogenous parameter b_N^*, and $\widehat{\widehat{b}}_N$.

Chapter 7

Conclusion

The objective of this thesis has been to implement, test and refine model selection and change-point detection problems in real time using a form of predictive stochastic complexity. Moreover, in this dissertation we proved that the original form of the adaptive controller developed in [Ger90a] can be computed in a much less expensive manner.

In Chapter 4 we showed that predictive stochastic complexity is a mathematically well understood criterion which can be used to solve model selection problems in real time. A consistent method for finding the best model order for a set of data among certain classes of ARMA models of different order was validated by extensive computer experimentation. The use of fixed gain in the prediction error estimation procedure had the effect of increasing the qualitative performance of the algorithm; thus showing that the "badness" of the estimator increased the "badness" of over-parametrization. The model order simulations involving AR models illustrated this fact very clearly. A successful model order selection simulation involving ARMA models was also presented. Finally, a simulation indicated that in some cases, the predominant effect of parameter uncertainty over model order uncertainty can result in misleading answers about the order of the system for large values of the fixed gain

of the recursive prediction error algorithm.

In Chapter 5 a change-point detection method for ARMA systems was obtained, which assumes a slow and non-decaying drift after the change-point. Also, the abrupt jump parameter case, and change-point detection with undermodeling were considered. Some partial results on the analysis of the scheme were obtained, showing that the methods were amenable to theoretical analysis. The extensive simulations showed that the approach exhibited surprisingly good detection capabilities. Moreover, they illustrated the robustness of the change-point detection procedure with respect to the fixed gain of the prediction error algorithm. In addition, we showed—mainly empirically—that it is possible to improve the performance of the change-point detection when using undermodeling. This fact opens the way to further research since the issue of undermodeling in change-point detection has, to the best of our knowledge, not been previously studied. Finally the comparison of the stochastic complexity change point detection method to a "naive" procedure based on unprocessed parameter estimates showed that the former outperformed the latter.

What was left in the analysis of the change-point detection method was finding lower bound for the tracking error $(\epsilon_N^\lambda)^2 - (e_N)^2$ in terms of the rate of change $\dot{S}$. Nevertheless, Theorem 5.5.2 seems to be an important step towards obtaining an expression for the delay time. Also, the process $\left(\bar{\epsilon}_\tau(\widehat{\theta}^{\,\lambda}_{\tau-1}, \theta^*)\right)^2$ was obtained through a computationally intensive procedure. However there is hope to overcome this deficiency using Theorem 3.3.1.

In Chapter 6, the adaptive control problem for finite dimensional time invariant linear stochastic systems, as introduced in [Ger90a], was described. We proved that the original form of the adaptive controller as found in [Ger90a] can be computed in a much less expensive manner. The simulations of the adaptive control methodology for an ARX system, illustrated the stability and tracking capability of the adaptive controller. Moreover, the effect of the dither process on the closed loop performance

was illustrated. It was shown that in some cases the use of a dither process is necessary to guarantee identifiability. An open area of research is to use predictive stochastic complexity in the optimization of the performance of continually perturbed adaptive controllers.

Bibliography

[Abr63] N. Abramson. *Information Theory and Coding.* McGraw-Hill, New York, 1963.

[ACHS2] H. Z. An, Z.C. Chen, and E.J. Hannan. Autocorrelation, autoregression and autoregressive approximation. *Ann. Statist.*, 10:926–936, 1982.

[Aka70] H. Akaike. Statistical predictor identification. *Ann. Inst. Statis. Math.*, 22:202–217, 1970.

[Aka73] H. Akaike. Information theory and an extension of the maximum likelihood principle. In Petrov and Czaki, editors, *Proc. of the 2nd International Symposium on Information Theory*, pages 267–281, 1973.

[Alo87] A. Aloneftis. *Stochastic Adaptive Control: Results and Simulations.* Springer-Verlag, Berlin, 1987.

[And71] T. W. Anderson. *The Statistical Analysis of Time Series.* John Wiley, New York, 1971.

[ÅS74] K. J. Åström and T. Söderström. Uniqueness of the maximum-likelihood estimates of the parameters of an ARMA model. *IEEE Trans. Automat. Control*, AC-19:769–773, 1974.

[Åst83] K. J. Åström. Theory and applications of adaptive control–A survey. *Automatica*, 37(2):367–377, 1983.

[AW89] K. J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley, 1989.

[Bas81] M. Basseville. Edge detection using sequential methods or change in level-part II: Sequential detection of change in mean. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-29(1):32–50, 1981.

[Bas86] M. Basseville. Two examples of application of the GLR method in signal processing. In M. Basseville and A. Benveniste, editors, *Detection of Abrupt Changes in signals and Dynamical Systems*, volume 77. Springer, Berlin, 1986.

[Bas88] M. Basseville. Detecting changes in signals and systems—A survey. *Automatica*, 24(3):309–326, 1988.

[Bat62] J.A. Bather. Bayes procedures for deciding the sign of a normal mean. *Math. Proc. Cambridge Philos. Soc.*, 58:599–620, 1962.

[BB86] M. Basseville and A. Benveniste, editors. *Detection of Abrupt Changes in Signals and Dynamical Systems*, volume 77. Springer, Berlin, 1986.

[Bea71] R. V. Beard. Failure accommodation in linear systems through self-reorganization. Dept. MVT-71-1, Man Vehicle Laboratory, Cambridge, MA, 1971.

[BEG81] M. Basseville, B. Espiau, and J. Gasnier. Edge detection using sequential methods for change in level—part I: A sequential edge detection algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-29(1):24–31, 1981.

[BG90] J. Baikovicius and L. Gerencsér. Change point detection in a stochastic complexity framework. In *Proc. of the 29th IEEE CDC*, volume 6, pages 3554–3555, 1990.

[BG92a] J. Baikovicius and L. Gerencsér. Applications of stochastic complexity and related computational experiments. In *Proc. of the 31th IEEE CDC*, volume 4, pages 3311–3316, Tucson, 1992.

[BG92b] J. Baikovicius and L. Gerencsér. Change point detection as model selection. *Informatica*, 3(1):3–20, 1992.

[Bha87] P. K. Bhattacharyya. ML estimation of change-points in the distribution of independent random variables. *Journal of Multivariate Analysis*, 23:183–208, 1987.

[BJ68] G. Bhattacharyya and R. Johnson. Nonparametric tests for shift at an unknown time point. *Ann. Math. Statist.*, 39:1731–1743, 1968.

[Bro74] R. B. Broen. A nonlinear voter-estimator for redundant systems. In *Proc. 28-th IEEE Conference on Decision and Control*, pages 743–748, Phonix, Arizona, 1974.

[Cai88] P.E. Caines. *Linear Stochastic Systems.* John Wiley & Sons, 1988.

[Car88] A. E. Carlstein. Nonparametric change-point estimation. *Ann. Statist.*, 16:188–197, 1988.

[CG91] H. F. Chen and L. Guo. *Identification and Stochastic Adaptive Control.* Birkhäuser, Boston, 1991.

[Cik86] H. A. Cikanek. Space shuttle main engine failure detection. *IEEE Trans. Automat. Control*, AC-21:13–18, 1986.

[Dav65] L. D. Davisson. Prediction error of stationary gaussian time series of unknown covariance. *IEEE Trans. Inform. Theory*, IT-19:783-795, 1965.

[Dav68] Yu. A. Davydov. Convergence of distributions generated by stationary stochastic processes. *Theory of Probab. Appl*, 13:691-696, 1968.

[Daw84] A.P. Dawid. Present position and potential developments: Some personal views, statistical theory, the prequential approach. *J.R.S.S. A, Part 2*, 147:278–292, 1984.

[DDDW77] J. C. Deckert, M. N. Desai, J. J. Deyst, and A. S. Willsky. F-8 DFBW sensor failure identification using analytic redundancy. *IEEE Trans. Automat. Control*, AC-22(5):795–803, 1977.

[DH89] M. Davis and T. Hemerley. Order determination and adaptive control of ARX models using the PLS criterion. In K. Helmes and M. Kohlmann, editors, *Lecture Notes in Control and Information Sciences*, pages 91–101. Proc. of the 4-th Bad Honnef Conference on Stochastic Differential Systems, Springer-Verlag, 1989.

[DP86] J. Deshayes and D. Picard. Off-line statistical analysis of change-point models using non parametric and likelihood methods. In M. Basseville and A. Benveniste, editors, *Detection of Abrupt Changes in signals and Dynamical Systems*, volume 77. Springer, Berlin, 1986.

[DW77] L. P. Devroye and G. L. Wise. Nonparametric detection of changes in system characteristics. In *Proc. on Twentieth Midwest Symposium on Circuits and Systems*, 1977.

[Fed75] P. I. Feder. On asymptotic distribution theory in segmented regression problems–identified case. *Ann. Statist.*, 3(1):49–83, 1975.

[FG82] B. Friedland and S. M. Grabousky. Estimating sudden changes of biases in linear dynamic systems. *IEEE Trans. Automat. Control*, AC-27(1):237–140, 1982.

[Fra90] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—A survey of new results. *Automatica*, 26:459–474, 1990.

[Gar69] L. A. Garden. On detecting changes in the mean of normal variates. *Ann. Math. Statist.*, 40(1):116–126, 1969.

[GB90] L. Gerencsér and J. Baikovicius. Model selection, stochastic complexity and badness amplification. In *Proc. of the 30th IEEE CDC*, volume 2, pages 1999–2004, 1990.

[GB91] L. Gerencsér and J. Baikovicius. Change-point detection using stochastic complexity. In *Identification and System Parameter Estimation*, pages 395–400, Budapest, 1991. *9th IFAC/IFORS Symposium.*

[Ger86] L. Gerencsér. Parameter tracking of time-varying continuous-time linear stochastic systems. In Ch. I. Byrnes and A. Lindquist, editors, *Modelling, Identification and Robust Control*, pages 581–595. North Holland, 1986.

[Ger88a] L. Gerencsér. On Rissanen's predictive stochastic complexity for stationary ARMA processes. *Submitted to the Annals of Statistics*, 1988.

[Ger88b] L. Gerencsér. Rate of convergence of a continuous-time stochastic approximation method. *Submitted to Stochastic Processes and Their Applications*, 1988.

[Ger89a] L. Gerencsér. Almost sure asymptotics of Rissanen's predictive stochastic complexity. In *In Proc. of the International Symposium on the Mathematical Theory of Networks and Systems*, Amsterdam, 1989.

[Ger89b] L. Gerencsér. Almost sure exponential stability of random linear differential equations. *McGill Research Center for Computer Vision and Intelligent Machines. Submitted for publication*, (TR-CIM-89-7), 1989.

[Ger89c] L. Gerencsér. On a class of mixing processes. *Stochastics*, 26:165–191, 1989.

[Ger89d] L. Gerencsér. On Rissanen's predictive stochastic complexity for stationary ARMA processes. *McGill Research Center for Intelligent Machines, TR-CIM-89-5*, 1989. Submitted to *The Annals of Statistics*.

[Ger89e] L. Gerencsér. On the normal approximation of the maximum likelihood estimator of ARMA parameters. In *Proc. of the 28th. IEEE CDC*, volume 1, Tampa, Florida, 1989.

[Ger89f] L. Gerencsér. Rate of convergence for Ljung's scheme. *Submitted to Stochastic Processes and Their Applications*, (TR-CIM-89-16), 1989.

[Ger90a] L. Gerencsér. Closed loop parameter identifiability and adaptive control of a linear stochastic system. *System & Control Letters*, 15:411–416, 1990.

[Ger90b] L. Gerencsér. On the martingale approximation of the estimation error of ARMA parameters. *System & Control Letters*, 15:417–423, 1990.

[Ger91a] L. Gerencsér. Fixed gain estimators of ARMA parameters. Submitted for publication, 1991.

[Ger91b] L. Gerencsér. The law of the cubic root. In *Identification and System Parameter Estimation*, volume 1, pages 63–65, Budapest, 1991. *9th IFAC/IFORS Symposium.*

[Ger91c] L. Gerencsér. Strong approximation results in estimation and adaptive control. In L. Gerencsér and P. Caines, editors, *Topics in Stochastic Systems: Modelling, Estimation and Adaptive Control*, pages 268–299. Lecture Notes in Control and Information Sciences, Springer-Verlag, 1991.

[Ger91d] L. Gerencsér. Strong approximation theorems for estimator processes in continuous time. In I. Berker, E. Csáki, and I. Rérész, editors, *Limit Theorems in Probability and Statistics.* Colloquia Mathematica Societatis LÁros Bolyai, North Holland, 1991. To appear.

[Ger92a] L. Gerencsér. AR(∞) estimation and nonparametric stochastic complexity. *IEEE Trans. Inform. Theory*, IT-38(6), 1992.

[Ger92b] L. Gerencsér. Predictive stochastic complexity associated with fixed gain estimators. Submitted for publication, 1992.

[GR86] L. Gerencsér and J. Rissanen. A prediction bound for gaussian ARMA processes. In *Proc. of the 25th. CDC*, volume 3, pages 1487–1490, Athens, 1986.

[GR91] L. Gerencsér and J. Rissanen. Asymptotics of predictive stochastic complexity: from parametric to nonparametric models. In E. Parzen, D. Brillinger, M. Rosenblatt, M Taqqu, J. Geweke, and C. Caines, editors, *New directions in time-series analysis.* Institute of Mathematics and its Applications, Minneapolis, 1991.

[GS84] G. C. Goodwin and K. S. Sin. *Adaptive Filtering Prediction and Control.* Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[GWW+78] D.E. Gustavson, A.S. Willsky, J. Wang, M.C. Lancaster, and J.H. Triebwasser. ECG/VCG rhythm diagnosis using statistical signal analysis. Part I: Identification of persistent rhythms. *IEEE Trans. Biomedical Engrg.*, BME-25(4):344–353, 1978.

[Han73] E. J. Hannan. The asymptotic theory of linear time series models. *J. Appl. Probab.*, 10:130–145, 1973.

[HD88] E. J. Hannan and M. Deistler. *The Stastical Theory of Linear Systems.* John Wiley & Sons, New York, 1988.

[HD89] E. Hemerley and M. Davis. Strong consistency of the PLS criterion for order determination of autoregressive processes. *Ann. Statist.*, 17:941–946, 1989.

[Hin70] D. Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3):509–523, 1970.

[HMP89] E. J. Hannan, A. J. McDougall, and D. S. Poskitt. Recursive estimation of autoregressions. *J. Roy, Stat. Soc*, 51, 1989. To appear.

[IH81] I. A. Ibragimov and R. Z. Hasminskii. *Statistical Estimation. Asymptotic Theory.* Springer-Verlag, Berlin, 1981.

[IK83] P. A. Ioannou and P. V. Kokotovic. *Adaptive Systems with Reduced Models.* Springer-Verlag, New York, 1983.

[IL71] I. A. Ibragimov and Yu. A. Linnik. Independent and stationary sequences of random variables. *Groningen*, 1971.

[Ise84] R. Iserman. Process fault detection based on modelling and estimation methods–A survey. *Automatica*, 20:387–404, 1984.

[JJS87] B. James, K. L. James, and D. Siegmund. Test for a change point. *Biometrika*, 74(1):71–83, 1987.

[Jon73] H. L. Jones. *Failure detection in linear systems.* PhD thesis, MIT, Cambridge MA, 1973.

[KA87] A. N. Kolmogorov and Upsenskii V. A. Algorithms and randomness. *Theoria Veroyatnostey i ee Primeneniya (Theory of Probabilities and its Applications)*, 32-33:425–455, 1987.

[Kab88] P. Kabaila. On Rissanen's lower bound on the accumulated prediction error. *J. Time Ser. Anal.*, 1988.

[Kai80] T. Kailath. *Linear Systems.* Prentice Hall, Englewood Cliffs, New Jersey, 1980.

[Kal58] R. E. Kalman. Design of a self-optimizing control system. *Trans. ASME Ser. E J. Appl. Mech.*, 80:468–478, 1958.

[KV86] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, identification, and adaptive control.* Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[Lan79] Y. D. Landau. *Adaptive Control–The Model Reference Approach.* Marcel Dekker, New York, 1979.

[Lev73] L. Levin. On the notion of a random sequence. *Soviet Math. Doklady*, 14:1413–1416, 1973.

[LJ74] A. Lempel and Ziv J. On the complexity of finite sequence. *IEEE Trans. Inform. Theory*, IT-22:75–81, 1974.

[Lju76] L. Ljung. On consistency and identifiability. *Mathematical Programming Study*, 5:169–190, 1976.

[Lju87] L. Ljung. *Theory for the user.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[Lom89] F. Lombard. Some recent developments in the analysis of changepoint data. *South African Statistical J.*, pages 1–21, 1989.

[Lor71] G. Lorden. Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42:1897–1908, 1971.

[LS84] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* MIT Press, Cambridge, Mass., 1984.

[LZ86] H. Linhart and W. Zucchini. *Model Selection.* John Wiley & Sons, 1986.

[Mac74] I.B. Macneill. Test for change of parameters at unknown times and distributions of some related functionlas on brownian motion. *Ann. Statist.*, 2:950–962, 1974.

[Mir80] L. A. Mironovski. Functional diagnosis of dynamic systems–A survey. *Automat. Remote Control*, 41:1122–1143, 1980.

[ML74] P. Martin-Lov. The notion of redundancy and its use as a quantitative measure of the discrepancy between a statistical hypothesis and a set of observational data. *Sc. J. Statistics*, 1:13–18, 1974.

[MP71] R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 12:637–6, 1971.

[Nik91] I. V. Nikiforov. Sequential detection of changes in stochastic processes. In *Identification and System Parameter Estimation*, pages 11–18, Budapest, 1991. 9th IFAC/IFORS Symposium.

[NM80] K. S. Narendra and R. V. Monopoli. *Applications of Adaptive Control.* Academic Press, New York, 1980.

[Pag54] E.S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[Par66] P. C. Parks. Liapunov redesign of model reference adaptive control systems. *IEEE Trans. Automat. Control*, AC-11:362–367, 1966.

[PD90] T. D. Popescu and S. Demetriu. Detecting changes in statistical models with application to seismic signal processing. In Utkin V. and Jaaksoo Ü, editors, *11th IFAC World Congress*, volume 3, pages 86–91, Tallinn, Estonia, 1990.

[Pet79] A. N. Pettitt. A non parametric approach to the change point problem. *Appl. Stat.*, 28:126–135, 1979.

[PFCS9] R. J. Patton, P. M. Frank, and R. N. Clark, editors. *Fault diagnosis in Dynamics Systems: Theory and Applications.* Prentice Hall International, UK or NL, 1989.

[Pic85] A.N. Picard. Testing and estimating change-points in time series. *Adv. in Appl. Probab.*, 17:841–867, 1985.

[QRS9] J.R. Quinlan and Rivest R.L. Inferring decision trees using minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[RC79] J. Rissanen and P. E. Caines. The strong consistency of maximum likelihood estimators for ARMA processes. *Ann. Statist.*, 7:297–315, 1979.

[Ris78] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[Ris84] J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Trans. Inform. Theory*, 30:629–636, 1984.

[Ris86] J. Rissanen. Stochastic complexity and modeling. *Ann. Statist.*, 14:1080–1100, 1986.

[Ris87] J. Rissanen. Stochastic complexity with discussions. *J. of the Royal Statistical Society*, 49(3), 1987.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publisher, 1989.

[RS73] V. G. Robinson and D. D. Sworder. Feedback regulators for jump parameter systems with state and control dependent transition states. *IEEE Transactions on Automatic Control*, AC-18:355–360, 1973.

[Saw78] T. Sawa. Information criteria for discriminating among alternative models. *Econometrica*, 46:1273–1291, 1978.

[SB89] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6:461–464, 1978.

[Shi61] A. Shiryaev. The problem of the most rapid detection of a disturbance in a stationary process. *Soviet Math. Dokl.*, 2:795–799, 1961.

[Shi63] A. Shiryaev. The problem of the most rapid detection of a disturbance in a stationary process. *Theory Probab. Appl.*, 8:22–46, 1963.

[Shi78] A. Shiryaev. *Optimal Stopping Rules.* Springer-Verlag, New York, 1978.

[Shi80] R. Shibata. Selection of the number of regression variables;A mini-max choice of generalized FPE. *The Ann. Stat.*, 38(3):459–474, 1980.

[Smi75] A.F.M. Smith. A Bayesian approach to inference about a changee-point in a sequence of random variables. *Biometrika*, 62(2):407–416, 1975.

[SS89] T. Söderström and P. Stoica. *System Identification.* Prentice Hall, 1989.

[Sto74] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. Royal Stat. Soc., Ser. B*, 36:111–147, 1974.

[Sto77] M. Stone. An asymptotics equivalence of choice of model by cross-validation and Akaike's criterion. *J. Royal Stat. Soc., Ser. B*, 39:44–47, 1977.

[SW86] M.S. Srivastava and K.J. Worsley. Likelihood ratio tests for a change in the multivariate normal mean. *J. Amer. Statist. Assoc.*, 81(393):199–204, 1986.

[Tay68] H. M. Taylor. The economic design of cumulative sum control charts. *Technometrics*, 10(3):479–488, 1968.

[Tel86] L. Telksnys, editor. *Detection of Changes in Random Processes.* Translations Series in mathematics and Engineering. Optimization Software, Inc., New York, 1986.

[WB68] C. S. Wallace and D. M. Boulton. An information measure for classification. *Computing Journal*, 1:185–195, 1968.

[Wei87] C. Z. Wei. Adaptive prediction by least squares predictors in stochastic regression models with application to time series. *Ann. Statist.*, 15:1667-1687, 1987.

[WES+80] A.S. Willsky, Chow E.Y., Gershwin S.B., Greene C.S., Houpt P.K., and Kurkjian A. L. Dynamic model-based techniques for the detection of incidents on freeways. *IEEE Trans. Automat. Control*, AC-25(3):347-360, 1980.

[WF90] J. Wünnenberg and P. Frank. Dynamic model based incipient fault detection concept for robots. volume 3, pages 76-81, Tallinn, Estonia, 1990. 11th IFAC World Congress.

[Wil76] A. S. Willsky. A survey of design methods for failure detection in dynamics systems. *Automatica*, 12:601-611, 1976.

[WJ76] A.S. Willsky and H.L. Jones. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Trans. Automat. Control*, 21:108-112, 1976.

[ZL70] A. K. Zhvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25:83-124, 1970.

[ZL78] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate. *IEEE Trans. Inform. Theory*, IT-24:530-536, 1978.

[ZW91] G. Zames and L. Y. Wang. Local-global double algebras for slow H^∞ adaptation: Part I-inversion and stability. *IEEE Trans. Automat. Control*, 36:130-142, 1991.