THESIS REPORT

Ph.D.

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

Fast Recursive Estimation of System Order and Parameters for Adaptive Control and IIR Filtering

by J-Q. Pan

Advisor: W.S. Levine

Ph.D. 91-10

Fast Recursive Estimation of System Order and Parameters for Adaptive Control and IIR Filtering

By

Ji-Qin Pan

Dissertation submitted to the Faculty of The Graduate School of
The University of Maryland in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
1991

Advisory Committee:

Professor William S. Levine, Chairman/Advisor
Professor P. S. Krishnaprasad
Professor Thomas J. McAvoy
Associate Professor Eyad Abed
Associate Professor Prakash Narayan

ABSTRACT

Title of Dissertation: Fast Recursive Estimation of System Order and Parameters for Adaptive Control and IIR Filtering

Ji-Qin Pan, Doctor of Philosophy, 1991

Thesis directed by: Willim S. Levine, Professor
Department of Electrical Engineering

In this dissertation, simultaneous on-line estimation of system order and parameters is studied. The key features are the direct exploitation of the Toeplitz structure in a Toeplitz submatrix system (of linear equations), the theoretical martingale analysis and systematic simulation study of estimation of ARX system order and parameters, and the stability study of IIR filters.

The fundmental Levinson-Durbin algorithm is generalized and consequently a similar fast algorithm is developed for solving Toeplitz submatrix systems. The algorithm is then applied to signal processing and modeling of time series, including a lattice form of LMMSE IIR filters, a fast time and order recursive algorithm (TORA) for determining parameter estimates for a family of ARX models, and a fast method for simultaneous estimation of ARX system order and parameters. The TORA converges to the LS algorithm provided the time series involved are uniformly bounded. The strong consistency of the TORA and proposed order estimation method is shown under some conditions, which are applicable to adaptive control and IIR signal processing. The key factors in the consistency and convergence rate are explained through several examples. Finally, a sufficient condition on instantaneous stability for TORA all-pole filters is established and then an implementable stabilizing algorithm is suggested for general SISO adaptive filters, which does not need prior knowledge of the system that generates the data being processed.

Dedication

To my devoted wife Yan
for her love, understanding, and support

Acknowledgements

I would like to first express my sincerest gratitude to my advisor Prof. W.S. Levine. He has provided me with superlative guidance, continuous encouragement, and unrivaled consideration. I admire his comprehensive insight and emphasis on both theoretical research and engineering applications. His persistence and serious attitude to research have been and will continuously be a reference model for me. I am also grateful to Prof. E. Abed, Prof. P.S. Krishnaprasad, Prof. T.J. McAvoy, and Prof. P. Narayan for their serving on my committee and giving me much useful advice.

I owe a significant debt of gratitude to Prof. C.Z. Wei for enlighting instructions in martingale convergence theory. I am greatly indebted to Prof. K.J. Ray Liu for his various invaluable help, including introducing me to the subject of adaptive signal processing. Also, I wish to thank my former advisor and mentor Prof. Y. Zhang, at Beijing Institute of Technology, China, for bringing me into scientific research.

In the course of this work, I have profited greatly from discussions with many other individuals, especially, Prof. M.A. Austin, Prof. W. Dayawansa, Prof. T.E. Fuja, Prof. A. Tits, Prof. E. Zafiriou, Prof. G.M. Zhang, at Maryland, Prof. H.-F. Chen at the Chinese Academy of Sciences, Prof. J. Doyle at the Caltech, Prof. M.J. Grimble at the University of Strathclyde, Prof. G. Stein at MIT, and Prof. Y.Z. Tsypkin at the USSR Academy of Sciences.

I would like to thank Mr. X.L. Chen, Mr. L. Kuang, Dr. D.C. Liaw, Dr. C.C. Liu, Mr. D.P. Tsakiris, Dr. N.-K. Tsing, Dr. L.S. Wang, Mr. R. Yang, Mr. J. Zhou, and all my group mates and colleagues in the CACSE Lab for helpful discussions and assistance; Dr. J. He at Jefferson Medical College for his friendship; Dr. M. Jodorkovsky for helping me in making figures; Mr. J. Reilly for assistance in using Matlab and in the typesetting of this thesis. In addition, I acknowledge the generous Fellowship of the Systems Research Center at the University of Maryland.

This research was supported by the National Science Foundation's Engineering Research Centers Program: NFSD CDR 8803012.

I would also express my deep thankfulness to my grandmother and my parents for their ever present support as well as for instilling within me the desire to make unremitting efforts to improve myself.

Last but certainly far from being least, I would like to thank a special person in my life, my wife Yan. She has devotedly given me her constant love, continuous encouragement, and thoughtful loving support. With her love, caring, and understanding, she stood firmly by me through difficulties. She has truly been an inspiration to me.

Jiqin Pan

August 1991, College Park, Maryland

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Adaptive control is an important and effective means for dealing with the control of slowly-varying uncertain systems, such as chemical processes and homing flight systems. Adaptive IIR filtering is a new and promising technique for achieving high performance of adaptive signal processors, which has found increasingly wide applications from telephone networks to radar. IIR filters and a large class of uncertain systems can be described by using autoregressive models with exogenous inputs (ARX models). The models are specified by both the model order and parameters. A rational approach for designing an adaptive controller or IIR filter is to first identify a model based on available information on the uncertain system or unknown signals, followed by solving an underlying design problem for the identified model. Consequently, simultaneous estimation of system order and parameters is a crucial component of adaptive control and IIR filtering problems, which often involves an extensive amount of computation. Only with fast recursive algorithms, can such a controller or filter be implemented on-line.

The research conducted in this dissertation is two-fold. The dissertation is part of on-going research [29][72] to understand system identification for ARX systems with martingale noise, including consistency of parameter and order estimation, convergence rate, and stability, etc.. IIR filters and linear quadratic adaptive control systems present examples of the systems. The dissertation is also a generalization of the Levinson-Durbin algorithm[66] from Toeplitz systems to Toeplitz-submatrix systems, or from pure autoregressive models to ARX models. Model reduction of linear discrete systems, least squares parameter estimation of ARX and ARMAX systems[1], and on-line estimation of ARX system order and parameters are instances in which Toeplitz-submatrix systems are important.

Order estimation has been under study for more than 20 years [1][40], during which many rigorous mathematical results have been established due to developments in statistics, stochastic processes, and information theory. On the other hand, applications in various engineering fields have revealed that the model order plays a crucial role, especially in control and signal processing, which illustrates the fundamentality of the problem of order estimation. Currently, there exists a big gap between the theoretical results of order estimation and engineering requirements, which prevents order estimation from being applied. Two of the major difficulties are understanding and implementing of the concepts. For instance, it is believed that the degree of complexity of a model should be determined by the intended use of the model. On the contrary, a conventional assumption in order estimation is that the model order is unique for a system to be modeled. Incorporating with the following chapters, Chapter 2 serves as an attempt to narrow the gap by considering the problem of order estimation from the point of view of adaptive control and adaptive signal processing. What kind of roles the order estimation could play is illustrated through analysis of several examples. Theoretical results are also summarized.

In the process of order estimation, the acquisition of a family of optimal models is often required so that an application-oriented criterion, and perhaps other prior information, can be applied to model order selection. This task is computationally prohibitive on-line unless some fast algorithms are developed.

[1]ARMAX systems represent autoregressive moving-average systems with exogenous inputs.

The problem of determining the optimal models for an ARX system in the sense of linear minimum mean-square error (LMMSE) leads ultimately to solving a block-Toeplitz submatrix system of linear equations. Chapter 3 develops an order recursive algorithm (ORA) for fast solution of block-Toeplitz submatrix systems through directly exploiting the Toeplitz structure of each submatrix. The algorithm can be made highly parallel. The key feature of the algorithm is that the solution is determined as the dimension of the diagonal submatrices successively expands and all the solutions corresponding to different dimension are generated. This fundamental approach provides us with an efficient tool for solving LMMSE estimation problems for a family of ARX models and a lattice form of LMMSE IIR filters. It also sheds some light on a general approach for solving Toeplitz-like systems without the need for any matrix factorization.

In applications of adaptive control and adaptive signal processing, engineers often have a time series of measurements. Often, these are not realizations of any stationary and ergodic processes and no information on the statistics of the measurements is available. In this circumstance, least-squares (error) models become the only available models. Chapter 4 suggests a time and order recursive algorithm (TORA) for approximating the parameters of an LS model by the solutions of a time series of Toeplitz submatrix systems. The algorithm preserves all the nice features of the ORA in Chapter 3 and updates the Toeplitz submatrix systems in a time-recursive way. Also, the algorithm treats measurements of system outputs and inputs symmetrically, which is useful when system actuators have imperfections or when system input measurements are contaminated with noise. The algebraic analysis of the approximation errors is conducted in a straightforward manner, which shows the convergence of the approximation error to zero and the convergence rate.

The main performance requirement of order and parameter estimation is consistency and fast convergence rate. The current consistent algorithms for ARX systems are faced with computational difficults in practice and the essence of consistency analysis of the algorithms is covered by complicated and tricky mathematical manipulation. A fast method for estimation of ARX system order and parameters is proposed in Chapter 5 by incorporating the TORA into the accumulated prediction error criterion. Based on the previous results [23][72], the strong consistency of the proposed algorithm is proved through martingale analysis. During the analysis of the proposed algorithm, a unified approach is shown for studying the convergence and convergence rate of LS parameter estimation. The effect of characteristics of ARX systems on the consistency of order estimation and convergence and convergence rate of parameter estimation are systematically studied and illustrated with five examples so that both knowledge and understanding of order estimation are enhanced.

An IIR filter can achieve significantly better performance than an FIR filter having the same number of coefficients. However, an IIR filter is not unconditionally stable unless some restrictions are placed on the denominator parameters of the filter. In adaptive IIR signal processing, it is customary to use the restriction that all the poles of the denominator are within the unit circle at each time instant [134], which we call instantaneous stability. Following this custom, a sufficient condition for instantaneous stability of all-pole filters is established in Chapter 6. Then, an implementable stabilizing algorithm is developed for general adaptive IIR filters without requiring stability monitoring. In addition, all the stability results are robust in the sense that the true model structure of the system which generates the data is not necessarily known.

This dissertation is concluded in Chapter 7 with highlights of the established research results and some suggestions for future research are outlined.

Chapter 2

Background

System identification deals with the problem of how to build mathematical models of dynamical systems based on observed data, and perhaps other (a priori) information, from the systems. Broadly speaking, a dynamical system is an object in which the observed output depends on interactions between internal variables of the system and both present and previous external stimuli. Aircraft, manufacturing plants, chemical processing plants, robotics, telephone networks, seismographs, and stock markets are a few examples of dynamical systems. When people deal with a system, they need some knowledge of how internal variables interact with each other and produce output. As pointed out by Ljung [102], any *assumed* relationship among system variables is a model of the system. When the relationship is described in terms of mathematical expressions such as difference or differential equations, it becomes a mathematical model. In all fields of engineering and physics, mathematical methods are used to make good designs, to improve products in quality, and to raise efficiency of manufacturing. As instrumental tools, they are also used in some areas of physical science, e.g., geophysics [127] and computer science, and in some other sciences such as economics [79], ecology, and biology. Consequently, system identification has wide areas of applications.

Mathematical models of a system may be constructed by relying on "laws of nature" governing the system, e.g., Newton's law and Ohm's law. Satisfactory models cannot, however, be derived in this way in many situations. These cases could be (*i*) the "laws of nature" governing the system are not well understood, (*ii*) it is technically impossible to get a complete model of the system, or (*iii*) it is inefficient and uneconomical to construct the relationship between internal variables by applying "laws of nature" to the system. Cases (*i*) and (*ii*) often occur in economics [79], meteorology, geophysics [127], and seismology. Situations (*ii*) and (*iii*) can be met in flight systems, chemical processes [116], and the manufacturing industry. In robotics, communication networks, and signal processing, the most frequent and difficult situation is that the characteristics of the systems involved are unknown or partially unknown. As an alternative and complementary method of building mathematical models, system identification provides a methodology for inferring mathematical models via analysis of input and output signals from systems. This feature makes system identification interesting and even important to many fields, especially in control, communication, and signal processing [67][102].

Mathematical models of a system may involve various types of equations (which may be time-continuous or time-discrete, deterministic or stochastic, linear or nonlinear, etc..) and varying degree of complexity. The type and degree of complexity of a model should be determined by the intended use of the model so that a mathematical model is always purposeful. Our primary aim in this dissertation is to study system identification of linear systems for adaptive control and adaptive IIR (for infinite impulse response) filtering, with the emphasis on simultaneous estimation of system order and parameters and fast computation. System identification of linear systems has been under study for a long time [102][138]. But the subject was largely oriented toward the parameter estimation of linear systems, with less attention to order estimation and its engineering significance such as in applications to adaptive control and adaptive signal processing. This background chapter serves as a review of previous results which are strongly related to the subject we are studying in this dissertation.

Some other previous results will be mentioned when it is necessary. The detailed table of contents provides an indication of the scope of the organization of this long chapter. Not only are many early results collected in this chapter, but a lot of exposition of mathematical results is added to explain important ideas. While doing this, we also gradually build up awareness of the motivation of conducting our research on the subject, the significance of the results obtained in the dissertation and their potential applications and possible generalizations.

2.1 System Identification for Control System Design

2.1.1 Parametric and Unparametric Uncertainties

In a large number of control system design problems, an accurate model of the plant to be controlled is not available to designers. This can be due to either high complexity of the plant, or poor understanding of the plant dynamics. Sometimes, even if an actual plant model is available, designers prefer using a less complex model to design a simple controller. As a result, model uncertainties have become an important issue in control system design.

In a framework similar to Doyle and Stein [44], an *actual* model of a linear time discrete system is of the from:

$$\begin{aligned} y_n &= G(z^{-1})u_n + v_n \\ v_n &= H(z^{-1})w_n \end{aligned} \tag{2.1}$$

where $G(z^{-1})$ and $H(z^{-1})$ are the actual transfer function models of system dynamics and system disturbance. z^{-1} denotes the back-shift operator: $z^{-1}y_n = y_{n-1}$. The model input, output, and disturbance are, respectively, denoted by u_n, y_n and v_n. The index n in Eq.(2.1) represents the iteration number. The product of sampling time T and iteration number n is equal to the time instant t, starting from zero. The disturbance is presented as the output of a linear time-invariant system with random input signal w_n. The system dynamics $G(z^{-1})$ is of the form:

$$G(z^{-1}) = G_\theta(z^{-1}) + \bar{G}_a(z^{-1}) \tag{2.2}$$

or

$$G(z^{-1}) = G_\theta(z^{-1})(1 + \bar{G}_m(z^{-1})) \tag{2.3}$$

where $G_\theta(z^{-1})$ is a parametric transfer function. The values of the parameters (components of θ) are not necessarily known or exactly known. This implies that the plant model has parametric uncertainties. $\bar{G}_a(z^{-1})$ and $\bar{G}_m(z^{-1})$ are referred to, respectively, as additive unstructured and multiplicative unstructured uncertainties. Similarly, the disturbance model is presented in the form:

$$H(z^{-1}) = H_\theta(z^{-1}) + \bar{H}_a(z^{-1}) \tag{2.4}$$

or

$$H(z^{-1}) = H_\theta(z^{-1})(1 + \bar{H}_m(z^{-1})) \tag{2.5}$$

where $H_\theta(z^{-1})$ is a parametric transfer function of the disturbance model. The transfer functions, $G_\theta(z^{-1})$ and $H_\theta(z^{-1})$, are also called a nominal model of the system, which is supposed to describe the system (2.1) within a frequency range of primary interest. A detailed discussion on the description of model uncertainties and their physical meaning can be found in [44].

A robust (non-adaptive) controller may be designed to control the uncertain system (2.1). That is, a *time-invariant* control structure and parameters are chosen so as to at least preserve stability and perhaps reduce sensitivity of the actual closed-loop transfer function to the parametric/unstructured uncertainties [19]. When the performance achieved by robust control is not satisfactory, it is worthwhile to consider adaptive control. Rather than postulate a transfer function of the plant, adaptive control includes a procedure to identify the parameters of G_θ and H_θ during the plant continues to operate. The aim is to reduce the parametric uncertainty. The resulting transfer functions $G_{\hat{\theta}}$ and $H_{\hat{\theta}}$ are used as a basis for control design.

2.1.2 Adaptive Control

There are mainly two approaches to implement the concept of adaptive control: self-tuning control and model reference adaptive control [11][132]. In self-tuning control, an identification or parameter estimation mechanism plays a crucial role. Self-tuning control can be further divided into two classes: direct and indirect self-tuning control. In direct self-tuning control, controller parameters are estimated directly based on the plant output and input measurements. In indirectly self-tuning control, a model of the plant to be controlled is identified first based on measurement data and then controller parameters are generated as a solution of an underlying design problem, where the identified model represents the underlying plant. Model reference adaptive control utilizes a different approach. Controller parameters are adjusted based on matching the difference between the output of a pre-assigned reference model and the output of the controlled plant and making these errors converge to zero. It can be shown that model reference adaptive control can be viewed as a kind of direct self-tuning control [11].

2.1.3 Linear System Identification

System identification is a joint process of structure selection and parameter estimation of a dynamical system using observed input and output data. It is through an identification scheme that system (controller) structure and parameters are identified in indirect (direct) self-tuning control. Most often, in self-tuning control, plants are treated as linear systems and linear controllers (control law) are used. Following this convention, we here consider linear system identification.

A nominal model of a linear SISO (single-input single-output) time invariant system can be described as follows:

$$y_n = G_\theta(z^{-1})u_n + H_\theta(z^{-1})w_n = z^{-K}\frac{b(z^{-1})}{a(z^{-1})}u_n + \frac{c(z^{-1})}{d(z^{-1})}w_n, \tag{2.6}$$

where y_n and u_n are model output and input. w_n is noise. $a(z^{-1})$, $b(z^{-1})$, $c(z^{-1})$, $d(z^{-1})$ are four polynomials. Specially, $a(z^{-1})$ and $d(z^{-1})$ are two monic polynomials in the forms:

$$a(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{na} z^{-na}, \; a_{na} \neq 0,$$

$$d(z^{-1}) = 1 + d_1 z^{-1} + \cdots + d_{nd} z^{-nd}, \; d_{nd} \neq 0,$$

and $b(z^{-1})$ and $c(z^{-1})$ are expressed as

$$b(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{nb} z^{-nb}, \; b_{nb} \neq 0, \; b_0 \neq 0,$$

$$c(z^{-1}) = c_0 + c_1 z^{-1} + \cdots + c_{nc} z^{-nc}, \; c_{nc} \neq 0.$$

All the polynomial coefficients represent model parameters. Besides the order of the polynomials in Eq.(2.6), na, nb, nc, and nd, another quantity related to model structure of a linear system is the delay K. Clearly, it represents how many iterations the effect of model input on model output is delayed. Keeping the model (2.6) in mind, we see that linear system identification is a process of estimating both model order, denoted by (na, nb, nc, nd, K), and model parameters.

The aim of parameter estimation is to reduce parametric uncertainties in direct self-tuning control or to reduce their effect on the matching error between the desired output and the actual output of closed-loop systems. The purpose of performing order estimation in dealing with the linear system (2.6) is to reduce the unstructured uncertainties, which are also known as unmodeled dynamics. In adaptive control design algorithms that have received most interest, model uncertainties are ignored. The estimated parameters are treated as if they were the parameters of the actual plant model. It is not surprised that these algorithms may fail in the presence of model uncertainties [11][132]. Algorithms which consider these uncertainties are now being studied. The performance of these algorithms pretty much depends on the unstructured uncertainties. To have some insight into how model uncertainties affect system identification and adaptive control systems, we briefly review some results.

2.1.4 Self-Tuning Control Based on Certainty Equivalence Principle

In adaptive control the certainty equivalence principle is a statement about how to adjust controller parameters in self-tuning control. It says that controller parameters should be a solution of an underlying control system design problem for the model generated by an identification scheme. In most algorithms based on the certainty equivalence principle, the structure of the nominal model is pre-chosen. Thus, the identification scheme becomes a parameter estimator.

The minimum requirement on an adaptive control system is stability. This is a requirement that all the states and parameter estimates remain bounded. It is also required that the actual output of an adaptive control system converge to the desired output. According to the certainty equivalence principle, it is naturally believed that if there are no unstructured uncertainties and parameter estimates converge to the nominal model parameters fast enough, then the stability and convergence of the whole system can be achieved. Many adaptive control algorithms have been proved stable based on nominal models without unstructured uncertainties [9][11][132]. However, these algorithms can become unstable in the presence of mild unmodeled dynamics. An important set of observations on the effect of unmodeled dynamics on stability of adaptive control systems was made by Rohrs, Athans, Valavani, and Stein [128][129], which stimulated the later research on the robustness of adaptive control algorithms. One of the famous examples by Rohrs et al. is a third-order system:

$$G(s) = \frac{2}{s+1} \cdot \frac{229}{s^2 + 30s + 229}.$$

The reference model is $\frac{3}{s+3}$ and the nominal model is $\frac{k_p}{s+a_p}$. Thus, the unmodeled dynamics are described by a stable and well-damped transfer function of $\frac{229}{s^2+30s+229}$. The transfer function has a pair of complex conjugate poles $-15 \pm j2$ and a very big damping coefficient of 0.9912. An important aspect of the example is that from the viewpoint of classic control design, the unmodeled dynamics would be considered rather harmless. However, the simulation results and theoretical analyses show that this mild unmodeled dynamics cause some adaptive control algorithms to become unstable even though they have been proved stable for nominal models without unstructured uncertainties.

A great deal of effort has been devoted to improving robustness of adaptive control algorithms. The predominant course is to modify the parameter update laws(See [132] and references therein).

The stability of a closed-loop adaptive control system is also dependent on the reference model and the richness and dominant frequency range of the signals used in parameter estimation. This creates a major difficulty for robustness enchancement. The robustness of an algorithm is problem-dependent. An adaptive control algorithm which can tolerate some degree of unstructured uncertainties may be called robust in theory, but not necessarily in practice (for each individual problem). For instance, an algorithm could be unstable if the actual uncertainties present in a plant exceed the range of tolerable uncertainties of the algorithm. Therefore, a scheme for reducing unstructured uncertainties is needed in adaptive control. Such a procedure would reduce the actual uncertainties to allowable levels. Actually, this is one of the reasons for introducing order estimation in adaptive control.

2.1.5 Self-tuning Control Based on the Robust Control Principle

Recent progress in robust control has resulted in the proposal of a new adaptive control design principle [97], illustrated in Figure 2.1. A set of models are estimated rather than a single model. To emphasize that the underlying design problem is a robust control design problem, we refer to the principle as the *robust control principle.* Not only a nominal model within the set, but also a bound on the unstructured and parametric uncertainties associated with the model are fed to the robust design scheme. Controller parameters are then determined

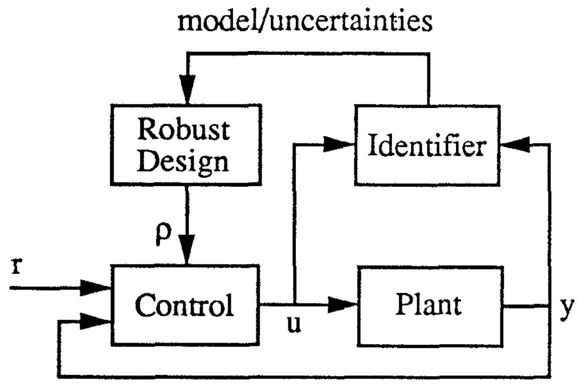


Figure 2.1: Figural illustration of the adaptive robust control

by solving a robust control design problem on-line. The difference between the robust control principle and the certainty equivalence principle is that model uncertainties are explicitly taken account in control design. In this sense, the robust control principle is an elaboration of, rather than a replacement for, the certainty equivalence principle. The success of the new design approach heavily depends on the choice of the presumed model set. If it does not contain the actual plant to be controlled, the desired performance, including stability, may not be guaranteed. If the prechosen model set is too big, the resulting adaptive control system will become very conservative and the system performance is unnecessarily degraded. For instance, the convergence becomes too slow. We are faced with an unknown plant in adaptive control. The introduction of an order estimation scheme could generate a model set of proper size which does contain the unknown plant generating measured data.

2.2 Adaptive IIR Filtering And Modeling

Adaptive IIR filtering is concerned with the use of an adjustable IIR (for infinite impulse response) filter whose zero-pole transfer function is adapted as the signals to be processed come. The aim of performing adaptive IIR filtering is to pass desired signals without degradation, to attenuate undesired or interfering signals, and/or to relieve any distortion on the channel input. An adaptive IIR filter is composed of two distinct parts: an IIR filter, whose structure is designed for the intended use of the filter, and an adaptation scheme for adjusting the parameters of the filter. The filter parameters are adjusted in a way such that the error between the filter output and the desired signal is minimized. The desired signal is unknown a priori. Consequently, the primary or key operation performed in adaptive IIR filtering is signal modeling. It involves two successive estimation procedures: (i) estimating the required filter parameters and (ii) then estimating the desired signal.

Though most adaptive filters used in practice are adaptive FIR (for finite impulse response) filters so far, over the last several years adaptive IIR filtering has been under active research. The primary interest in using adaptive IIR filters is that an IIR filter can achieve significantly better performance than an FIR filter having the same number of coefficients. In other words, an IIR filter has the potential for substantial decrease in computation over an FIR filter. Such a reduction is badly desired in some applications, including linear prediction [74], channel equalization [117], adaptive notch filtering [51][110], adaptive differential pulse code modulation [78], echo cancellation [105], and adaptive array processing [55]. For instance, in compensation for propagation effects [37], the transmitted signal s_n is propagated through a channel with distortion and the channel characteristics are described by a transfer function $P(z^{-1})$. The objective in designing an adaptive filter or compensator $H(z^{-1})$ is to recover the transmitted signal by processing the received signal (the channel output) in such a way that the matching error $s_n - H(z^{-1})P(z^{-1})s_n$ is minimized[1]. As a result, $H(z^{-1})P(z^{-1})s_n$ forms the estimate of the desired signal s_n. Often a transmission channel for a radio signal is described by an FIR filter. Consequently, the inverse of the channel transfer function $(\frac{1}{P(z^{-1})})$ is a rational function. Using an FIR filter as a compensator is equivalent to approximating a rational function

[1]This way of adjusting parameters of $P(z^{-1})$ is also referred to as inverse modeling [37].

by a finite-dimensional polynomial. Thus, the FIR filter needs a large number of coefficients to achieve satisfactory performance, especially when zeros of the transmission channel are close to the unit circle. This is true for digital communication of high frequency signals because a high sampling frequency causes the zeros to cluster near $z = 1$. However, the IIR counterpart needs as many filter coefficients as the channel transfer function $P(z^{-1})$ does. Such dramatic reduction of the number of coefficients can also be found in direct modeling [37]. For instance, a single strong resonance in a noisy background can be well modeled by a second-order IIR filter while an FIR filter would need hundreds of parameters to achieve the same performance.

The primary performance requirements on an adaptive filter are the convergence of the filter output (to the desired signal) and convergence rate. There are two kinds of filter design techniques for achieving the goal, which can be seen through considering the following example. Consider a stable feedback model:

$$d_n + \sum_{i=1}^{na} a_i d_{n-i} = \sum_{j=0}^{nb} b_j x_{n-j}, \tag{2.7}$$

where d_n is the desired signal which is measured by a noisy sensor: $y_n = d_n + w_n$, where y_n is the measurement and w_n is white noise. An IIR filter may be used to estimate d_n. In this case, suppose that the filter structure is designed in such a way that the estimation error becomes

$$e_n \triangleq \hat{d}_n - d_n = \sum_{j=0}^{nb} (\hat{b}_{n,j} - b_j) x_{n-j} - \sum_{i=1}^{na} (\hat{a}_{n,i} \hat{d}_{n-i} - a_i d_{n-i}).$$

It suffices to choose $\hat{b}_{n,j} = b_j$ and $\hat{a}_{n,i} = a_i$ to minimize the performance measure $\sum_{n=1}^{N} e_n^2, N > 0$. If so, then the estimation error e_n is governed by an AR model

$$e_n + \sum_{i=1}^{na} a_i e_{n-i} = 0$$

and $\lim\limits_{n \to \infty} e_n = 0$ due to the stability of the model (2.7). Here, the convergence of the filter output is achieved through the consistency of parameter estimation, which is related to system identification problems.

However, there is an important difference between a system identification problem and a filtering problem. In the former, a cost function of the error e_n is used only as a means of obtaining small parameter estimation error; in the later, a small error e_n is instead the desired end. As a result, adaptive filters in certain cases can tolerate substantial parameter estimation error while performing satisfactorily. This phenomenon reveals that parametrization of linear systems can play a greater role in filtering problems than in system identification problems. Indeed, this is one of the reasons why the use of FIR filters in adaptive signal processing is so popular. In fact, the model (2.7) can be well approximated by a finite dimensional open-loop (non-recursive) model

$$\tilde{d}_n = \sum_{j=0}^{Nb} b_j x_{n-j}. \tag{2.8}$$

The approximation error $\epsilon_n \triangleq d_n - \tilde{d}_n$ can be made arbitrarily small by assigning a big enough value of Nb. Thus, choosing an FIR filter structure yields the filter output error:

$$e_n = \sum_{j=0}^{Nb} (\hat{b}_{n,j} - b_j) x_{n-j} + \epsilon_n,$$

and $\lim_{n\to\infty} e_n = \epsilon_\infty$ if the filter parameters converge to the parameters of the fictitious model (2.8).

Effect of model order: Great flexibility in parameterizing a linear system is allowed in FIR signal processing. This by no means indicates that model order is not important. The importance of model order can be seen from the viewpoint of the convergence rate, which is of primary interest in some signal processing applications such as equalization and multipath reduction in high-frequency (3 to 30 MHz) digital communication channels [37]. If the order of an FIR filter, Nb, is too small, the resulting matching error is not satisfactory. Whereas, when Nb is too big, the convergence rate of the resulting matching error is slowed down and the computational complexity of the FIR filter becomes even higher.

A considerable improvement in computational load may be achieved by replacing an FIR filter by an IIR filter when the FIR filter results in a heavy computational burden. It may also reasonably be expected that a well-designed IIR filter will converge faster than its FIR counterpart. These benefits come at certain costs, however. In particular, an IIR filter is not unconditionally stable unless some restrictions are placed on the denominator parameters of the filter. So far, the stability and convergence issue for IIR filters is still an open problem. However, one point is certain. The model order plays a crucial role in IIR filtering. This can be seen from the example by Rohrs et al. [128] because IIR filters can be viewed as a special kind of adaptive control systems. In situations wherein the convergence of IIR filters relies on the consistency of parameter estimation, the model order is naturally considered as a key factor affecting the convergence rate.

Computational complexity: The potential of saving computation brought by IIR filters could be sacrificed unless some fast algorithms are developed. In FIR signal processing, many fast, numerically stable and robust, and parallel algorithms have been developed with a computational complexity less than $\mathbf{O}(n^2)$, where n denotes the number of filter coefficients [67]. Some counterparts are badly desired in IIR filtering.

2.3 Order Estimation

The effect of model order of linear systems on linear prediction was initially investigated by Davisson in 1965 [40] and 1966 [41]. Several years later, Akaike started an extensive investigation of estimation of model order [1][2][3][5][6]. He then proposed an information criterion for order estimation [4][7]. Since then, a lot of identification algorithms have been explored. Performance analysis of identification algorithms is usually conducted in a stochastic set-up, which is partially motivated by the presence of noise on any of available sensors and by the fact that noise terms are usually most gracefully described as stochastic processes. Here we briefly review these algorithms. The order is from strong to weak in the assumptions on the stochastic processes in question. The aim is to show which algorithms work well in adaptive control and signal processing.

2.3.1 Algorithms Based on Canonical Variate Analysis

A class of order estimation algorithms is based on canonical variate analysis, in which output and input processes are assumed weakly stationary[2] [6][14][96].

Given a sequence of cross-correlation matrices $\{R_l, l = 0, 1, \ldots\}$ of two zero-mean stationary processes $\{\mathbf{y}_n, \mathbf{u}_n, n = 0, 1, \ldots\}$ satisfying the assumption that there exist an integer p^* and a set of numbers $\{a_i\}$ such that for $\forall k \geq p^*$, $\sum_{i=0}^{p^*-1} a_i R_{k-i} = 0$, where $R_k = E\{\mathbf{y}_n \mathbf{u}_{n-k}^T\}$. Let $U_n^- = (\mathbf{u}_{n-1}^T, \mathbf{u}_{n-2}^T, \ldots, \mathbf{u}_0^T)^T$ be the vector of past inputs and $Y_n^+(p) = (\mathbf{y}_n^T, \mathbf{y}_{n+1}^T, \ldots, \mathbf{y}_{n+p}^T)^T$ be the vector of p-step future outputs. Denote by $R(Y_n^+(p))$ and $R(U_n^-)$, respectively, the spaces of all finite linear combinations of the components of $Y_n^+(p)$ and U_n^- with the mean square norm. By the theory of canonical correlations and variates, introduced by Hotelling [75], there exists an orthonormal basis $\bar{V} = (\bar{\nu}_1, \bar{\nu}_2, \ldots, \bar{\nu}_s)^T$ in the future space $R(Y_n^+(p))$ and another orthonormal basis $\bar{U} = (\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_r)^T$ in the past space $R(U_n^-)$. Also, they are related to each other through $E\bar{u}_i\bar{\nu}_j = r_{ij},\ i = 1, 2, \ldots, s,\ j = 1, 2, \ldots, r$, where $r_{ij} = 0\ (i \neq j), 1 \geq r_{ii} \geq r_{i+1,i+1} \geq 0$. The variables $\bar{u}_i$ and $\bar{\nu}_i$ are referred to as the canonical variates and r_{ii} as the canonical correlation coefficient of $\bar{u}_i$ and $\bar{\nu}_i$ ($i = 1, 2, \ldots, min(r, s)$). The number of nonzero canonical correlation coefficients , denoted by k, is equal to the rank of the Hankel correlation matrix

$$E\{Y_n^+(p)(U_n^-)^T\} = \begin{pmatrix} R_1 & R_2 & \ldots & R_n \\ R_2 & R_3 & \ldots & R_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ R_p & R_{p+1} & \ldots & R_{p+n-1} \end{pmatrix}.$$

The projection of the future space $R(Y_n^+(p))$ on the past space $R(U_n^-)$, called the prediction space, is spanned by k canonical variates, $(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_k)^T$, in the past space $R(U_n^-)$. This fact indicates that the next p-step optimal prediction

[2] For simplicity, we will refer to weak stationarity as stationarity later.

of the output process can be expressed by the past inputs. In other words, the state $\mathbf{x}_n$ defined as $\mathbf{x}_n \triangleq (\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_k)^T$ can be interpreted as a minimal amount of information describing the future behavior of the output process $\mathbf{y}_n$, which is consistent with the definition of states of deterministic systems [82]. By some arguments, the stochastic process can be described by

$$\begin{aligned} \mathbf{x}_{n+1} &= F\mathbf{x}_n + B\mathbf{u}_n \\ \mathbf{y}_n &= C\mathbf{x}_n + D\mathbf{u}_n, \end{aligned} \tag{2.9}$$

which is essentially Akaike's representation [6] although the $D\mathbf{u}_n$ is missing in Akaike's representation due to the inclusion of $\mathbf{u}_n$ in U_n^-. By the result of Yaglom [152], such a realization exists if and only if the process has a rational power spectrum, i.e., is a finite-order Markov process. The parameter matrices F, B, C, D can be computed by using the Ho-Kalman algorithm for realization of general systems [6]. They can also be obtained by

$$\begin{aligned} A &= E\{\mathbf{x}_{n+1}\omega_n^T\}[E\{\mathbf{x}_n\omega_n^T\}]^{-1} \\ B &= E\{\mathbf{x}_{n+1}\mathbf{u}_n^T\}[E\{\mathbf{u}_n\mathbf{u}_n^T\}]^{-1} \\ C &= E\{\mathbf{y}_n\omega_n^T\}[E\{\mathbf{x}_n\omega_n^T\}]^{-1} \\ D &= E\{\mathbf{y}_n\mathbf{u}_n^T\}[E\{\mathbf{u}_n\mathbf{u}_n^T\}]^{-1} \end{aligned}$$

provided $\mathbf{u}_n$ is an uncorrelated process and ω_n is a k-dimensional vector of elements of the past space $U_n^-(P)$ such that the matrix $E\{\mathbf{x}_n\omega_n^T\}$ is of full rank [14]. The canonical variates can be computed by using a computationally efficient, numerically robust and stable method, based upon the singular value decomposition of the Hankel correlation matrix [54].

Larimore enriched the understanding of the canonical variate analysis in identification, filtering, and adaptive control [95][96]. He pointed out that Akaike's realization provides an optimal one in terms of mutual information for predicting an output process because of the statistical meaning of the state $\mathbf{x}_n$ in Akaike's realization (2.9). He suggested a few approximation methods for state-space realization of general stationary processes. For example, an optimal procedure in terms of prediction error was derived based on truncated correlation functions to estimate parameters in Akaike's realization (2.9).

When the exact, unknown correlations are replaced by their sample correlations, the finite structure of the sample Hankel correlation matrix is normally lost due to the discontinuity of matrix rank over matrix coefficients. Another major consequence is that the minimal order of realization (2.9) cannot be determined exactly from any finite dimensional Hankel matrix of sample correlations. Baram and Porat proposed a nonrecursive procedure to overcome this difficulty [14]. Under the assumption of Gaussian stationary input and output processes, they derived a procedure for calculating an approximate probability distribution of the squared singular values of a sample Hankel correlation matrix. This approximate probability distribution converges to the true one as the number of data points becomes large. Based on this approximate probability and the principle of hypothesis testing, the singularity of the Hankel correlation matrix can be decided simply by sequentially testing the squared singular values of the sample Hankel correlation matrices of the different dimension. Consequently, the above procedure can be interpreted as a procedure to estimate the minimal order of the state space representation of a linear time-invariant system since the order is equal to the rank of the Hankel correlation matrix.

2.3.2 Algorithms Based on Analysis of Residuals

Consider a regression model with additive noise:

$$\begin{aligned} y_n &= a_1\phi(n-1) + a_2\phi(n-2) + \cdots + a_m\phi(n-m) + w_n, \quad a_m \neq 0 \\ &= \theta_m^T \phi_{n-1}^p + w_n, \end{aligned} \tag{2.10}$$

where $\phi_{n-1}^p \triangleq [\phi(n-1)\ \phi_2(n-2) \cdots \phi_p(n-p)]^T$ is the regression vector and $\theta_p \triangleq [a_1\ a_2 \cdots a_p]^T$ is the model parameter vector. w_n is noise. The corresponding predictor is of the form:

$$\begin{aligned} \hat{y}_n &= \hat{a}_1\phi(n-1) + \hat{a}_2\phi(n-2) + \cdots + \hat{a}_p\phi(n-p) \\ &= \hat{\theta}_p^T \phi_{n-1}^p, \end{aligned} \tag{2.11}$$

where $\hat{\theta}_p$ is defined in the same manner as θ_p. Note that the predictor order is given by the integer p. Define the least mean-square (prediction) error

$$\sigma_p^2 \triangleq \min_{\tilde{\theta}_p \in R^p} E(y_k - \tilde{\theta}_p^T \phi_{k-1}^p)^2.$$

Thus [21],

$$\sigma_p^2 > \sigma_m^2 \quad \text{for} \quad p < m \tag{2.12}$$

and

$$\sigma_p^2 = \sigma_m^2 \quad \text{for} \quad p \geq m. \tag{2.13}$$

In the case of time series, the predictor parameters may be determined by using the LS algorithm [21] and the corresponding LS predictor is:

$$\begin{aligned} \hat{y}_n(\hat{\theta}_p^n) &= \hat{\theta}_p^{n^T} \phi_{n-1}^p, \\ \hat{\theta}_p^n &= (\textstyle\sum_{t=1}^n \phi_{t-1}^p \phi_{t-1}^{p^T})^{-1} (\sum_{t=1}^n \phi_{t-1}^p y_t). \end{aligned} \tag{2.14}$$

Assume that y_n and w_n are ergodic in the almost sure sense and y_n is of full-rank (a kind of persistent excitation condition)[3]. Under these assumptions, the time-average prediction error (residual) $\sigma_p^2(n) \triangleq \frac{1}{n} \sum_{k=1}^n (y_k - \hat{y}_k(\hat{\theta}_p^k))^2$ will converge to σ_p^2 almost surely. This plus (2.12) means that the asymptotics of $\sigma_p^2(n)$ can be used to exclude under-modeled predictors. On the other hand, it is intuitively thought that for an over-modeled LS predictor of order $p > m$, its time-average prediction error satisfies $\sigma_p^2(n) > \sigma_m^2(n) - \epsilon_p(n), \forall n < \infty$, for some positive sequence $\{\epsilon_p(k)\}_{k=1}^\infty$ satisfying $\epsilon_p(k) \searrow 0$ $a.s.$ as $k \to \infty$. This is so due to the effect of the extra nonzero estimates of the zero coefficients $a_{m+1}, \cdots, a_p$. Subsequently, there exists a sequence of $\{\epsilon(k)\}_{k=1}^\infty$ such that

$$\sigma_p^2(n) > \sigma_m^2(n) - \epsilon(n), \quad \forall p > m \tag{2.15}$$

if the order p is finite. Meanwhile, it follows from (2.12) that (2.15) is also true for the undermodeled case of $p < m$ where n is big enough. Note that for ϵ

[3]The assumptions are unnecessarily strong.

small, $\log(1+\epsilon) \approx \epsilon$. Thus, the above arguments on both under-modeled and over-modeled cases yield that, asymptotically,

$$\log \sigma_p^2(n) > \log \sigma_m^2(n) - \frac{\epsilon(n)}{\sigma_m^2(n)} \tag{2.16}$$

for all finite $p \neq m$. Equivalently, $\log \sigma_p^2(n) + \frac{C_n}{n} \geq \log \sigma_m^2(n)$ for all finite $p \neq m$ and the equality is attained only when $p = m$, where $C_n \triangleq \frac{n\epsilon(n)}{\sigma_m^2(n)}$. Following this idea, several order estimation algorithms with different definitions of C_n are proposed. The principle algorithms among them are the AIC (for Akaike information criterion), BIC (for Bayesian information criterion), Hannen criterion:

AIC $= \log \sigma_p^2(n) + \frac{2p}{n}$ (Akaike [7])

BIC $= \log \sigma_p^2(n) + \frac{p}{2}\frac{\log n}{n}$ (Schwarz [133])

Hannen Criterion $= \log \sigma_p^2(n) + cp\frac{\log\log n}{n}$, (Hannen and Quinn [60])

where p represents predictor order and the quantity c appearing in the Hannen criterion is a constant. Not only have the AIC, BIC and Hannen criterion been applied to direct variations of regression model, AR models or MA models, but also to ARMAX models described as follows [61]:

$$\sum_{j=0}^{p} \alpha_i y_{k-j} = \sum_{j=1}^{q} \mu_j u_{k-j} + \sum_{j=0}^{r} \beta_j w_{k-j}, \quad \alpha_0 = 1.$$

In the latter case, the order of the model feedback and feedforward polynomials and the noise feedforward polynomial (p, q, r) is estimated. Under some sort of stablility and ergodicity assumptions [62], the consistency of order estimation performed by minimizing the criterion

$$\log \sigma_p^2(n) + \frac{C_n}{n} \tag{2.17}$$

is guaranteed if

$$\begin{aligned} &\lim_{n\to\infty} \inf \frac{C_n}{2\log\log n} > 1 \\ &\lim_{n\to\infty} \frac{C_n}{n} = 0. \end{aligned} \tag{2.18}$$

The constraints on the function C_n stated in (2.18) imply that C_n/n cannot approch zero too fast. This constraint is reasonably expected from (2.16); otherwise the inequality (2.16) would be violated for some predictor order p close to the true order m. Clearly, the AIC does not satisfy (2.18), but the BIC and Hannen criteria do. Indeed the AIC is not a consistent order estimation criterion under broad assumptions. Note that some sort of stablility and ergodicity assumptions are also necessary to ensure the consistency of order estimation performed by minimizing the criterion (2.17). This is because consistency relies on the convergence of the time-average residual $\sigma_p^2(n)$ to σ_p^2.

2.3.3 Algorithms Applicable to Adaptive Control and Filtering

In the time-series framework, and in particular in the above mentioned algorithms, some sort of stability and erdodicity of the stochastic processes involved are usually assumed. However, feedback control signals are usually not stationary [102] and in most applications of adaptive signal processing, the signal waveforms are usually at best only quasi-stationary [102]. As a result, these results and algorithms cannot directly be applied to feedback control systems.

The Information Criterion for Control Systems

Considering multivariate ARX models, Chen et al. proposed an information criterion C_n as a modified version of the BIC. Applying the criterion to the systems for order estimation and using the adaptive control law developed in reference [25], they devised an asymptotically optimal adaptive control minimizing a quadratic cost function [28]. In addition, both the order and the coefficients of the systems are strongly consistently estimated during the control process, The previous results were then generalized to ARMAX systems, where the information criterion for order estimation is a new criterion called the CIC to exphasize that the criterion is devised for control systems [59]. Specifically, the on-line order estimation is performed in the following way: (i) estimate parameters of all possible models, (ii) determine the order estimate as the minimizer of the CIC over all possible models with parameters being the corresponding estimates. Note that, this order estimation approach is computationally expensive because it requires the estimation of the coefficients of all possible models at each sampling time.

Very recently, they addressed an integrated identification and adaptive tracking problem for SISO (for single-input and single-output) ARX systems with delay, where an adaptive LQ regulator was used and the model order, delay, and parameters were strongly consistently estimated [29]. To reduce computational load, the CIC is modified such that only one model, the most complicated possible model, explicitly requires LS parameter estimation. The resulting parameter estimates are then truncated to replace the desired parameter estimates of less complicated models which are needed in order estimation. Notice that the convergence rate of parameter estimation is dominated by the minimum eigenvalues of the normal matrices involved in parameter estimation of ARX models. In a considerably long time period, the performance of order estimation is heavily dependent on the accuracy of parameter estimates and the condition number of the normal matrices. This can also be seen from the simulation results presented in Chapter 5. Therefore, it is reasonably believed that the transient performance of parameter and order estimation is degraded considerably although the computational loads are reduced significantly.

Minimum Description Length Principle

In the conventional identification frameworks, all observations are assumed to be "random samples" from a parent population with an associated distribution. But the fact which is frequently overlooked is that there is usually no "true" parent distribution behind the data because no finitely parameterized distribution can be the generator for the observations. A clever solution to the basic modeling dilemma, which is free from the difficulty, has been proposed and defined as the minimum description length principle (MDL) by Rissanen [119][125]. In his approach there is no need to assume any parent population at all. The prefix code of the observed data is used as its description since the data can be written out by decoding the prefix code. The best understanding of the given data may be interpreted as the prefix code with the shortest possible codelength since such a code has no redundancy. To avoid the problems involved in the existence of such a code and to build an algorithm to get the optimal code, a class of probabilistic models, $P_\theta(\mathbf{y}^n)$, is chosen and the "best" understanding of the data is reduced to the prefix code with the shortest possible codelength, called the *stochastic complexity*, relative to the selected class of probabilistic models. As Rissanen pointed out, the selection of the model class is clearly of crucial importance, and it requires "sound judgement" using all the prior knowledge.

The total coding of the data, $\mathbf{y}^n = \{\mathbf{y}_n^T, \mathbf{y}_{n-1}^T, \cdots, \mathbf{y}_1^T\}^T$, can be done with about $\min_{k,\theta}\{-\log P_\theta(\mathbf{y}^n) + L(\theta) + L(k)\}$ bits, where $L(\theta)$ and $L(k)$ are, respectively, the code length functions for the k-dimensional parameter vector θ and number k. It has been established by Rissanen [119] that the length function $L(\theta)$ for large samples is about $L(\theta) = \frac{k}{2}\log n$ bits. Compared with $L(\theta)$, the coding cost $L(k)$ can be ignored because k is an integer. Thus, the nonpredictive MDL criterion is intuitively defined as

$$I_{NP}(\mathbf{y}^n) = \min_{k,\theta}\{-\log P_\theta(\mathbf{y}^n) + \frac{k}{2}\log n\}.$$

It is also called the nonpredictive stochastic complexity by a slight abuse of terminology.

Instead of estimating $(\hat{k}_n, \hat{\theta}_n)$ based on the entire string $\mathbf{y}^n$, we can encode each data point $\mathbf{y}_{n+1}$ in the string by use of a model, $P_{\hat{\theta}_k^n}(\mathbf{y}_{n+1}|\mathbf{y}^n)$, where $\hat{\theta}_k^n$ denotes, say, the ML estimates of the k-dimensional parameter vector, based on the data $\mathbf{y}^n$. By Shannon's coding theorem, the predictive stochastic complexity is

$$I_P(\mathbf{y}^n) = \min_k\{-\sum_{l=0}^{n-1}\log P_{\hat{\theta}_k^l}(\mathbf{y}_{l+1}|\mathbf{y}^l) + L(k)\}, \tag{2.19}$$

which is also called the predictive MDL criterion [125]. The MDL principle has had a great impact on stochastic inference and information theory [120][121][122] [123].

Suppose that the observation y_n is the output of a linear regression model (2.10) with the noise w_n being a zero-mean i.i.d. gaussian process. Then, the solution of the minimization problem described by the right side of Eq.(2.19) is equal to the solution to the following problem:

$$\min_k \sum_{t=1}^{n} (y_t - \hat{y}_t(\theta_k^{t-1}))^2$$

where $\hat{y}_t(\hat{\theta}_k^{t-1}) \triangleq \hat{\theta}_k^{t-1^T} \phi_{t-1}^k$ is the LS "*honest*" prediction of y_n rather than the LS *posterior* prediction. As shown in (2.14), this is because $\hat{\theta}_k^{t-1}$ is determined by data measurements up to time $t-1$. The information criterion

$$\sum_{t=1}^{n} (y_t - \hat{y}_t(\hat{\theta}_k^{t-1}))^2 \tag{2.20}$$

is often called the accumulated ("honest") prediction error criterion and the APE for short [148] or the predictive least-squares criterion [71][72]. Specifically, consider a MIMO MAX predictor

$$\hat{\mathbf{y}}_k = \sum_{i=1}^{p} A_i \mathbf{y}_{k-i} + \sum_{i=0}^{q} B_i \mathbf{u}_{k-i}$$

which, of course, corresponds to an ARX model. Then the APE looks like

$$\sum_{t=1}^{n} (\mathbf{y}_t - \hat{\theta}_{t-1}^T(p,q)\phi_{t-1}(p,q))^2 \tag{2.21}$$

where

$$\begin{aligned} \hat{\theta}_n^T(p,q) &\triangleq [\hat{A}_{n,1}, \ldots, \hat{A}_{n,p}, \hat{B}_{n,0}, \ldots, \hat{B}_{n,q}] \\ &= (\textstyle\sum_{t=1}^{n} \phi_{t-1}(p,q)\phi_{t-1}^T(p,q))^{-1} \sum_{t=1}^{n} \phi_{t-1}(p,q)\mathbf{y}_t^T. \end{aligned} \tag{2.22}$$

and

$$\phi_{n-1}^T(p,q) = [\mathbf{y}_{n-1}^T \cdots \mathbf{y}_{n-p}^T \; \mathbf{u}_n^T \cdots \mathbf{u}_{n-q}^T]^T. \tag{2.23}$$

The APE has been justified in several ways. For example, Gerencser [52] has shown that the stochastic complexity is an achievable lower bound for the APE if the observations come from a stationary Gaussian ARMA process. The APE criterion can be applied to consistent order selection of an AR model with a stationary Gaussian noise process [148]. More significantly, the APE, combined with the adaptive control strategy devised in [25], has been used for self-tuning control of ARX systems with martingale difference noise [72]. It is shown there that, under some mild assumptions, the parameters and order are estimated in a strongly consistent way while the optimal cost of the whole adaptive control system is achieved asymptotically.

2.4 Fast Algorithms

Adaptation schemes contained in adaptive control and adaptive signal processing usually involve an extensive amount of computations for each iteration. For instance, in order estimation in adaptive signal processing or in adaptive control, LS parameter estimates of all possible models have to be determined on-line in order to achieve good transient performance of the order estimation and then of the overall adaptive systems. When a model set of scalar ARX models is chosen and the model order (p,q) varies from $(0,0)$ to (p^*,q^*), the number of operations required to solve the resulting LS parameter estimation problem is of the order of $(p^*+q^*)^4$ multiplications and additions. This can be prohibitive if p^* or q^* is large. This fact has forced the development of fast algorithms to implement adaptation procedures derived from various disciplines.

2.4.1 The Fundamental Levinson Algorithm and Its Impact

In 1947 Levinson presented a recursive algorithm for solving the LMMSE (for linear minimum mean-square error) prediction problem of a scalar stationary autoregressive process [99]:

$$\hat{y}_{p|p-1} = \sum_{i=1}^{p} a_{p,i} y_{n-i}$$

with the minimum mean-square error σ_p. Motivated by the fact that the non-increasing quantity σ_p could be used as a measure to decide if p should be larger in order to achieve the desired mean-square error, Levinson stressed the importance of finding a method to successively compute $\sigma_p, p = 1, 2, \cdots$. To take maximum advantage of the previous efforts made to calculate σ_p and the associated predictor parameters, he introduced a recursive procedure by exploiting the Toeplitz structure of the Yule-Walker equation, where the previous solution to the Yule-Walker equation of $p+1$ dimension was used as a partial solution to a new Yule-Walker equation of $p+2$ dimension, which corresponds to an LMMSE problem of order $p+1$. The whole solution is then equal to a weighted summation of the partial solutions and an auxiliary solution which is composed of the least mean-square error and the associated parameters of a *"reversed-time" auxiliary predictor.*

The significance of the Levinson algorithm was not immediately realized. In Levinson's words [99], "A few months after Wiener's work appeared, the author, in order to facilitate computation procedure, worked out an approximate, and one might say, mathematically trivial procedure." However, 30 years later, this "mathematically trivial" algorithm has had tremendous impact on various fields, both directly and indirectly [80].

The Direct Impact of the Levinson Algorithm

After the work by Durbin [45], Whittle [150], and Wiggins and Robbinson [151], the relation between the Levinsion algorithm and (block) Toeplitz systems of linear equations was well established. As a result, the Levinson-Whittle-Wiggins-Robbinson algorithm becomes a fast algorithm for solving (block) Toeplitz systems. A well known fact is that Toeplitz systems of linear equations arise from many sources (see [20] and references therein.) and Toeplitz systems are closely related to Hankel systems [76], which also appear in many physics and engineering problems [82]. On the other hand, the Levinsion-Durbin algorithm has been proved numerically stable [38]. As a result of numerical stability, the computational efficiency, and the feature of recursively solving all Toeplitz systems of successively increasing dimension as well as the plenty of Toeplitz systems, the Levinson algorithm and its various modifications have been applied to many areas, including geophysics [127], spectral estimation [86], speech analysis [107], linear filtering [53][66], image processing [77], control theory [138], and statistics [65].

The Indirect Impact of the Levinson Algorithm

Note that the Levinson algorithm involves a procedure of using a forward predictor as a partial solution and using a backward predictor as an auxiliary solution to correct the partial solution. In fact, this is a "powerful idea", which has had great influnce in many areas [80]. Here, we discuss some indirect impacts of the Levinson algorithm on linear filtering.

Order recursive LS algorithm: The use of LS predictors is very popular in signal processing because no information on the statistics of the data is required. When a linear regression model is used to describe prediction errors, the parameters of an LS predictor are a solution to the associated normal equation, which is usually of non-Toeplitz structure [67]. The loss of the Toeplitz structure prevents the Levinson algorithm from being applied to the fast solution to a non-Toeplitz normal equation directly. However, the backward predictors introduced in the Levinson algorithm are crutial to the development of the recursive LS lattice algorithm [37][53], which can determine, at each time instant, the parameters and prediction errors of a moving-average LS predictor successively in predictor order.

Joint Estimation: In certain cases of adaptive signal processing, including joint estimation, a sequence of signals is needed to be orthonormalized and the orthonormalized signals are then used in processing other related signals. The backward predictors introduced for the Levinson algorithm actually orthogonalize the input signal in the mean-square sense. This feature has played a great role in joint estimation as another indirect impact of the Levinson algorithm [37][66].

Lattice Algorithm: The lattice structure is a very popular option for realizing digital filters. It does have several advantageous properties including cascading of identical stages, easy stability inspection, robustness against round-off errors. For most practical lattice filters, backward predictors are a necessary component. This is another evidance of the indirect impact from the Levinson algorithm [49].

2.4.2 Algebraic Methods for Toeplitz-like Systems

Problems in many fields lead ultimately to solving a system of linear equations

$$R_k x = y, \tag{2.24}$$

where R_k is a given square matrix of dimension k which is not necessarily a Toeplitz matrix. And y is a given column vector of dimension k. As is well known, the computational complexity for solving the equations in (2.24) is of order of k^3 multiplications and additions. Such a computation load is not always affordable in applications, espectially when k is large (500 or 1000 or 3000, as can arise in many power system or econometric problems). When the linear equations in (2.24) are a Toeplitz system, the number of operations required can be reduced to the order of k^2. This suggests that (i) exploiting the underlying structure of matrix R_k in (2.24) could lead to a big saving in computation, (ii) for a non-Toeplitz matrix R_k, there should exist an integer, denoted by α, say, such that it takes $\mathbf{O}(\alpha k^2)$ operations of multiplications and additions to solve the equtions in (2.24), where $1 \leq \alpha \leq k$. Indeed, such an integer has been found by Friedlander et al. [47][48] and is given the name of the displacement rank by Kailath et al. [81]. The displacement rank α is defined as follows

$$\alpha(R_k) = \text{rank}(R_k - Z_k R_k Z_k^T), \tag{2.25}$$

where Z_k is the $k \times k$ lower shift matrix, zero being everywhere except for 1's on the first subdiagonal[4]. A Toeplitz matrix T has a displacement rank of 2 no matter what the dimension of the matrix T is. For a normal matrix appearing in

[4]The definition of the displacement rank for non-square matrices can be found in reference [31].

a parameter estimation problem for an autoregressive model, the rank is always less than or equal to 4 [81]. In fact, the displacement rank can be viewed as a measure of the "distance" between a general matrix and a Toeplitz matrix. When the rank $\alpha(R_k)$ is small compared with the dimension k of the matrix R_k, the matrix is called as a Toeplitz-like matrix. A fast algorithm for determining $[R_k]^{-1}$ and solving

$$R_k A_k = (0 \quad 0 \quad \cdots \quad 0 \quad N_k^T)^T, \quad A_k^T \triangleq (I \quad A_{k,1}^T \quad \cdots \quad A_{k,k}^T)$$

has been developed [47][48]. Of course, it can also be used to fast solutions to Toeplitz-like systems of linear equations. *The starting point of the algorithm is the displacement representation of a matrix.*

Denote a block Toeplitz-like matrix by

$$R_k = [r_{i,j}], \quad 0 \leq i, j \leq k, \tag{2.26}$$

where $r_{i,j}, 0 \leq i, j \leq k$, are $p \times p$ block elements. Thus, the matrix R_k can be presented in the form, which is called the displacement representation:

$$R_k = (R_k - Z_k R_k Z_k^T) + Z_k R_k Z_k^T, \tag{2.27}$$

where Z_k is a $kp \times kp$ lower shift matrix and

$$R_k - Z_k R_k Z_k^T = \begin{pmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,k} \\ r_{1,0} & & & \\ \vdots & & \delta[R_k] & \\ r_{k,0} & & & \end{pmatrix} \tag{2.28}$$

and the "shifted-difference" operator $\delta[\cdot]$ is

$$\delta[R_k] = \begin{pmatrix} r_{1,1} & \cdots & r_{1,k} \\ \vdots & \ddots & \vdots \\ r_{k,1} & \cdots & r_{k,k} \end{pmatrix} - \begin{pmatrix} r_{0,0} & \cdots & r_{0,k-1} \\ \vdots & \ddots & \vdots \\ r_{k-1,0} & \cdots & r_{k-1,k-1} \end{pmatrix}. \tag{2.29}$$

For a non-Toeplitz matrix R_k, the "shifted-difference" $\delta[R_k]$ is not equal to a zero matrix. Furthermore, when R_k is a symmetric matrix, $\delta[R_k]$ can be decomposed into

$$\delta[R_k] = D_k \Sigma D_k^T, \quad D_k \text{ is } kp \times \hat{\alpha}p, \quad \Sigma \text{ is } \hat{\alpha}p \times \hat{\alpha}p, \tag{2.30}$$

and Σ *is a diagonal matrix consisting of* $\pm$ *1's.* The dimension $\hat{\alpha}$ is greater than or equal to $\text{rank}(\delta[R_k]) \leq \alpha(R_k)$.

To close the recursion in the fast algorithm proposed by Friedlander [47][48], an auxiliary solution to the equation

$$R_k B_k = \begin{pmatrix} I & 0 \\ 0 & \\ \vdots & \bar{D}_{k+1} \\ 0 & \end{pmatrix} M_k, \quad B_k \triangleq (\, B_{k,k}^T \quad B_{k,k-1}^T \quad \cdots \quad I\,)^T$$

is also solved, where the matrix $\bar{D}_{k+1}$ denotes the first k block rows of D_{k+1} in (2.30) and M_k is an invertible matrix calculated recursively. *Note that only when $\bar{D}_{k+1}$ is equal to zero, i.e., R_k is a block Toeplitz matrix, the auxiliary solution B_k could be interpreted as the parameters of an LMMSE MA predictor.*

In some cases of interest, the decomposition can be discovered by exploiting the problem structure (See [47][48][67] and references therein.) and no computation is needed. However, in some problems, the factorization (2.30) requires some numerical methods, for instance, the singular value decomposition (SVD) method. As we know, the SVD for a general square matrix of k dimension takes $\mathbf{O}(k^3)$. As a result, this severely constrains the generality of the algorithms based on the displacement representation. In fact, few applications of the algorithms based on the displacement representation have been seen in signal processing although fast algorithms are very much wanted there. Maybe the difficulty of obtaining the displacement representation is one of the reasons.

Example 1: Consider a linear minimum mean-square error parameter estimation problem

$$\min_{a_i,\ c_j} E(y_n + \sum_{i=1}^{3} a_i y_{n-i} - \sum_{j=0}^{2} c_j u_{n-j})^2. \tag{2.31}$$

It is well known that the minimizer to the problem (2.31) is equal to the solution to a Yule-Walker equation if the output y_n and input u_n are a jointly-stationary process.

$$\begin{pmatrix} 1 & a_1 & a_2 & a_3 & c_0 & c_1 & c_2 \end{pmatrix} R_7 = \begin{pmatrix} r^f & 0 & \cdots & 0 \end{pmatrix}, \tag{2.32}$$

where the Yule-Walker matrix R_7 is a symmetric matrix denoted by

$$\left(\begin{array}{cccc|ccc}
r_0^{yy} & r_1^{yy} & r_2^{yy} & r_3^{yy} & r_0^{yu} & r_1^{yu} & r_2^{yu} \\
r_{-1}^{yy} & r_0^{yy} & r_1^{yy} & r_2^{yy} & r_{-1}^{yu} & r_0^{yu} & r_1^{yu} \\
r_{-2}^{yy} & r_{-1}^{yy} & r_0^{yy} & r_1^{yy} & r_{-2}^{yu} & r_{-1}^{yu} & r_0^{yu} \\
r_{-3}^{yy} & r_{-2}^{yy} & r_{-1}^{yy} & r_0^{yy} & r_{-3}^{yu} & r_{-2}^{yu} & r_{-1}^{yu} \\
\hline
r_0^{uy} & r_1^{uy} & r_2^{uy} & r_3^{uy} & r_0^{uu} & r_1^{uu} & r_2^{uu} \\
r_{-1}^{uy} & r_0^{uy} & r_1^{uy} & r_2^{uy} & r_{-1}^{uu} & r_0^{uu} & r_1^{uu} \\
r_{-2}^{uy} & r_{-1}^{uy} & r_0^{uy} & r_1^{uy} & r_{-2}^{uu} & r_{-1}^{uu} & r_0^{uu}
\end{array}\right)$$

and r_k^{yy}, r_k^{uu}, r_k^{yu}, and r_k^{uy} are the autocorrelation coefficients and cross-correlation coefficients of the output and input processes. Note that the Yule-Walker equation here is not a Toeplitz system.[5] As a result, the "shifted difference" of the matrix is not a zero matrix and it is of the form:

$$\begin{pmatrix}
r_0^{yy} & r_1^{yy} & r_2^{yy} & r_{-1}^{yu} & r_0^{yu} & r_1^{yu} \\
r_{-1}^{yy} & r_0^{yy} & r_1^{yy} & r_{-2}^{yu} & r_{-1}^{yu} & r_0^{yu} \\
r_{-2}^{yy} & r_{-1}^{yy} & r_0^{yy} & r_{-3}^{yu} & r_{-2}^{yu} & r_{-1}^{yu} \\
r_1^{uy} & r_2^{uy} & r_3^{uy} & r_0^{uu} & r_1^{uu} & r_2^{uu} \\
r_0^{uy} & r_1^{uy} & r_2^{uy} & r_{-1}^{uu} & r_0^{uu} & r_1^{uu} \\
r_{-1}^{uy} & r_0^{uy} & r_1^{uy} & r_{-2}^{uu} & r_{-1}^{uu} & r_0^{uu}
\end{pmatrix}
-
\begin{pmatrix}
r_0^{yy} & r_1^{yy} & r_2^{yy} & r_3^{yy} & r_0^{yu} & r_1^{yu} \\
r_{-1}^{yy} & r_0^{yy} & r_1^{yy} & r_2^{yy} & r_{-1}^{yu} & r_0^{yu} \\
r_{-2}^{yy} & r_{-1}^{yy} & r_0^{yy} & r_1^{yy} & r_{-2}^{yu} & r_{-1}^{yu} \\
r_{-3}^{yy} & r_{-2}^{yy} & r_{-1}^{yy} & r_0^{yy} & r_{-3}^{yu} & r_{-2}^{yu} \\
r_0^{uy} & r_1^{uy} & r_2^{uy} & r_3^{uy} & r_0^{uu} & r_1^{uu} \\
r_{-1}^{uy} & r_0^{uy} & r_1^{uy} & r_2^{uy} & r_{-1}^{uu} & r_0^{uu}
\end{pmatrix}$$

$$= \begin{pmatrix}
0 & 0 & 0 & r_{-1}^{yu} - r_3^{yy} & 0 & 0 \\
0 & 0 & 0 & r_{-2}^{yu} - r_2^{yy} & 0 & 0 \\
0 & 0 & 0 & r_{-3}^{yu} - r_1^{yy} & 0 & 0 \\
r_1^{uy} - r_{-3}^{yy} & r_2^{uy} - r_{-2}^{yy} & r_3^{uy} - r_{-1}^{yy} & r_0^{uu} - r_0^{yy} & r_1^{uu} - r_{-3}^{yu} & r_2^{uu} - r_{-2}^{yu} \\
0 & 0 & 0 & r_{-1}^{uu} - r_3^{uy} & 0 & 0 \\
0 & 0 & 0 & r_{-2}^{uu} - r_2^{uy} & 0 & 0
\end{pmatrix}.$$

[5]In fact, there is a special structure within the matrix such that the matrix can be partitioned into four submatrices with the upper-left submatrix and the lower-right submatrix of dimension 4×4 and 3×3. Under such a partition, all four submatrices are of the Toeplitz structure, i.e., "constant along any block diagonal."

No analytic procedure for decomposing the above "shifted difference" has been found so far for arbitrary autocorrelation coefficients and cross-correlation coefficients.

Example 2: Consider a least-squares (LS) parameter estimation problem for an ARX model with order of $(2,1)$. The LS parameter estimate $(a_1\ a_2\ -c_0\ -c_1)$ is a solution to the following normal equation

$$(\,a_1 \quad a_2 \quad -c_0 \quad -c_1\,)\, R_4 = m,$$

where the symmetric normal matrix R_k is denoted by

$$\frac{1}{N}\sum_{n=1}^{N}\begin{pmatrix} y_{n-1}y_{n-1} & y_{n-1}y_{n-2} & y_{n-1}u_n & y_{n-1}u_{n-1} \\ y_{n-2}y_{n-1} & y_{n-2}y_{n-2} & y_{n-2}u_n & y_{n-2}u_{n-1} \\ u_n y_{n-1} & u_n y_{n-2} & u_n u_n & u_n u_{n-1} \\ u_{n-1}y_{n-1} & u_{n-1}y_{n-2} & u_{n-1}u_n & u_{n-1}u_{n-1} \end{pmatrix}$$

and the row vector m is defined as

$$m = \frac{1}{N}\sum_{n=1}^{N} y_n\,(\,y_{n-1} \quad y_{n-2} \quad u_n \quad u_{n-1}\,).$$

Assume that $y_{-n} = u_{-n} = 0, \forall n > 0$. The "shifted difference" of R_k is equal to

$$\frac{1}{N}\sum_{n=1}^{N}\left(\begin{pmatrix} y_{n-2}y_{n-2} & y_{n-2}u_n & y_{n-2}u_{n-1} \\ u_n y_{n-2} & u_n u_n & u_n u_{n-1} \\ u_{n-1}y_{n-2} & u_{n-1}u_n & u_{n-1}u_{n-1} \end{pmatrix} - \begin{pmatrix} y_{n-1}y_{n-1} & y_{n-1}y_{n-2} & y_{n-1}u_n \\ y_{n-2}y_{n-1} & y_{n-2}y_{n-2} & y_{n-2}u_n \\ u_n y_{n-1} & u_n y_{n-2} & u_n u_n \end{pmatrix}\right)$$

$$= \frac{1}{N}\begin{pmatrix} -y_{N-1}y_{N-1} & \sum_{n=1}^{N}(y_{n-2}u_n - y_{n-1}y_{n-2}) & -y_{N-1}u_N \\ \sum_{n=1}^{N}(u_n y_{n-2} - y_{n-2}y_{n-1}) & \sum_{n=1}^{N}(u_n u_n - y_{n-2}y_{n-2}) & \sum_{n=1}^{N}(u_n u_{n-1} - y_{n-2}u_n) \\ -u_N y_{N-1} & \sum_{n=1}^{N}(u_{n-1}u_n - u_n y_{n-2}) & -u_N u_N \end{pmatrix}.$$

No analytic procedure of decomposing the above "shifted difference" has been found so far for arbitrary input and output signals.

The primary interesting feature of the fast algorithms based on displacement representations is the potential saving in solving Toeplitz-like systems of linear equation. The necessary number of multiplication and addition operations is of order of αk^2 if the displacement representation of a Toeplitz-like matrix is available. However, as shown in the examples, not every problem has the analytic form of the desired displacement representation. Consequently, some numerical algorithms have to be used to decompose the "shifted difference," which usually takes arithmetic computations of $\mathbf{O}(k^3)$. As a result, the applicability of the fast algorithms to a problem depends on how many computations are required in the calcutation of the factorization (2.30).

2.4.3 QR-based Fast Algorithms

In recent years, a great effort had been devoted towards the development of QR-based fast parallel algorithms for digital signal processing [15][17][67]. These algorithms are better conditioned than the algorithms based on normal matrices. They also take full advantage of fast-developing parallel processing technology. However, these algorithms do not have either the feature of order-recursion or the lattice structure, which is preferred in many applications of signal processing [49][50][67]. This drawback has been very recently overcome by applying the QR-decomposition technique to inverting the prediction error matrices occurring in the Levinson algorithm [100]. The method used in reference [100] also suggests a general approach for improving the Levinson-type algorithms on numerical condition.

2.5 Summary

The importance of simultaneous estimation of ARX system order and parameters has been illustrated in adaptive control and adaptive IIR filtering. The primary interest of introducing both order and parameter estimation during the operation of an adaptive system comes from the fact that the estimation potentially provides a means of improving system performance in terms of robustness, convergence, and convergence rate, etc.. For many years, available order estimation schemes did not generate consistent order estimation for a closed-loop system. This situation has been changed recently by Chen et al. [28][29] and by Hemmerly and Davis [72]. However, their approach involves an extensive amount of computations and is difficult for on-line implementation unless some fast algorithms are developed. On the other hand, adaptive IIR filtering is expected to play a great role in adaptive signal processing and to have an increasingly wide applications. But the understanding of IIR filters is still very limited and the lack of fast algorithms for adaptive IIR filtering has not changed. As a result, fast algorithms for order and parameter estimation of control systems and IIR filters have to be developed.

Chapter 3

Fast Order Recursive Algorithms for LMMSE Parameter Estimation and Prediction of ARX Models

In this chapter, we devote our main efforts to developing a fast order-recursive algorithm (ORA) for determining optimal parameter estimates and output prediction of ARX models/IIR filters in the sense of the linear minimum mean-square error (LMMSE)[1]. We set up the LMMSE parameter estimation/prediction problem in a conventional way, with the difference that the Toeplitz structure of the Yule-Walker equations involved is subtly exploited. Consequently, the desired fast algorithm, featuring a recursion in the number of model parameters, can be developed. The preliminaries are expressed in Section 3.1, where the motivation for developing the ORA is discussed and the concept of a block-Toeplitz submatrix system (of linear equations) is introduced. The key ideas in the derivation of the ORA are illustrated in Section 3.2, including the introduction of three kinds of auxiliary LMMSE estimators and some observations on the block-Toeplitz

[1]An ARX model represents an autoregressive model with exogenous input.

submatrix systems involved. In addition, some extra advantages of the ORA brought by the auxiliary estimators are also explored there. The ORA is composed of two parallel parts: the recursive algorithm in the number of feedback parameters of an ARX model and the recursive algorithm in the number of feedforward parameters. The former is referred as to the ORA in p and the latter as the ORA in q. Section 3.3 and Section 3.4, respectively, present the algorithm description and computational complexity of the ORA in p and the ORA in q. Combining the results contained in Section 3.3 and Section 3.4 yields Section 3.5, where the ORA is discussed. To illustrate the operation procedure of the ORA, three numerical examples are reported in Section 3.6. Finally, this chapter is summarized in Section 3.7.

3.1 Preliminaries

Linear filtering theory is extensively utilized in a large variety of scientific applications[21][39][53][80][86][127][138]. Communication, control, geophysics, and economics are a few specific examples. Several basic and key problems in linear filtering theory use the linear minimum mean square error (LMMSE) criterion[53][80]. This is because, in comparison with other performance criteria such as the maximum a posteriori error and the maximum likelihood, the LMMSE criterion requires very little information on the statistics of the data, but often performs well. Furthermore, its optimal solution is simple to implement. The description and solutions of LMMSE prediction and parameter estimation problems closely depend on the model which is used to describe the data[102]. In this chapter, we mainly consider autoregressive models with exogenous inputs (ARX models), which are preferred in identification of control systems because of their "parsimonious parametrization"[18][39]. In the prediction of an ARMA process, an ARX model is often used as a predictor model of the ARMA process, where the exogenous input represents some available estimate of unmeasurable noise and the model noise is the prediction error[39][80].

An ARX model can be expressed as follows:

$$\mathbf{y}_n + \sum_{i=1}^{p} A_i \mathbf{y}_{n-i} = \sum_{j=0}^{q} C_j \mathbf{u}_{n-j} + \omega_n, \tag{3.1}$$

where A_i and C_j are respectively referred as to the *feedback parameters* and the *feedforward parameters* of the ARX model. The pair of integers (p, q) represents the model order within the order space $\mathcal{O}_{p_0,q_0}^{p^*,q^*}$ defined as

$$\mathcal{O}_{p_0,q_0}^{p^*,q^*} \triangleq \{(p,q) \mid p_0 \leq p \leq p^*;\ q_0 \leq q \leq q^*\}, \tag{3.2}$$

where each element $(p,q) \in \mathcal{O}_{p_0,q_0}^{p^*,q^*}$ represents the order of an ARX model[2]. The first and second components of order (p, q) represent, respectively, the number of feedback parameters of the AR (for autoregressive) part and the number of feedforward parameters of the moving-average (MA) part of the exogenous inputs. Hence, we will call integers p and q the AR order and MA order, respectively. $\mathbf{y}_n$ and $\mathbf{u}_n$ are zero-mean m-dimensional output and l-dimensional input processes, which are both available for measurement. And ω_n is a zero-mean noise process of dimension m.

For any given order (p, q), the LMMSE parameter estimation problem for an ARX model is equivalent to the following minimization problem:

$$\min_{A_i,\ C_j} \{E\|\mathbf{y}_n + \sum_{i=1}^{p} A_i \mathbf{y}_{n-i} - \sum_{j=0}^{q} C_j \mathbf{u}_{n-j}\|_2^2\}. \tag{3.3}$$

It is well known that the solution to problem (3.3) also gives the parameters of the corresponding LMMSE moving-average predictor with exogenous inputs (MAX predictor):

$$\mathbf{y}_{n|n-1}(p,q) = -\sum_{i=1}^{p} A_i^{p,q} \mathbf{y}_{n-i} + \sum_{j=0}^{q} C_j^{p,q} \mathbf{u}_{n-j},$$

where $\mathbf{y}_{n|n-1}(p,q)$ is the LMMSE prediction of $\mathbf{y}_n$ based on $\{\mathbf{y}_k\}_{k=n-p}^{n-1}$ and $\{\mathbf{u}_k\}_{k=n-q}^{n}$ [39]. If the output and input processes are jointly-stationary, the

[2]Here, we have made the convention, which will be used throughout, that if $a > b$, then $\sum_{i=a}^{b} f(i) = 0$ regardless of $f(i)$. Thus, problem (3.3) depends only on $\mathbf{y}_n$ and the moving-average of $(q+1)$ inputs if $p = 0$ or $\mathbf{y}_n$ and p lagged outputs if $q = -1$.

parameters of the LMMSE predictor are also the solution to the Yule-Walker equation

$$(\; I \; A_1^{p,q} \; \cdots \; A_p^{p,q} \; -C_0^{p,q} \; \cdots \; -C_q^{p,q} \;)\underline{R}^1(p,q) = (\; R^f(p,q) \; 0 \; \cdots \; 0 \; 0 \; \cdots \; 0 \;) \quad (3.4)$$

where the Yule-Walker matrix $\underline{R}^1(p,q)$ is denoted by

$$\underline{R}^1(p,q) \triangleq \left(\begin{array}{ccc|ccc} R_{yy}(0) & \cdots & R_{yy}(p) & R_{yu}(0) & \cdots & R_{yu}(q) \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ R_{yy}(-p) & \cdots & R_{yy}(0) & R_{yu}(-p) & \cdots & R_{yu}(q-p) \\ \hline R_{uy}(0) & \cdots & R_{uy}(p) & R_{uu}(0) & \cdots & R_{uu}(q) \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ R_{uy}(-q) & \cdots & R_{uy}(p-q) & R_{uu}(-q) & \cdots & R_{uu}(0) \end{array} \right) \quad (3.5)$$

and $\{R_{yy}(k), R_{uu}(k), R_{yu}(k), R_{uy}(k), k = 0, 1, 2, \cdots\}$ are a sequence of autocorrelation matrices (ACM's) and cross-correlation matrices (CCM's) of the output and input processes. It is worth noting that the Yule-Walker matrix $\underline{R}^1(p,q)$ is not a Toeplitz matrix. But, it can be partitioned in such a way that the resulting matrix consists of four submatrices, each of which is a block-Toeplitz matrix. That is, all block elements of the submatrices along any diagonals are constant. This special structure which will play a crucial role in the future. To describe the structure precisely, we introduce the following definition.

Definition 3.1 A matrix Γ is a submatrix-block-Toeplitz matrix if and only if it can be partitioned into:

$$\Gamma = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix},$$

where all four submatrices are block-Toeplitz matrices. That is, all block elements along any block diagonal are constant. Further, all linear equations

$$\Gamma x = b$$

with Γ being a submatrix-block-Toeplitz matrix will be called block-Toeplitz submatrix systems. ∎

According to this definition, the matrix $\underline{R}^1(p,q)$ defined in (3.5) is a submatrix-block-Toeplitz matrix because the matrix $\underline{R}^1(p,q)$ consisting of the ACM's and CCM's can be partitioned into four submatrices with the upper-left submatrix and the lower-right submatrix of dimension $(p+1)m \times (p+1)m$ and $(q+1)l \times (q+1)l$ and each of these four submatrices is block-Toeplitz. Therefore, the Yule-Walker equation (3.4) is a block-Toeplitz submatrix system. Note that a block-Toeplitz submatrix system is not necessarily a Yule-Walker equation because the blocks within the matrix Γ are not necessarily made up from ACM's and CCM's. Throughout this chapter, we will consider block-Toeplitz submatrix systems with a *symmetric* matrix Γ.

Of course, in the real world one often does not have exact values for the ACM's and CCM's. Numerous methods have been proposed to estimate the parameters of the LMMSE predictors when the second moments are not available [53][86][138]. Many of them are based on the equation error modeling approach or the method of moments, in which the parameter estimates of the LMMSE predictor are found by substituting estimates of the CCM's and ACM's into the Yule-Walker equation and solving the resulting equation. The resulting equation is a block-Toeplitz submatrix system when the popular correlation method is used to estimate the ACM's and CCM's[53][86]. On the other hand, many methods are based on performance criteria determined from statistics of the data. Two very successful examples are the least squares methods and the instrumental variable methods. Although their theoretical foundation and justifications are different from the method of moments, in computation least squares methods and instrumental variable methods could operate in the same way as the method of moments[53][86]. For example, the results computed by using the least squares methods are mathematically equal to solutions to the modified Yule-Walker equation with the ACM's and CCM's estimated by using the variance method. In fact, they are also very close to the solution to a block-Toeplitz submatrix system when the length of the data sequence is large[112]. As a result of the similarity of many useful methods in computation, fast algorithms for solving a block-Toeplitz submatrix system are of great importance for practical linear filtering.

3.2 LMMSE Predictors and Block-Toeplitz Submatrix Systems

3.2.1 The LMMSE Output Predictors and Auxiliary Estimators

Consider a finite family of LMMSE output predictors:

$$\mathbf{y}_{n|n-1}(p,q) = \sum_{i=1}^{p} -A_i^{p,q}\mathbf{y}_{n-i} + \sum_{j=0}^{q} C_j^{p,q}\mathbf{u}_{n-j}, \tag{3.6}$$

where the order $(p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}$. The key component of the ORA, as the name suggests, is the recursion between the parameters of the LMMSE predictor of order (p,q) and those of order $(p+1,q)$ or those of order $(p,q+1)$. It can be shown that this recursion comes from using the output backward predictors, as originally introduced in the Levinson algorithm [99], and introducing three kinds of auxiliary estimators. They are

output backward predictor

$$\mathbf{y}_{n-p|n}(p,q) = \sum_{i=1}^{p} -B_i^{p,q}\mathbf{y}_{n-p+i} + \sum_{j=1}^{q} D_j^{p,q}\mathbf{u}_{n-q+j}, \ (p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}, \tag{3.7}$$

input predictor

$$\mathbf{u}_{n|n-1}(p,q) = \sum_{i=1}^{p} H_i^{p,q}\mathbf{y}_{n-i} + \sum_{j=1}^{q} -G_j^{p,q}\mathbf{u}_{n-j}, \ (p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}, \tag{3.8}$$

input backward predictor

$$\mathbf{u}_{n-q|n}(p,q) = \sum_{i=0}^{p} F_i^{p,q}\mathbf{y}_{n-p+i} + \sum_{j=1}^{q} -E_j^{p,q}\mathbf{u}_{n-q+j}, \ (p,q) \in \mathcal{O}_{-1,0}^{p^*,q^*}, \tag{3.9}$$

where $A_i^{p,q} \in R^{m\times m}, C_j^{p,q} \in R^{m\times l}, B_i^{p,q} \in R^{m\times m}, D_j^{p,q} \in R^{m\times l}, H_i^{p,q} \in R^{l\times m}, G_j^{p,q} \in R^{l\times l}, F_i^{p,q} \in R^{l\times m}, E_j^{p,q} \in R^{l\times l}$, The parameter coefficients of the predictors in (3.6)- (3.9) have two superscripts and one subscript. The two superscripts are used to express the effect of the order (p,q). The subscript is an index. The corresponding forward and backward prediction errors of the forward and backward predictors in (3.6) - (3.9) are equal to

$$\begin{aligned} e_{n,p,q}^{f,y} &= y_n - y_{n|n-1}(p,q), & e_{n,0,-1}^{f,y} &= y_n, \\ e_{n,p,q}^{b,y} &= y_{n-p} - y_{n-p|n}(p,q), & e_{n,0,0}^{b,y} &= y_n, \\ e_{n,p,q}^{f,u} &= u_n - u_{n|n-1}(p,q), & e_{n,-1,0}^{f,u} &= u_n, \\ e_{n,p,q}^{b,u} &= u_{n-q} - u_{n-q|n}(p,q), & e_{n,-1,0}^{b,u} &= u_n. \end{aligned} \tag{3.10}$$

The primary purpose of introducing the above auxiliary estimators is to reduce the computational load in solving the LMMSE problem (3.3). However, these estimators could be useful in other applications. For instance, the measured u_n may be thought as the commanded input and the actually applied input, denoted by $\hat{u}_n$, may differ from the commanded input due to various actuator imperfections. This phenomenon may be mathematically described as

$$u_n = \hat{u}_n + v_n.$$

In the circumstance, the LMMSE input predictor generates the LMMSE estimation of the actual input $\hat{u}_n$. The input predictor could also be used to monitor the failure or malfunction of a controller. In economic systems, the measurement of real input $\hat{u}_n$ is often contaminated with some noise. In the case, the LMMSE input predictor can be used in noise cancellation.

Like the backward output predictors introduced in the Levinson algorithm[99], the backward output and input predictors defined in Eqs.(3.7) and (3.9) generate an orthogonal basis for a space spanned by the past signals, which is very useful in joint estimation[66]. For simplicity, let us consider a scalar case. Denote by $H(y_n \ \cdots \ y_{n-p} \ u_n \ \cdots \ u_{n-q})$ a Hilbert space composed of all linear combinations of the stationary random variables $y_n \ \cdots \ y_{n-p} \ u_n \ \cdots \ u_{n-q}$.[3] By the orthogonality principle[39], we have that

$$e^{b,y}_{n,p,q} \perp H(y_n \ \cdots \ y_{n-p+1} \ u_n \ \cdots \ u_{n-q+1}), p \geq 0, q \geq 0 \tag{3.11}$$

and

$$e^{b,y}_{n,p,q} \in H(y_n \ \cdots \ y_{n-p+1} \ y_{n-p} \ u_n \ \cdots \ u_{n-q+1}), p \geq 0, q \geq 0. \tag{3.12}$$

Similarly, it is easy to see

$$e^{b,u}_{n,p,q} \perp H(y_n \ \cdots \ y_{n-p} \ u_n \ \cdots \ u_{n-q+1}), p \geq 0, q \geq 0 \tag{3.13}$$

and

$$e^{b,u}_{n,p,q} \in H(y_n \ \cdots \ y_{n-p} \ u_n \ \cdots \ u_{n-q+1} \ u_{n-q}), p \geq 0, q \geq 0. \tag{3.14}$$

Therefore, $\{e^{b,y}_{n,i,0}\}^p_{i=0}$ forms an orthogonal basis of the space $H(y_n \ \cdots \ y_{n-p})$ and, furthermore, $\{\{e^{b,y}_{n,i,0}\}^p_{i=0}, \{e^{b,u}_{n,p,j}\}^q_{j=0}\}$ becomes an orthogonal sequence which spans the space $H(y_n \ \cdots \ y_{n-p+1} \ y_{n-p} \ u_n \ \cdots \ u_{n-q})$. This is also true for random vectors $\mathbf{y}_n$ and $\mathbf{u}_n$. (For details, see reference[39].)

3.2.2 Combined Yule-Walker Equations

It is well known that for any given order $(p,q) \in \mathcal{O}^{p^*,q^*}_{0,0}$, the parameters and estimation error matrices of the LMMSE estimators in (3.6)-(3.9) are, respectively, the solutions of the following four Yule-Walker equations:

$$(\ I \ A^{p,q}_1 \ \ldots \ A^{p,q}_p \ -C^{p,q}_0 \ \ldots \ -C^{p,q}_q \ )\underline{R}^1(p,q) = (\ R^f(p,q) \ 0 \ \ldots \ 0 \ 0 \ \ldots \ 0\ ) \tag{3.15}$$

$$(\ B^{p,q}_p \ \ldots \ B^{p,q}_1 \ I \ -D^{p,q}_q \ \ldots \ -D^{p,q}_1 \ )\underline{R}^2(p,q) = (\ 0 \ \ldots \ 0 \ R^b(p,q) \ 0 \ \ldots \ 0\ ) \tag{3.16}$$

[3]Based on the definition of the space $H(\cdot)$, it is easy to see

$$\begin{array}{ll} H(y_n \ y_{n+1} \ u_n \ \cdots \ u_{n-q}) = H(u_n \ \cdots \ u_{n-q}) & \text{for whatever } q, \\ H(y_n \ \cdots \ y_{n-p} \ u_n \ u_{n+1}) = H(y_n \ \cdots \ y_{n-p}) & \text{for whatever } p, \\ H(u_n \ \ u_{n+1}) = \{0\}, & \\ H(y_n \ \ y_{n+1}) = \{0\}. & \end{array}$$

$$(-H_1^{p,q} \ \ldots \ -H_p^{p,q} \ I \ G_1^{p,q} \ \ldots \ G_q^{p,q} \ )\underline{R}^3(p,q) = (\ 0 \ \ldots \ 0 \ V^f(p,q) \ 0 \ \ldots \ 0 \ ) \quad (3.17)$$

$$(-F_p^{p,q} \ \ldots \ -F_0^{p,q} \ E_q^{p,q} \ \ldots \ E_1^{p,q} \ I \ )\underline{R}^4(p,q) = (\ 0 \ \ldots \ 0 \ 0 \ \ldots \ 0 \ V^b(p,q) \ ) \quad (3.18)$$

where

$$\begin{aligned}
\underline{R}^1(p,q) &= E(\mathbf{y}_{n-p}^{n^T} \ \mathbf{u}_{n-q}^{n^T})^T(\mathbf{y}_{n-p}^{n^T} \ \mathbf{u}_{n-q}^{n^T}),\\
\underline{R}^2(p,q) &= E(\mathbf{y}_{n-p}^{n^T} \ \mathbf{u}_{n-q+1}^{n^T})^T(\mathbf{y}_{n-p}^{n^T} \ \mathbf{u}_{n-q+1}^{n^T}),\\
\underline{R}^3(p,q) &= E(\mathbf{y}_{n-p}^{n-1^T} \ \mathbf{u}_{n-q}^{n^T})^T(\mathbf{y}_{n-p}^{n-1^T} \ \mathbf{u}_{n-q}^{n^T}),\\
\underline{R}^4(p,q) &= E(\mathbf{y}_{n-p}^{n^T} \ \mathbf{u}_{n-q}^{n^T})^T(\mathbf{y}_{n-p}^{n^T} \ \mathbf{u}_{n-q}^{n^T}) = \underline{R}^1(p,q),\\
\mathbf{y}_{n-p}^n &= (\mathbf{y}_n^T \ \mathbf{y}_{n-1}^T \ \cdots \ \mathbf{y}_{n-p}^T)^T,\\
\mathbf{u}_{n-p}^n &= (\mathbf{u}_n^T \ \mathbf{u}_{n-1}^T \ \cdots \ \mathbf{u}_{n-q}^T)^T,
\end{aligned}$$

The combined Yule-Walker equation: To find the recursive relation between the output predictor of order (p,q) and that of order $(p+1,q)$, which we call the order-recursion in p(-direction), let us merge the Yule-Walker equations in (3.6)-(3.9) into one equation, which we call the *combined Yule-Walker equation*, by introducing six more unknown matrices $\alpha(p,q) \in \mathcal{R}^{m\times m}$, $\beta(p,q) \in \mathcal{R}^{m\times m}$, $\delta(p,q) \in \mathcal{R}^{m\times l}$, $\eta(p,q) \in \mathcal{R}^{l\times m}$, $\zeta(p,q) \in \mathcal{R}^{l\times m}$, and $\lambda(p,q) \in \mathcal{R}^{l\times m}$:

$$\begin{pmatrix}
I & A_1^{p,q} \ \cdots \ A_p^{p,q} & 0 & -C_0^{p,q} & -C_1^{p,q} \ \cdots \ -C_q^{p,q}\\
0 & B_p^{p,q} \ \cdots \ B_1^{p,q} & I & 0 & -D_q^{p,q} \ \cdots \ -D_1^{p,q}\\
0 & -H_1^{p,q} \ \cdots \ -H_p^{p,q} & 0 & I & G_1^{p,q} \ \cdots \ G_q^{p,q}\\
-F_p^{p,q} & -F_{p-1}^{p,q} \ \cdots \ -F_0^{p,q} & 0 & E_q^{p,q} & E_{q-1}^{p,q} \ \cdots \ I
\end{pmatrix} \times \Gamma(p+1,q)$$

$$= \begin{pmatrix}
R^f(p,q) & \overbrace{0 \ \cdots \ 0}^{p} & \alpha(p,q) & 0 & \overbrace{0 \ \cdots \ 0}^{q}\\
\beta(p,q) & 0 \ \cdots \ 0 & R^b(p,q) & \delta(p,q) & 0 \ \cdots \ 0\\
\eta(p,q) & 0 \ \cdots \ 0 & \zeta(p,q) & V^f(p,q) & 0 \ \cdots \ 0\\
0 & 0 \ \cdots \ 0 & \lambda(p,q) & 0 & 0 \cdots V^b(p,q)
\end{pmatrix}, \quad (3.19)$$

where the matrix $\Gamma(p+1,q) = \begin{pmatrix} T_{11}(p+1,q) & T_{12}(p+1,q) \\ T_{21}(p+1,q) & T_{22}(p+1,q) \end{pmatrix}$ with its submatrices being

$$T_{11}(p+1,q) = \begin{pmatrix} R_{yy}(0) & R_{yy}(1) & \cdots & R_{yy}(p) & R_{yy}(p+1) \\ R_{yy}(-1) & R_{yy}(0) & \cdots & R_{yy}(p-1) & R_{yy}(p) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{yy}(-p) & R_{yy}(-p+1) & \cdots & R_{yy}(0) & R_{yy}(1) \\ R_{yy}(-p-1) & R_{yy}(-p) & \cdots & R_{yy}(-1) & R_{yy}(0) \end{pmatrix}$$

$$T_{12}(p+1,q) = \begin{pmatrix} R_{yu}(0) & R_{yu}(1) & \cdots & R_{yu}(q-1) & R_{yu}(q) \\ R_{yu}(-1) & R_{yu}(0) & \cdots & R_{yu}(q-2) & R_{yu}(q-1) \\ \vdots & \vdots & & \vdots & \vdots \\ R_{yu}(-p) & R_{yu}(1-p) & \cdots & R_{yu}(q-p-1) & R_{yu}(q-p) \\ R_{yu}(-p-1) & R_{yu}(-p) & \cdots & R_{yu}(q-p-2) & R_{yu}(q-p-1) \end{pmatrix}$$

$$T_{21}(p+1,q) = \begin{pmatrix} R_{uy}(0) & R_{uy}(1) & \cdots & R_{uy}(p) & R_{uy}(p+1) \\ R_{uy}(-1) & R_{uy}(0) & \cdots & R_{uy}(p-1) & R_{uy}(p) \\ \vdots & \vdots & & \vdots & \vdots \\ R_{uy}(1-q) & R_{uy}(2-q) & \cdots & R_{uy}(p-q+1) & R_{uy}(p-q+2) \\ R_{uy}(-q) & R_{uy}(1-q) & \cdots & R_{uy}(p-q) & R_{uy}(p-q+1) \end{pmatrix}$$

$$T_{22}(p+1,q) = \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \cdots & R_{uu}(q-1) & R_{uu}(q) \\ R_{uu}(-1) & R_{uu}(0) & \cdots & R_{uu}(q-2) & R_{uu}(q-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{uu}(1-q) & R_{uu}(2-q) & \cdots & R_{uu}(0) & R_{uu}(1) \\ R_{uu}(-q) & R_{uu}(1-q) & \cdots & R_{uu}(-1) & R_{uu}(0) \end{pmatrix}.$$

Eq.(3.19) yields immediately Eqs.(3.15)–(3.18) and the following equations:

$$\alpha(p,q) = \sum_{i=0}^{p} A_i^{p,q} R_{yy}(p+1-i) - \sum_{i=0}^{q} C_i^{p,q} R_{uy}(p+1-i), \tag{3.20}$$

$$\beta(p,q) = \sum_{i=0}^{p} B_{p-i}^{p,q} R_{yy}(-i-1) - \sum_{i=0}^{q-1} D_{q-i}^{p,q} R_{uy}(-i-1), \tag{3.21}$$

$$\delta(p,q) = \sum_{i=0}^{p} B_{p-i}^{p,q} R_{yu}(-i-1) - \sum_{i=0}^{q-1} D_{q-i}^{p,q} R_{uu}(-i-1), \tag{3.22}$$

$$\lambda(p,q) = -\sum_{i=0}^{p} F_{p-i}^{p,q} R_{yy}(p+1-i) + \sum_{i=0}^{q} E_{q-i}^{p,q} R_{uy}(p+1-i), \tag{3.23}$$

$$\eta(p,q) = -\sum_{i=1}^{p} H_i^{p,q} R_{yy}(-i) + \sum_{i=0}^{q} G_i^{p,q} R_{uy}(-i), \tag{3.24}$$

$$\zeta(p,q) = -\sum_{i=1}^{p} H_i^{p,q} R_{yy}(p+1-i) + \sum_{i=0}^{q} G_i^{p,q} R_{uy}(p+1-i), \tag{3.25}$$

and

$$R^f(p,q) = \sum_{i=0}^{p} A_i^{p,q} R_{yy}(-i) - \sum_{i=0}^{q} C_i^{p,q} R_{uy}(-i), \tag{3.26}$$

$$R^b(p,q) = \sum_{i=0}^{p} B_{p-i}^{p,q} R_{yy}(p-i) - \sum_{i=0}^{q-1} D_{q-i}^{p,q} R_{uy}(p-i), \tag{3.27}$$

$$V^b(p,q) = -\sum_{i=0}^{p} F_{p-i}^{p,q} R_{yu}(q-i) + \sum_{i=0}^{q} E_{q-i}^{p,q} R_{uu}(q-i), \tag{3.28}$$

$$V^f(p,q) = -\sum_{i=1}^{p} H_i^{p,q} R_{yu}(-i) + \sum_{i=0}^{q} G_i^{p,q} R_{uu}(-i), \tag{3.29}$$

where $A_0^{p,q} = I, B_0^{p,q} = I, E_0^{p,q} = I, G_0^{p,q} = I$. Therefore, the existence of the solution to Eqs.(3.6)-(3.9) is equivalent to that of Eq.(3.15) and the solution to Eqs.(3.6)-(3.9) is also a part (submatrix) of the solution (matrix) to Eq.(3.15). Notice from the partition of the matrix $\Gamma(p+1,q)$ in (3.19) that the combined Yule-Walker equation is also a block-Toeplitz submatrix system.

By carefully checking the matrix $\Gamma(p+1,q)$ in Eq.(3.15), we can see that the block-Toeplitz property of the four submatrices of $\Gamma(p+1,q)$ leads to the following crucial observation:

$$\begin{pmatrix} I & A_1^{p,q} & \cdots & A_p^{p,q} & -C_0^{p,q} & -C_1^{p,q} & \cdots & -C_q^{p,q} \\ B_p^{p,q} & B_{p-1}^{p,q} & \cdots & I & -D_q^{p,q} & -D_{q-1}^{p,q} & \cdots & 0 \\ 0 & -H_1^{p,q} & \cdots & -H_p^{p,q} & I & G_1^{p,q} & \cdots & G_q^{p,q} \\ -F_p^{p,q} & -F_{p-1}^{p,q} & \cdots & -F_0^{p,q} & E_q^{p,q} & E_{q-1}^{p,q} & \cdots & I \end{pmatrix} \times \Gamma(p,q)$$
$$= \begin{pmatrix} R^f(p,q) & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & R^b(p,q) & 0 & 0 & \cdots & \nu(p,q) \\ \eta(p,q) & 0 & \cdots & 0 & V^f(p,q) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & V^b(p,q) \end{pmatrix} \tag{3.30}$$

where the extra unknown $\nu(p,q)$ is expressed as

$$\nu(p,q) = \sum_{i=0}^{p} B_{p-i}^{p,q} R_{yu}(q-1-i) + \sum_{i=0}^{q-1} D_{q-i}^{p,q} R_{uu}(q-1-i). \tag{3.31}$$

The combined Yule-Walker equation for deriving the ORA in q: Similarly, to find the recursive relation between LMMSE output predictor of order (p, q) described in (3.6) and that of order $(p, q+1)$, which we call the order-recursion in q(-direction), let us combine the Yule-Walker equations described in (3.15) – (3.18) into one equation, which we also call the combined Yule-Walker equation, by introducing six more unknown matrices $\gamma(p,q) \in \mathcal{R}^{m\times l}$, $\nu(p,q+1) \in \mathcal{R}^{m\times l}$, $\eta(p,q) \in \mathcal{R}^{l\times m}$, $\kappa(p,q) \in \mathcal{R}^{l\times l}$, $\mu(p-1,q) \in \mathcal{R}^{l\times m}$, and $\xi(p-1,q) \in \mathcal{R}^{l\times l}$:

$$\begin{pmatrix} I & A_1^{p,q} \cdots A_p^{p,q} & -C_0^{p,q} & -C_1^{p,q} \cdots -C_q^{p,q} & 0 \\ B_p^{p,q+1} & B_{p-1}^{p,q+1} \cdots I & -D_{q+1}^{p,q+1} & -D_q^{p,q+1} \cdots -D_1^{p,q+1} & 0 \\ 0 & -H_1^{p,q} \cdots -H_p^{p,q} & I & G_1^{p,q} \cdots G_q^{p,q} & 0 \\ 0 & -F_{p-1}^{p-1,q} \cdots -F_0^{p-1,q} & 0 & E_q^{p-1,q} \cdots E_1^{p-1,q} & I \end{pmatrix} \times \Gamma(p,q+1)$$

$$= \begin{pmatrix} R^f(p,q) & \overbrace{0\cdots 0}^{p} & 0 & \overbrace{0\cdots 0}^{q} & \gamma(p,q) \\ 0 & 0\cdots R^b(p,q+1) & 0 & 0\cdots 0 & \nu(p,q+1) \\ \eta(p,q) & 0\cdots 0 & V^f(p,q) & 0\cdots 0 & \kappa(p,q) \\ \mu(p-1,q) & 0\cdots 0 & \xi(p-1,q) & 0\cdots 0 & V^b(p-1,q) \end{pmatrix} \tag{3.32}$$

Eq.(3.32) implies Eqs.(3.15)— (3.18), and

$$\gamma(p,q) = \sum_{i=0}^{p} A_i^{p,q} R_{yu}(q+1-i) - \sum_{j=0}^{q} C_j^{p,q} R_{uu}(q+1-j) \tag{3.33}$$

$$\kappa(p,q) = -\sum_{i=1}^{p} H_i^{p,q} R_{yu}(q-i) + \sum_{j=0}^{q} G_j^{p,q} R_{uu}(q+1-j) \tag{3.34}$$

$$\mu(p-1,q) = -\sum_{i=0}^{p-1} F_{p-1-i}^{p-1,q} R_{yy}(-i-1) + \sum_{j=0}^{q} E_{q-i}^{p-1,q} R_{uy}(-j-1) \tag{3.35}$$

$$\xi(p-1,q) = -\sum_{i=0}^{p-1} F_{p-1-i}^{p-1,q} R_{yu}(-i-1) + \sum_{j=0}^{q} E_{q-j}^{p-1,q} R_{uu}(-j-1) \tag{3.36}$$

According to Definition 3.1, Eq.(3.32) is also a block-Toeplitz submatrix system. Comparing Eqs.(3.22) and (3.34) with Eqs.(3.25) and (3.36) and using Eqs.(3.17) and (3.22) yields

$$\delta(p,q) = \zeta^T(p,q) \quad \text{and} \quad \xi(p-1,q) = \kappa^T(p,q). \tag{3.37}$$

By carefully checking (3.32), we can see that the block-Toeplitz structure of the four submatrices of $\Gamma(p,q+1)$ results in the following key equation for deriving the recursive relation we want:

$$\begin{pmatrix} I & A_1^{p,q}\cdots A_p^{p,q} & -C_0^{p,q} & -C_1^{p,q}\cdots - C_{q-1}^{p,q} & -C_q^{p,q} \\ B_p^{p,q+1} & B_{p-1}^{p,q+1}\cdots I & -D_{q+1}^{p,q+1} & -D_q^{p,q+1}\cdots - D_2^{p,q+1} & -D_1^{p,q+1} \\ 0 & -H_1^{p,q}\cdots - H_p^{p,q} & I & G_1^{p,q}\cdots G_{q-1}^{p,q} & G_q^{p,q} \\ -F_{p-1}^{p-1,q} & -F_{p-2}^{p-1,q}\cdots 0 & E_q^{p-1,q} & E_{q-1}^{p-1,q}\cdots E_1^{p-1,q} & I \end{pmatrix} \times\Gamma(p,q)$$

$$= \begin{pmatrix} R^f(p,q) & \overbrace{0\cdots 0}^{p-1} & 0 & 0 & \overbrace{0\cdots 0}^{q-1} & 0 \\ 0 & 0\cdots 0 & R^b(p,q+1) & 0 & 0\cdots 0 & 0 \\ \eta(p,q) & 0\cdots 0 & 0 & V^f(p,q) & 0\cdots 0 & 0 \\ 0 & 0\cdots 0 & \lambda(p-1,q) & 0 & 0\cdots 0 & V^b(p-1,q) \end{pmatrix} \tag{3.38}$$

where $\lambda(p-1,q)$ is defined in (3.23).

3.2.3 Some Useful Properties of the Combined Yule-Walker Equations

Some useful properties of the block-Toeplitz submatrix system in Eq.(3.19) are described below. Before expressing them, let us define four normal matrices:

Definition 3.2 From Eqs.(3.15)–(3.18) or from (3.30) and (3.38), one can deduce four normal matrices $R^1(p,q)$, $R^2(p,q)$, $R^3(p,q)$, and $R^4(p,q)$,where

$R^1(p,q) \triangleq$ a matrix generated from $\Gamma(p,q)$ by deleting the first (block) row and (block) column of $\Gamma(p,q)$,

$R^2(p,q) \triangleq$ a matrix generated from $\Gamma(p,q)$ by deleting the $(p+1)$th and last (block) rows and (block) columns of $\Gamma(p,q)$,

$R^3(p,q) \triangleq$ a matrix generated from $\Gamma(p,q)$ by deleting the first and $(p+2)$th (block) rows and (block) columns of $\Gamma(p,q)$,

$R^4(p,q) \triangleq$ a matrix generated from $\Gamma(p,q)$ by deleting the last (block) row and (block) column of $\Gamma(p,q)$. ∎

Hence, we can express a sufficient condition for well posedness of the proposed ORA.

Theorem 3.1 If symmetric matrices $\Gamma(p,q)$ of the block-Toeplitz submatrix systems in (3.19) are nonsingular for any order $(p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}$, then for any order $(p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}$,

(i) the matrices

$$\begin{pmatrix} R^b(p,q) & \delta(p,q) \\ \delta^T(p,q) & V^f(p,q) \end{pmatrix}, \begin{pmatrix} V^f(p,q) & \kappa(p,q) \\ \kappa^T(p,q) & V^b(p-1,q) \end{pmatrix}, \begin{pmatrix} R^f(p,q) & 0 \\ \eta(p,q) & V^f(p,q) \end{pmatrix}$$

are nonsingular[4],

(ii) the matrices$R^1(p,q)$, $R^2(p,q)$, $R^3(p,q)$, and $R^4(p,q)$ are nonsingular.

∎

Corollary 3.1 Suppose that the matrix $\Gamma(p^*,q^*)$ is a Yule-Walker matrix. If $\Gamma(p^*,q^*)$ is positive definite, then all the symmetric matrices in Theorem 3.1 are positive definite. ∎

Proof: Notice the special positions of the zero and identity matrices on the left sides of Eqs.(3.19) and (3.32) and the nonzero matrices on the right sides. Then proposition (i) comes immediately from the following property of a nonsingular symmetric matrix Γ: $x^T\Gamma x \neq 0$ for any non-zero vector x of proper dimension. Proposition (ii) holds since $R^i(p,q), i = 1,2,3,4$, are nested submatrices of $\Gamma(p,q)$. Corollary 3.1 can be proved in the same way. ∎

Remark 3.1 As shown in [111][114], the nonsingularity of the matrix $R^1(p^*,q^*)$ and all the matrices in Theorem 3.1 for any order $(p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}/(p^*,q^*)$ is needed to ensure that the fast solution of a Toeplitz submatrix system of order (p^*,q^*) in (3.15) can be obtained via the ORA. This fact reflects the cost of achieving computational efficiency. One only needs to assume that the normal matrix $R^1(p^*,q^*)$ is nonsingular for the uniqueness of the solution to Eq.(3.15). ∎

Remark 3.2 The conditions in Theorem 1 are also related to the persistence of excitation [138]. ∎

[4]This implies that the matrices $R^f(p,q)$, $R^b(p,q)$, $V^f(p,q)$, $V^b(p,q)$, are nonsingular.

3.3 The Order-Recursive Algorithm in p

In this section, we intend to develop a fast algorithm, which we call the order-recursive algorithm in p (ORA in p), for fast solution of a Yule-Walker equation (described in (3.15)) of order $(p, q) \in \mathcal{O}_{p_0,q}^{p^*,q} \subset \mathcal{O}_{0,-1}^{p^*,q}$. The only required assumption is that the symmetric matrices $\Gamma(p, q)$ in (3.19) are nonsingular for all $(p, q) \in \mathcal{O}_{p_0,q}^{p^*,q}$. The idea for developing the ORA in p is very simple. It is nothing but introducing four linear combinations of the four block rows on the left side of Eq.(3.19) such that each of the combinations is equal to one block row on the left side of Eq.(3.30) of order $(p + 1, q)$. This is possible when the matrix $\Gamma(p + 1, q)$ is nonsingular. The ORA in p is presented in Section 3.3.1, where the computational complexity of the ORA in p is also given. Subsequently, a simplified version of the ORA in p is expressed when a Yule-Walker equation has some extra structure. In Section 3.3.3, the lattice implementation of LMMSE MAX predictor is illustrated.

3.3.1 The General Version of the ORA in p for LMMSE Parameter Estimation

The aim of this section is to describe the ORA in p, which is derived in Appendix A.2. Before describing the ORA in p, we need to introduce some fictitious parameters of the estimators defined in (3.6) – (3.9) such that we can have a compact expression of the ORA in p.

Definition 3.3 The fictitious parameters of the LMMSE estimators are defined as the matrix variables which have no effect on the LMMSE estimators, but have the same notation as the estimator parameters which are defined as a solution to Eqs.(3.15)–(3.18). The values of the fictitious parameters are assigned in the following way:

1. for any order $(p, q) \in \mathcal{O}_{0,-1}^{p^*,q^*}$,
   $A_0^{p,q} = I, A_i^{p,q} = 0, \forall i > p,$ and $C_j^{p,q} = 0, \forall j > q,$

2. for any order $(p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}$,

$B_0^{p,q} = I, B_i^{p,q} = 0, \forall i > p, D_0^{p,q} = 0,$ and $D_j^{p,q} = 0, \forall j > q,$

3. for any order $(p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}$,

$E_0^{p,q} = I, E_j^{p,q} = 0, \forall j > q, F_{-1}^{p,q} = 0,$ and $F_i^{p,q} = 0, \forall i > p,$

4. for any order $(p,q) \in \mathcal{O}_{-1,0}^{p^*,q^*}$,

$H_0^{p,q} = 0, H_i^{p,q} = 0, \forall i > p, G_0^{p,q} = I,$ and $G_j^{p,q} = 0, \forall j > q.$

∎

The ORA in p generates the LMMSE parameter estimates of an ARX model of order (p^*, q) recursively in the number of feedback parameters. It is composed of three parts: the parameter recursion, calculator of the partial correlation coefficients, and calculator of the intermediate variables, which are described as follows:

Algorithm ORA-in-p $(p_0, p^*, q, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$

Parameter recursion:

$$
\begin{aligned}
A_i^{p+1,q} &= A_i^{p,q} + K_2^{p,q} B_{p+1-i}^{p,q} - K_3^{p,q} H_i^{p,q}, & i &= 1, \cdots, p+1, \\
C_j^{p+1,q} &= C_j^{p,q} - K_2^{p,q} D_{q+1-j}^{p,q} - K_3^{p,q} G_j^{p,q}, & j &= 0, \cdots, q, \\
B_i^{p+1,q} &= B_i^{p,q} + L_1^{p,q} A_{p+1-i}^{p,q} + L_3^{p,q} H_{p+1-i}^{p,q} - L_4^{p,q} F_{i-1}^{p,q}, & i &= 1, \cdots, p+1, \\
D_j^{p+1,q} &= D_j^{p,q} + L_1^{p,q} C_{q-j}^{p,q} - L_3^{p,q} G_{q-j}^{p,q} - L_4^{p,q} E_j^{p,q}, & j &= 1, \cdots, q, \\
H_i^{p+1,q} &= H_i^{p,q} - M_2^{p,q} B_{p+1-i}^{p,q}, & i &= 1, \cdots, p+1, \\
G_j^{p+1,q} &= G_j^{p,q} - M_2^{p,q} D_{q+1-j}^{p,q}, & j &= 1, \cdots, q, \\
F_i^{p+1,q} &= F_{i-1}^{p,q} - N_2^{p,q} B_i^{p+1,q}, & i &= 0, \cdots, p+1, \\
E_j^{p+1,q} &= E_j^{p,q} - N_2^{p,q} D_j^{p+1,q}, & j &= 1, \cdots, q,
\end{aligned}
\tag{3.39}
$$

where the variable p varies from p_0 up to $p^* - 1$.

Calculator of partial correlation coefficients: The partial correlation coefficients used in the parameter recursion are calculated in the following way:

$$\begin{aligned}
K_2^{p,q} &= \alpha(p,q)\{\delta(p,q)[V^f(p,q)]^{-1}\delta^T(p,q) - R^b(p,q)\}^{-1}, && K_2^{p,q} \in \mathcal{R}^{m\times m} \\
K_3^{p,q} &= -K_2^{p,q}\delta(p,q)[V^f(p,q)]^{-1}, && K_3^{p,q} \in \mathcal{R}^{m\times l} \\
L_1^{p,q} &= \{\delta(p,q)[V^f(p,q)]^{-1}\eta(p,q) - \beta(p,q)\}[R^f(p,q)]^{-1}, && L_1^{p,q} \in \mathcal{R}^{m\times m} \\
L_3^{p,q} &= -\delta(p,q)[V^f(p,q)]^{-1}, && L_3^{p,q} \in \mathcal{R}^{m\times l} \\
L_4^{p,q} &= D_1^{p,q} + L_1^{p,q}C_q^{p,q} - L_3^{p,q}G_q^{p,q}, && L_4^{p,q} \in \mathcal{R}^{m\times l} \\
M_2^{p,q} &= -\delta^T(p,q)[R^b(p,q)]^{-1}, && M_2^{p,q} \in \mathcal{R}^{l\times m} \\
N_2^{p,q} &= -\lambda(p,q)[R^b(p+1,q)]^{-1}, && N_2^{p,q} \in \mathcal{R}^{l\times m},
\end{aligned} \tag{3.40}$$

where the variable p varies from p_0 up to $p^* - 1$.

Calculator of intermediate variables:

$$\begin{aligned}
R^f(p+1,q) &= R^f(p,q) + K_2^{p,q}\beta(p,q) + K_3^{p,q}\eta(p,q), && p = p_0,\cdots,p^*-1, \\
R^b(p+1,q) &= R^b(p,q) + L_1^{p,q}\alpha(p,q) + L_3^{p,q}\delta^T(p,q) + L_4^{p,q}\lambda(p,q), && p = p_0,\cdots,p^*-1, \\
V^f(p+1,q) &= V^f(p,q) + M_2^{p,q}\delta(p,q), && p = p_0,\cdots,p^*-1, \\
V^b(p+1,q) &= V^b(p,q) + N_2^{p,q}L_4^{p,q}V^b(p,q), && p = p_0,\cdots,p^*-1, \\
\eta(p+1,q) &= \eta(p,q) + M_2^{p,q}\beta(p,q), && p = p_0,\cdots,p^*-2, \\
\alpha(p,q) &= \sum_{i=0}^{p} A_i^{p,q}R_{yy}(p+1-i) - \sum_{i=0}^{q} C_i^{p,q}R_{uy}(p+1-i), && p = p_0,\cdots,p^*-1, \\
\beta(p,q) &= \sum_{i=0}^{p} B_{p-i}^{p,q}R_{yy}(-i-1) - \sum_{i=0}^{q-1} D_{q-i}^{p,q}R_{uy}(-i-1), && p = p_0,\cdots,p^*-1, \\
\delta(p,q) &= \sum_{i=0}^{p} B_{p-i}^{p,q}R_{yu}(-i-1) - \sum_{i=0}^{q-1} D_{q-i}^{p,q}R_{uu}(-i-1), && p = p_0,\cdots,p^*-1, \\
\lambda(p,q) &= -\sum_{i=0}^{p} F_{p-i}^{p,q}R_{yy}(p+1-i) + \sum_{i=0}^{q} E_{q-i}^{p,q}R_{uy}(p+1-i), && p = p_0,\cdots,p^*-1.
\end{aligned} \tag{3.41}$$

Initial condition at order $(p_0, q_0) = (p_0, q)$:

$$\begin{pmatrix}
A_1^{p_0,q} & \cdots & A_p^{p_0,q} & -C_0^{p_0,q} & \cdots & -C_q^{p_0,q} & R^f(p_0,q) \\
B_{p_0}^{p_0,q} & \cdots & B_1^{p_0,q} & -D_q^{p_0,q} & \cdots & -D_1^{p_0,q} & R^b(p_0,q) \\
-H_1^{p_0,q} & \cdots & -H_{p_0}^{p_0,q} & G_1^{p_0,q} & \cdots & G_q^{p_0,q} & V^f(p_0,q) \\
-F_{p_0}^{p_0,q} & \cdots & -F_0^{p_0,q} & E_q^{p_0,q} & \cdots & E_{q-1}^{p_0,q} & V^b(p_0,q)
\end{pmatrix} \tag{3.42}$$

Remark 3.3 The ORA in p is dealing with LMMSE parameter estimation of *a family* of ARX models of order $(p,q) \in \mathcal{O}^{p^*,q}_{p_0,q}$. To guarantee that the ORA in p works properly, all the matrices $\Gamma(p,q), (p,q) \in \mathcal{O}^{p^*,q}_{p_0,q}$ associated with the LMMSE parameter estimation problems in question are required to be nonsingular. Otherwise, the partial correlation coefficients could blow up. ∎

Remark 3.4 The ORA in p generates the parameters and prediction error variance (PEV) of the LMMSE output predictor (defined in (3.6)) of order (p,q) recursively as p is increased from p_0 up to p^*. As will be shown, this approach is an efficient approach computationally. ∎

Remark 3.5 In obtaining the parameters and PEV of the LMMSE output predictor of order (p^*,q) one actually computes the parameters and PEV's of all the LMMSE output predictors of order (p,q), $p_0 \leq p \leq p^*$. ∎

Remark 3.6 The by-products of the above computation process are the parameters and PEV's, of the three auxiliary estimators of order (p,q), $p_0 \leq p \leq q^*-1$, which are defined in (3.7)-(3.9). ∎

Lattice Version

To have a lattice version of the parameter recursion, introduce the following matrix polynomials in z:

$$\begin{aligned}
A_{p,q}(z) &= A_0^{p,q} + A_1^{p,q} z + \cdots + A_p^{p,q} z^p, & A_0^{p,q} &= I \\
C_{p,q}(z) &= C_0^{p,q} + C_1^{p,q} z + \cdots + C_q^{p,q} z^q \\
B_{p,q}(z) &= B_p^{p,q} + B_{p-1}^{p,q} z + \cdots + B_1^{p,q} z^{p-1} + B_0^{p,q} z^p, & B_0^{p,q} &= I \\
D_{p,q}(z) &= D_q^{p,q} + D_{q-1}^{p,q} z + \cdots + D_1^{p,q} z^{q-1} + D_0^{p,q} z^q, & D_0^{p,q} &= 0 \\
H_{p,q}(z) &= H_0^{p,q} + H_1^{p,q} z + H_2^{p,q} z^2 + \cdots + H_p^{p,q} z^p, & H_0^{p,q} &= 0 \\
G_{p,q}(z) &= G_0^{p,q} + G_1^{p,q} z + \cdots + G_{q-1}^{p,q} z^{q-1} + G_q^{p,q} z^q, & G_0^{p,q} &= I \\
F_{p,q}(z) &= F_p^{p,q} + F_{p-1}^{p,q} z + \cdots + F_0^{p,q} z^p \\
E_{p,q}(z) &= E_q^{p,q} + E_{q-1}^{p,q} z + \cdots + E_1^{p,q} z^{q-1} + E_0^{p,q} z^q, & E_0^{p,q} &= I
\end{aligned} \tag{3.43}$$

It follows from (3.39) and (3.40) that

$$
\begin{aligned}
A_{p+1,q}(z) &= A_{p,q}(z) + K_2^{p,q} \cdot zB_{p,q}(z) - K_3^{p,q} H_{p,q}(z), \\
B_{p+1,q}(z) &= zB_{p,q}(z) + L_1^{p,q} A_{p,q}(z) - L_3^{p,q} H_{p,q}(z) - L_4^{p,q} F_{p,q}(z), \\
H_{p+1,q}(z) &= H_{p,q}(z) - M_2^{p,q} \cdot zB_{p,q}(z), \\
F_{p+1,q}(z) &= F_{p,q}(z) - N_2^{p,q} B_{p+1,q}(z), \\
C_{p+1,q}(z) &= C_{p,q}(z) + K_2^{p,q} \cdot zD_{p,q}(z) - K_3^{p,q} G_{p,q}(z), \\
D_{p+1,q}(z) &= zD_{p,q}(z) + L_1^{p,q} C_{p,q}(z) - L_3^{p,q} G_{p,q}(z) - L_4^{p,q} E_{p,q}(z), \\
G_{p+1,q}(z) &= G_{p,q}(z) - M_2^{p,q} \cdot zD_{p,q}(z), \\
E_{p+1,q}(z) &= E_{p,q}(z) - N_2^{p,q} D_{p+1,q}(z).
\end{aligned}
\tag{3.44}
$$

As a result, we have a lattice structure, as depicted in Figure 3.1, for computing the matrix polynomials $A_{p,q}(z)$, $B_{p,q}(z)$, $F_{p,q}(z)$, $H_{p,q}(z)$ and $C_{p,q}(z)$, $D_{p,q}(z)$, $E_{p,q}(z)$, $G_{p,q}(z)$. Figure 3.1 illustrates how the ORA in p works, where the argument z of the matrix polynomials and the arguments (p,q) of the partial correlation coefficients are omitted. Each lattice section has the same structure, but different partial correlation coefficients. The figure shows that the parameters of LMMSE estimators of order $(p+1,q)$ are a weighted summation of at most four terms. Each of them is the parameters of an LMMSE estimator of order (p,q). The weighting factors are the partial correlation coefficients. From the implementation point of view, we only need to add one more lattice section to obtain the parameters of a new LMMSE output predictor of order $(p+1,q)$. Computationally, we just have to add a small additional computation to obtain a new predictor instead of repeating the previous work for computing the predictor of order (p,q).

Computational complexity

Note that multiplication of an $m \times n$ matrix and $n \times l$ matrix neeeds mnl multiplication operations and $ml(n-1)$ addition operations. Notice also that the computational complexity of n linear equations with n unknowns is $\frac{1}{6}n^3 + \mathbf{O}(n^2)$ if the Cholesky factorization method can be applied [43]. Thus, following

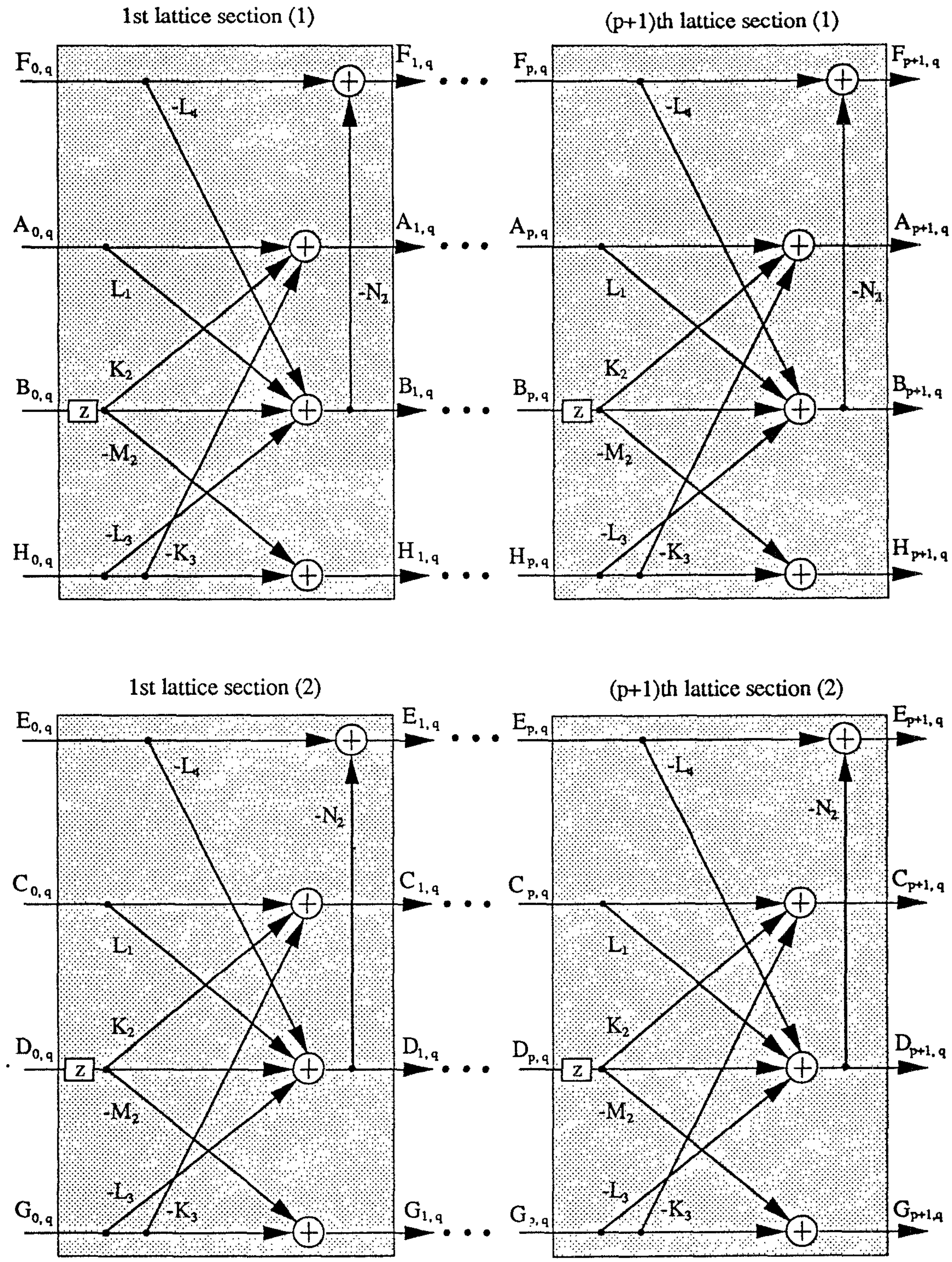


Figure 3.1: The lattice structure of the ORA in p

Task	Multi./Div.	Addition
$\{A_i^{p_0,q}\}_{i=1}^{p_0}, \{-C_j^{p_0,q}\}_{j=0}^{q}$	$\frac{m}{6}(p_0 m + (q+1)l)^3$	$\frac{m}{6}(p_0 m + (q+1)l)^3$
$\{B_i^{p_0,q}\}_{i=1}^{p_0}, \{-D_j^{p_0,q}\}_{j=1}^{q},$	$\frac{m}{6}(p_0 m + ql)^3$	$\frac{m}{6}(p_0 m + ql)^3$
$\{-H_i^{p_0,q}\}_{i=1}^{p_0}, \{G_j^{p_0,q}\}_{j=1}^{q},$	$\frac{l}{6}(p_0 m + ql)^3$	$\frac{l}{6}(p_0 m + ql)^3$
$\{-F_i^{p_0,q}\}_{i=0}^{p_0}, \{E_j^{p_0,q}\}_{j=1}^{q}$	$\frac{l}{6}((p_0+1)m + ql)^3$	$\frac{l}{6}((p_0+1)m + ql)^3$
$R^f(p_0, q)$	$p_0 m^3 + (q+1)m^2 l$	$p_0 m^3 + m^2 + (q+1)m^2 l$
$R^b(p_0, q)$	$p_0 m^3 + qm^2 l$	$p_0 m^3 + m^2 + qm^2 l$
$V^f(p_0, q)$	$p_0 l^2 m + ql^3$	$p_0 l^2 m + ql^3 + l^2$
$V^b(p_0, q)$	$(p_0+1)l^2 m + ql^3$	$ql^3 + (p_0+1)l^2 m$
$\alpha(p_0, q)$	$p_0 m^3 + (q+1)m^2 l$	$p_0 m^3 + m^2 + (q+1)m^2 l$
$\beta(p_0, q)$	$p_0 m^3 + qm^2 l$	$p_0 m^3 + m^2 + qm^2 l$
$\delta(p_0, q)$	$p_0 m^2 l + qml^2$	$ml + p_0 m^2 l + qml^2$
$\eta(p_0, q)$	$p_0 lm^2 + ql^2 m$	$p_0 lm^2 + lm + ql^2 m$
$\lambda(p_0, q)$	$(p_0+1)lm^2 + ql^2 m$	$(p_0+1)lm^2 + lm + ql^2 m$

Table 3.1: The computational load of the pre-recursion part

the description of the ORA in p, we can figure out the computational load for the pre-recursion part and each order iteration of the ORA in p, which are listed in Table 3.1 and Table 3.2.

Based on Table 3.1 and Table 3.2, we can eventually obtain the computational complexity of the ORA in p by following the description of the algorithm.

Theorem 3.2 Suppose that we use the ORA in p to solve an LMMSE prediction problem of order (p, q) or a set of the problems of order (p, q) with $p_0 \leq p \leq p^*$. Then, the computational complexity is

multiplication/division:

$$\frac{1}{2}(4m^3 + 7m^2 l)(p^{*^2} - p_0^2) + (3m^2 l + 7ml^2)(p^* - p_0)q + \frac{1}{3}(m+l)(p_0 m + ql)^3 + \mathbf{O}(p^* \vee (p_0 m + ql)^2)$$

addition:

$$(4m^3 + 8m^2 l)(p^{*^2} - p_0^2) + (5m^2 l + 7ml^2)(p^* - p_0)q + \frac{1}{3}(m+l)(p_0 m + ql)^3 + \mathbf{O}(p^* \vee (p_0 m + ql)^2)$$

∎

Task	Multi./Div.	Addition
$[V^f(p,q)]^{-1}$	l^3	l^3
$[R^f(p,q)]^{-1}$	m^3	m^3
$[R^b(p,q)]^{-1}$	m^3	m^3
$-L_3(p,q)$	ml^2	$ml(l-1)$
$K_2(p,q)$	m^2l+2m^3	$m^2l+3m^3-2m^2$
$K_3(p,q)$	m^2l	$m^2(l-1)$
$L_1(p,q)$	m^2l+m^3	$m^2(l-1)+m^3$
$L_4(p,q)$	m^2l+ml^2	m^2l+ml^2
$M_2(p,q)$	lm^2	$lm(m-1)$
$N_2(p,q)$	lm^2	$lm(m-1)$
$R^f(p+1,q)$	m^3+m^2l	m^3+m^2l
$R^b(p+1,q)$	m^3+2m^2l	m^3+2m^2l
$V^f(p+1,q)$	l^2m	l^2m
$\eta(p+1,q)$	lm^2	lm^2
$V^b(p+1,q)$	l^2m+l^3	$l^2(m-1)+l^3$
$\{A_i^{p+1,q}\}_{i=1}^{p+1}$	$(p+1)m^3+pm^2l$	pm^3+pm^2l
$\{-C_j^{p+1,q}\}_{j=0}^{q}$	$qm^2l+(q+1)ml^2$	$qm^2l+(q+1)ml^2+ml$
$\{B_i^{p+1,q}\}_{i=1}^{p+1}$	$(p+1)m^3+(2p+1)m^2l$	$(p+1)m^3+2m^2+(2p+1)m^2l$
$\{-D_j^{p+1,q}\}_{j=1}^{q}$	$qm^2l+2qml^2$	$qml(m+2l)$
$\{-H_i^{p+1,q}\}_{i=1}^{p+1}$	$(p+1)lm^2$	$2(p+1)lm(m-1)$
$\{G_j^{p+1,q}\}_{j=1}^{q}$	ql^2m	ql^2m
$\{-F_i^{p+1,q}\}_{i=0}^{p+1}$	$(p+2)lm^2$	$(p+2)lm^2$
$\{E_j^{p+1,q}\}_{j=1}^{q}$	ql^2m	ql^2m
$\alpha(p+1,q)$	$(p+2)m^3+(q+1)m^2l$	$(p+2)m^3+(q+1)m^2l-2m^2$
$\beta(p+1,q)$	$(p+2)m^3+qm^2l$	$(p+2)m^3+qm^2l-2m^2$
$\delta(p+1,q)$	$(p+2)m^2l+qml^2$	$(p+2)m^2l+qml^2-2ml$
$\lambda(p+1,q)$	$(p+2)lm^2+(q+1)l^2m$	$(p+2)lm^2+(q+1)l^2m-2lm$

Table 3.2: The computational load of each iteration in order-recursion in p

It can easily be seen from Table 3.2 that the ${p^*}^2$ term in the computational complexity of the ORA in p is contributed only by the parameter recursion described in (3.39) and the calculation of the intermediate variables $\alpha(p,q), \beta(p,q), \delta(p,q)$, and $\lambda(p,q)$ described in (3.41). In fact, for any order (p,q), the *all* multiplication computation involved in parameter recursion and the calculation of the above intermediate variables can be divided into $p+q$ or so individual matrix-multiplication operations. *All* of those operations can be processed in parallel and each of those operations takes only less than $(l \vee m)^3$ multiplication operations. However, the addition operations involved can not be parallelized as much as the multiplication operations are because some inner products occur in computing $\alpha(p,q), \beta(p,q), \delta(p,q)$, and $\lambda(p,q)$. As a result, we have the following conclusion.

Corollary 3.2 The procedure for each order iteration in the ORA in p is highly parallelizable with respect to multiplication. For solving a set of LMMSE prediction problems of order (p,q) with $p_0 \leq p \leq p^*$, the computational complexity of unparallelizable processes within the order-recursion in p is

multiplication/division: $(l \vee m)^3(p^* - p_0) + \mathbf{O}(1)$

addition: $(p_1 - p_0)\log_2[((p^*+1) + (q+1))(l \vee m)] + \mathbf{O}(p^* - p_0)$ ∎

Remark 3.7 As shown in Table 3.1, the cost of computing the initial condition described in (3.42) is of $\mathbf{O}((p_0 m + ql)^3)$ of multiplications and additions. This fact implies that, from the computational point of view, the initial order (p_0, q_0) should be chosen as low as possible. ∎

3.3.2 A Simplified Version of the ORA in p

If the underlying system is a causal system[5] with white driving noise $\{u_n\}$, one can see from (3.17) and (3.19) that the parameters of the input predictor in

[5] This implies $R_{yu}(-k) = 0$ and $R_{uy}(k) = 0$ whenever $k > 0$.

(3.8) are equal to zero and $V^f(p,q) = I$, $\eta(p,q) = R_{uy}(0)$, and $\zeta(p,q) = 0$. As a result, the number of auxiliary predictors within the ORA in p can be reduced from three to two and the resulting algorithm has a simpler form described as follows:

Algorithm ORA-in-p $(p_0, p^*, q, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$

Parameter recursion:

$$
\begin{aligned}
A_i^{p+1,q} &= A_i^{p,q} + K_2^{p,q} B_{p+1-i}^{p,q}, & i &= 1,\cdots,p+1,\\
C_j^{p+1,q} &= C_j^{p,q} - K_2^{p,q} D_{q+1-j}^{p,q}, & j &= 0,\cdots,q,\\
B_i^{p+1,q} &= B_i^{p,q} + L_1^{p,q} A_{p+1-i}^{p,q} - L_4^{p,q} F_{i-1}^{p,q}, & i &= 1,\cdots,p+1,\\
D_j^{p+1,q} &= D_j^{p,q} + L_1^{p,q} C_{q-j}^{p,q} - L_4^{p,q} E_j^{p,q}, & j &= 1,\cdots,q,\\
F_i^{p+1,q} &= F_{i-1}^{p,q} - N_2^{p,q} B_i^{p+1,q}, & i &= 0,\cdots,p+1,\\
E_j^{p+1,q} &= E_j^{p,q} - N_2^{p,q} D_j^{p+1,q}, & j &= 1,\cdots,q,
\end{aligned}
\tag{3.45}
$$

where the variable p varies from p_0 up to $p^* - 1$.

Calculator of partial correlation coefficients: The partial correlation coefficients used in the parameter recursion are calculated in the following way:

$$
\begin{aligned}
K_2^{p,q} &= -\alpha(p,q)\{R^b(p,q)\}^{-1}, & K_2^{p,q} &\in \mathcal{R}^{m\times m},\\
L_1^{p,q} &= -\beta(p,q)[R^f(p,q)]^{-1}, & L_1^{p,q} &\in \mathcal{R}^{m\times m},\\
L_4^{p,q} &= D_1^{p,q} + L_1^{p,q} C_q^{p,q}, & L_4^{p,q} &\in \mathcal{R}^{m\times l},\\
N_2^{p,q} &= -\lambda(p,q)[R^b(p+1,q)]^{-1}, & N_2^{p,q} &\in \mathcal{R}^{l\times m},
\end{aligned}
\tag{3.46}
$$

where the variable p varies from p_0 up to $p^* - 1$.

Calculator of intermediate variables:

$$
\begin{aligned}
R^f(p+1,q) &= R^f(p,q) + K_2^{p,q}\beta(p,q), & p &= p_0,\cdots,p^*-1,\\
R^b(p+1,q) &= R^b(p,q) + L_1^{p,q}\alpha(p,q) + L_4^{p,q}\lambda(p,q), & p &= p_0,\cdots,p^*-1,\\
V^b(p+1,q) &= V^b(p,q) + N_2^{p,q} L_4^{p,q} V^b(p,q), & p &= p_0,\cdots,p^*-1,\\
\alpha(p,q) &= \sum_{i=0}^{p} A_i^{p,q} R_{yy}(p+1-i) - \sum_{i=0}^{q} C_i^{p,q} R_{uy}(p+1-i), & p &= p_0,\cdots,p^*-1,\\
\beta(p,q) &= \sum_{i=0}^{p} B_{p-i}^{p,q} R_{yy}(-i-1) - \sum_{i=0}^{q-1} D_{q-i}^{p,q} R_{uy}(-i-1), & p &= p_0,\cdots,p^*-1,\\
\lambda(p,q) &= -\sum_{i=0}^{p} F_{p-i}^{p,q} R_{yy}(p+1-i), & p &= p_0,\cdots,p^*-1.
\end{aligned}
\tag{3.47}
$$

Furthermore, the ORA in p direction is reduced to the Levinson-Whittle-Wiggins-Robinson (LWWR) algorithm [21][138] when a pure AR model is considered, i.e., $q = -1$. The computational complexity of the resulting algorithm is:

multiplication: $\frac{1}{2}(4m^3 + 3m^2 l)(p_1^2 - p_0^2) + \frac{1}{3}m(p_0 m)^3 + \mathbf{O}(p_1 \vee (p_0 m)^2)$

addition: $\frac{1}{2}(4m^3 + 3m^2 l)(p_1^2 - p_0^2) + \frac{1}{3}m(p_0 m)^3 + \mathbf{O}(p_1 \vee (p_0 m)^2)$

3.3.3 The Lattice Algorithm of LMMSE MAX Predictors

The estimation errors defined in (3.10) can be expressed as follows:

$$\begin{aligned}
\mathbf{e}^{f,y}_{n,p,q} &= A_{p,q}(z)\mathbf{y}_n - C_{p,q}(z)\mathbf{u}_n, \\
\mathbf{e}^{b,y}_{n,p,q} &= B_{p,q}(z)\mathbf{y}_n - D_{p,q}(z)\mathbf{u}_n, \\
\mathbf{e}^{f,u}_{n,p,q} &= -H_{p,q}(z)\mathbf{y}_n + G_{p,q}(z)\mathbf{u}_n, \\
\mathbf{e}^{b,u}_{n,p,q} &= -F_{p,q}(z)\mathbf{y}_n + E_{p,q}(z)\mathbf{u}_n.
\end{aligned} \tag{3.48}$$

Substituting (3.44) into (3.48) yields the following lattice algorithm of LMMSE ARX predictors.

$$\begin{aligned}
\mathbf{e}^{f,y}_{n,p+1,q} &= \mathbf{e}^{f,y}_{n,p,q} + K_2^{p,q}\mathbf{e}^{b,y}_{n-1,p,q} + K_3^{p,q}\mathbf{e}^{f,u}_{n,p,q}, \\
\mathbf{e}^{b,y}_{n,p+1,q} &= \mathbf{e}^{b,y}_{n,p,q} + L_1^{p,q}\mathbf{e}^{f,y}_{n,p,q} + L_3^{p,q}\mathbf{e}^{f,u}_{n,p,q} + L_4^{p,q}\mathbf{e}^{b,u}_{n,p,q}, \\
\mathbf{e}^{f,u}_{n,p+1,q} &= \mathbf{e}^{f,u}_{n,p,q} + M_2^{p,q}\mathbf{e}^{b,y}_{n-1,p,q}, \\
\mathbf{e}^{b,u}_{n,p+1,q} &= \mathbf{e}^{b,u}_{n,p,q} + N_2^{p,q}\mathbf{e}^{b,y}_{n,p+1,q},
\end{aligned} \tag{3.49}$$

where the involved partial correlation coefficients are calculated according to Eqs.(3.40) and (3.41).

3.4 The Order-Recursive Algorithm in q

This section is a section parallel to Section 3.3. The main difference between this section and Section 3.3 is that it presents an fast algorithm which determines the solution to the Yule-Walker equation described in (3.15) recursively in the number of *feedforward parameters*. Since we denote by q the number of *feedforward parameters* in Eq. (3.15), we call the algorithm the order-recursive algorithm in q (ORA in q).

3.4.1 The General Version of the ORA in q for LMMSE Parameter Estimation

Like the ORA in p, the ORA in q consists of three parts: the parameter recursion, calculator of the partial correlation coefficients, and calculator of the intermediate variables. The ORA in q can be expressed as follows:

Algorithm ORA-in-q $(q_0, q^*, p, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$

Parameter recursion:

$$
\begin{aligned}
A_i^{p,q+1} &= A_i^{p,q} - P_3^{p,q} H_i^{p,q} - P_4^{p,q} F_{p-i}^{p-1,q}, && i = 1, \cdots, p, \\
C_j^{p,q+1} &= C_j^{p,q} - P_3^{p,q} G_j^{p,q} - P_4^{p,q} E_{q+1-j}^{p-1,q}, && j = 0, \cdots, q+1, \\
H_i^{p,q+1} &= H_i^{p,q} + S_4^{p,q} F_{p-i}^{p-1,q}, && i = 1, \cdots, p, \\
G_j^{p,q+1} &= G_j^{p,q} + S_4^{p,q} E_{q+1-j}^{p-1,q}, && j = 1, \cdots, q+1, \\
F_i^{p-1,q+1} &= F_{i+1}^{p-1,q} - T_1^{p,q} A_{p-1-i}^{p,q} - T_2^{p,q} B_{i+1}^{p,q+1} + T_3^{p,q} H_{p-1-i}^{p,q}, && i = 0, \cdots, p-1, \\
E_j^{p-1,q+1} &= E_j^{p-1,q} - T_1^{p,q} C_{q+1-j}^{p,q} - T_2^{p,q} D_j^{p,q+1} + T_3^{p,q} G_{q+1-j}^{p,q}, && j = 1, \cdots, q+1, \\
B_i^{p,q+2} &= B_i^{p,q+1} - Q_4^{p,q} F_{i-1}^{p-1,q+1}, && i = 1, \cdots, p, \\
D_j^{p,q+2} &= D_{j-1}^{p,q+1} - Q_4^{p,q} E_{j-1}^{p-1,q+1}, && j = 1, \cdots, q+2,
\end{aligned}
\tag{3.50}
$$

where the variable q varies from q_0 up to $q^* - 1$ if it is not involved in $B_j^{p,q+2}$ or $D_j^{p,q+2}$. Otherwise, it varies from q_0 up to $q^* - 2$.

Calculator of partial correlation coefficients:

$$
\begin{aligned}
P_4^{p,q} &= -\gamma(p,q)[V^b(p-1,q) - \kappa^T(p,q)[V^f(p,q)]^{-1}\kappa(p,q)]^{-1}, && P_4^{p,q} \in \mathcal{R}^{m\times l},\\
P_3^{p,q} &= -P_4^{p,q}\kappa^T(p,q)[V^f(p,q)]^{-1}, && P_3^{p,q} \in \mathcal{R}^{m\times l},\\
S_4^{p,q} &= -\kappa(p,q)[V^b(p-1,q)]^{-1}, && S_4^{p,q} \in \mathcal{R}^{l\times l},\\
T_3^{p,q} &= -\kappa^T(p,q)[V^f(p,q)]^{-1}, && T_3^{p,q} \in \mathcal{R}^{l\times l},\\
T_1^{p,q} &= -[\mu(p-1,q) + T_3^{p,q}\eta(p,q)][R^f(p,q)]^{-1}, && T_1^{p,q} \in \mathcal{R}^{l\times m},\\
T_2^{p,q} &= -T_1^{p,q}A_p^{p,q} + T_3^{p,q}H_p^{p,q} + F_0^{p-1,q}, && T_2^{p,q} \in \mathcal{R}^{l\times m},\\
Q_4^{p,q} &= -\nu(p,q+1)[V^b(p-1,q+1)]^{-1}, && Q_4^{p,q} \in \mathcal{R}^{m\times l},
\end{aligned}
\tag{3.51}
$$

where the variable q varies from q_0 up to q^*-1.

Calculator of intermediate variables:

$$
\begin{aligned}
&R^f(p,q+1) = R^f(p,q) + P_3^{p,q}\eta(p,q) + P_4^{p,q}\mu(p-1,q), && q=q_0,\cdots,q^*-1,\\
&R^b(p,q+2) = R^b(p,q+1) + Q_4^{p,q}T_2^{p,q}R^b(p,q+1), && q=q_0,\cdots,q^*-2,\\
&V^f(p,q+1) = V^f(p,q) + S_4^{p,q}\kappa(p,q), && q=q_0,\cdots,q^*-1,\\
&V^b(p-1,q+1) = V^b(p-1,q) + T_1^{p,q}\gamma(p,q)\\
&\qquad\qquad + T_2^{p,q}\nu(p,q+1) + T_3^{p,q}\kappa(p,q), && q=q_0,\cdots,q^*-1,\\
&\eta(p,q+1) = \eta(p,q) + S_4^{p,q}\mu(p-1,q), && q=q_0,\cdots,q^*-2,\\
&\kappa(p,q) = -\sum_{i=1}^{p} H_i^{p,q}R_{yu}(q-i) + \sum_{j=0}^{q} G_j^{p,q}R_{uu}(q+1-j), && q=q_0,\cdots,q^*-1,\\
&\mu(p-1,q) = -\sum_{i=0}^{p-1} F_{p-1-i}^{p-1,q}R_{yy}(-i-1)\\
&\qquad\qquad + \sum_{j=0}^{q} E_{q-i}^{p-1,q}R_{uy}(-j-1), && q=q_0,\cdots,q^*-1,\\
&\gamma(p,q) = \sum_{i=0}^{p} A_i^{p,q}R_{yu}(q+1-i) - \sum_{j=0}^{q} C_j^{p,q}R_{uu}(q+1-j), && q=q_0,\cdots,q^*-1,\\
&\nu(p,q) = \sum_{i=0}^{p} B_{p-i}^{p,q}R_{yu}(q-1-i) + \sum_{i=0}^{q-1} D_{q-i}^{p,q}R_{uu}(q-1-i), && q=q_0,\cdots,q^*-1.
\end{aligned}
\tag{3.52}
$$

Initial condition at order $(p_0,q_0)=(p,q_0)$:

$$
\begin{pmatrix}
A_1^{p,q_0} & \cdots & A_p^{p,q_0} & -C_0^{p,q_0} & \cdots & -C_{q_0}^{p,q_0} & R^f(p,q_0)\\
B_p^{p,q_0+1} & \cdots & B_1^{p,q_0+1} & -D_{q_0+1}^{p,q_0+1} & \cdots & -D_1^{p,q_0+1} & R^b(p,q_0+1)\\
-H_1^{p,q_0} & \cdots & -H_p^{p,q_0} & G_1^{p,q_0} & \cdots & G_{q_0}^{p,q_0} & V^f(p,q_0)\\
-F_{p-1}^{p-1,q_0} & \cdots & -F_0^{p-1,q_0} & E_{q_0}^{p-1,q_0} & \cdots & E_1^{p-1,q_0} & V^b(p,q_0)
\end{pmatrix}
\tag{3.53}
$$

It is clear that the operating procedures of the ORA in p and the ORA in q are very similar. Therefore, Remarks 3.3 – 3.7 can be used to interpret the ORA in q with mild revisions. For simplicity, we do not repeat these arguments. Instead, we request readers to refer to Remarks 3.3 – 3.7.

Lattice Version

Substituting (3.50) and (3.51) into (3.43) yields the following lattice version of the parameter recursion:

$$\begin{aligned}
A_{p,q+1}(z) &= A_{p,q}(z) - P_3^{p,q} H_{p,q}(z) - P_4^{p,q} \cdot zF_{p-1,q}(z), \\
C_{p,q+1}(z) &= C_{p,q}(z) - P_3^{p,q} G_{p,q}(z) - P_4^{p,q} \cdot zE_{p-1,q}(z), \\
H_{p,q+1}(z) &= H_{p,q}(z) - S_4^{p,q} zF_{p-1,q}(z), \\
G_{p,q+1}(z) &= G_{p,q}(z) + S_4^{p,q} zE_{p-1,q}(z), \\
F_{p-1,q+1}(z) &= zF_{p-1,q}(z) - T_1^{p,q} A_{p,q}(z) - T_2^{p,q} B_{p,q+1}(z) + T_3^{p,q} H_{p,q}(z), \\
E_{p-1,q+1}(z) &= zE_{p-1,q}(z) - T_1^{p,q} C_{p,q}(z) - T_2^{p,q} D_{p,q+1}(z) + T_3^{p,q} G_{p,q}(z), \\
B_{p,q+2}(z) &= B_{p,q+1}(z) - Q_4^{p,q} F_{p-1,q+1}(z), \\
D_{p,q+2}(z) &= D_{p,q+1}(z) - Q_4^{p,q} E_{p-1,q+1}(z),
\end{aligned} \tag{3.54}$$

where $A_{p,q}(z)$, $C_{p,q}(z)$, $B_{p,q}(z)$, $D_{p,q}(z)$, $H_{p,q}(z)$, $G_{p,q}(z)$, and $F_{p-1,q}(z)$, $E_{p-1,q}(z)$ are defined in (3.43) . The lattice structure of the ORA in q is expressed in Figure 3.2. There, the argument z of matrix polynomials and the arguments (p, q) of the partial correlation coefficients are omitted.

Computational Complexity

Following the description of the ORA in q, we can determine the computational load for pre-recursion part and each order iteration of the ORA in q, which are list in Table 3.3 and Table 3.4.

Based on Table 3.3 and Table 3.4, we eventually obtain the computational complexity of the ORA in q by following the description of the algorithm.

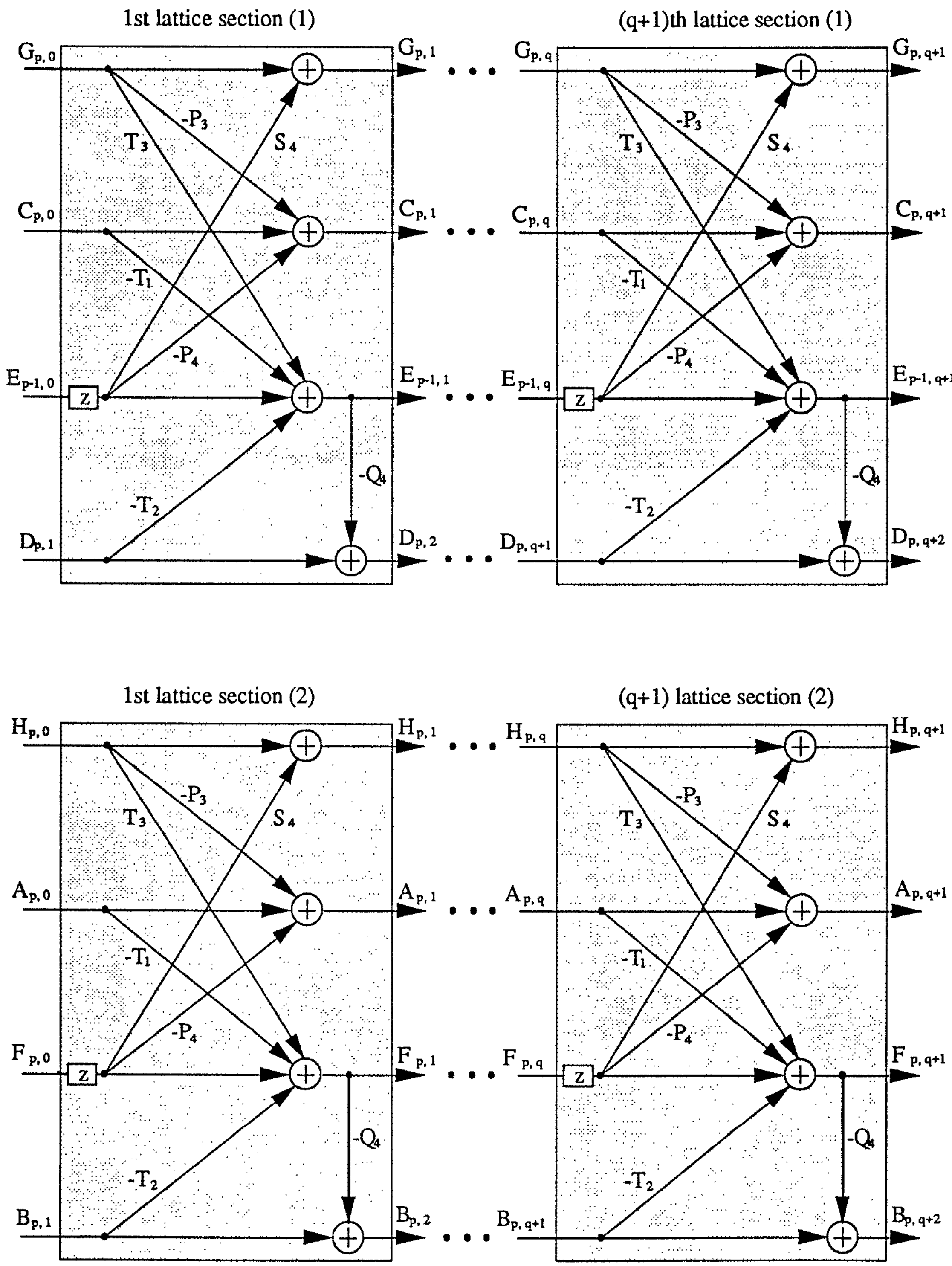


Figure 3.2: The lattice structure of the ORA in q

Task	Multi./Div.	Addition
$\{A_i^{p,q_0}\}_{i=1}^p, \{-C_j^{p,q_0}\}_{j=0}^{q_0}$	$\frac{m}{6}(pm+(q_0+1)l)^3$	$\frac{m}{6}(pm+(q_0+1)l)^3$
$\{-H_i^{p,q_0}\}_{i=1}^p, \{G_j^{p,q_0}\}_{j=1}^{q_0},$	$\frac{l}{6}(pm+q_0l)^3$	$\frac{l}{6}(pm+q_0l)^3$
$\{-F_i^{p-1,q_0}\}_{i=0}^{p-1}, \{E_j^{p-1,q_0}\}_{j=1}^{q_0}$	$\frac{l}{6}(pm+q_0l)^3$	$\frac{l}{6}(pm+q_0l)^3$
$\{-B_i^{p,q_0+1}\}_{i=1}^p, \{-D_j^{p,q_0+1}\}_{j=0}^{q_0},$	$\frac{l}{6}(pm+q_0l)^3$	$\frac{l}{6}(pm+q_0l)^3$
$R^f(p,q_0)$	$pm^3+(q_0+1)m^2l$	$pm^3+m^2+(q_0+1)m^2l$
$V^f(p,q_0)$	$pl^2m+q_0l^3$	$pl^2m+q_0l^3+l^2$
$V^b(p-1,q_0)$	$pl^2m+q_0l^3$	$q_0l^3+pl^2m-l^2$
$\overline{R}^b(p,q_0)$	$pm^3+(q_0+1)m^2l$	$pm^3+m^2+(q_0+1)m^2l$
$\gamma(p,q_0)$	$pm^2l+(q_0+1)ml^2$	$pm^2l+(q_0+1)ml^2-ml$
$\overline{\tau}(p,q_0)$	$pm^2l+(q_0+1)ml^2$	$pm^2l+(q_0+1)ml^2-ml$
$\eta(p,q_0)$	$plm^2+q_0ml^2$	$plm^2+q_0l^2m-lm$
$\kappa(p,q_0)$	$pl^2m+q_0l^3$	$pl^2m+q_0l^3-l^2$
$\mu(p-1,q_0)$	$pl^2m+q_0l^3$	$pml^2+q_0l^3-l^2$

Table 3.3: The computational load of the pre-recursion part in the ORA in q

Theorem 3.3 Suppose that we use the ORA in q to solve an LMMSE prediction problem of order (p,q^*) or a set of the problems of order (p,q) with $q_0 \le q \le q^*$. Then, the computational complexity is

multiplication/division:

$$\frac{1}{2}(7l^2m+4l^3)(q^{*^2}-q_0^2)+(7m^2l+4ml^2)(q^*-q_0)p$$
$$+\frac{1}{3}(m+l)(pm+(q_0+1)l)^3+\mathbf{O}(q^*\vee(pm+(q_0+1)l)^2)$$

addition:

$$\frac{1}{2}(7l^2m+4l^3)(q^{*^2}-q_0^2)+(7m^2l+4ml^2)(q^*-q_0)p$$
$$+\frac{1}{3}(m+l)(pm+(q_0+1)l)^3+\mathbf{O}(q^*\vee(pm+(q_0+1)l)^2)$$

∎

It can easily be seen from Table 3.4 that the q^{*^2} term in the computational complexity of the ORA in q is contributed only by the parameter recursion described in (3.50) and calculation of intermediate variables $\gamma(p,q+1), \nu(p,q+1), \kappa(p,q+1)$ and $\mu(p-1,q+1)$, described in (3.52). Furthermore, for any order (p,q), the *all* multiplication computation involved in the calculation of

Task	Multi./Div.	Addition
$[V^f(p,q)]^{-1}$	l^3	l^3
$[V^b(p-1,q)]^{-1}$	l^3	l^3
$[R^f(p,q)]^{-1}$	m^3	m^3
$P_4(p,q)$	$2l^3+ml^2$	$2l^3+ml^2-ml-l^2$
$P_3(p,q)$	ml^2	$ml(l-1)$
$S_4(p,q)$	l^3	$l^2(l-1)$
$T_3(p,q)$	l^3	$l^2(l-1)$
$T_1(p,q)$	lm^2+l^2m	l^2m+lm^2-lm
$T_2(p,q)$	lm^2+l^2m	l^2m+lm^2
$Q_4(p,q)$	ml^2	$ml(l-1)$
$\{A_i^{p,q+1}\}_{i=1}^{p}$	$2pm^2l$	$2pm^2l$
$\{-C_j^{p,q+1}\}_{j=0}^{q+1}$	$2ql^2m$	$2ql^2m+2ml$
$\{-H_i^{p,q+1}\}_{i=1}^{p}$	pml^2	pml^2
$\{G_j^{p,q+1}\}_{j=1}^{q+1}$	ql^3	ql^3
$\{-F_i^{p-1,q+1}\}_{i=0}^{p-1}$	$p(2m^2l+l^2m)$	$p(2lm^2+l^2m)$
$\{E_j^{p,q+1}\}_{j=1}^{q+1}$	$q(2l^2m+l^3)+2l^2m$	$q(2l^2m+l^3)+2l^2m$
$\{B_i^{p,q+2}\}_{j=1}^{p}$	pm^2l	pm^2l
$\{-D_j^{p,q+2}\}_{j=0}^{q+1}$	$(q+1)ml^2$	$(q+1)ml^2$
$R^f(p,q+1)$	$2m^2l$	$2m^2l$
$V^f(p,q+1)$	l^3	l^3
$V^b(p-1,q+1)$	$2l^2m+l^3$	$2l^2m+l^3$
$R^b(p,q+2)$	m^2l+m^3	$m^3+m^2(l-1)$
$\eta(p,q+1)$	ml^2	ml^2
$\gamma(p,q+1)$	$pm^2l+(q+2)ml^2$	$pm^2l+(q+2)ml^2-ml$
$\nu(p,q+1)$	$pm^2l+(q+2)ml^2$	$pm^2l+(q+2)ml^2-ml$
$\kappa(p,q+1)$	$pl^2m+(q+1)l^3$	$pl^2m+(q+1)l^3-l^2$
$\mu(p-1,q+1)$	$pl^2m+(q+1)l^3$	$pml^2+(q+1)l^3-l^2$

Table 3.4: The computational load of each iteration in order-recursion in q

each of the above intermediate variables or the parameters of an LMMSE estimator can be divided into $p+q$ or so individual matrix-multiplication operations. *All* of those operations can be processed in parallel and each of those operations takes only less than $(l \vee m)^3$ multiplication operations. However, the addition operations involved can not be parallelized as much as the multiplication operations are because some inner products occur in computing $\gamma(p, q+1), \nu(p, q+1), \kappa(p, q+1)$ and $\mu(p-1, q+1)$.

Corollary 3.3 The procedure for each order iteration in the ORA in q is highly parallelizable with respect to multiplication. For solving a set of LMMSE prediction problems of order (p, q) with $q_0 \leq q \leq q^*$, the computational complexity of unparallelizable processes within the order recursion involved is

multiplication/division: $(l \vee m)^3(q^* - q_0) + \mathbf{O}(1)$

addition: $(q^* - q_0)\log_2[((p+1) + (q^*+2))(l \vee m)] + \mathbf{O}(q^* - q_0)$ ∎

3.4.2 The ORA in q for LMMSE Parameter Estimation of an All-Zero Model

All-zero models are a class of popular models having wide applications in signal processing. In this case, the number of feedback parameters is equal to zero and the model order is of the form of $(0, q)$. In this subsection, we present the ORA in q for the special class of all-zero models. Starting from the general version of the ORA in q, we can derive the following simpler description of the ORA in q.

Parameter recursion:

$$\begin{aligned}
C_j^{0,q+1} &= C_j^{0,q} - P_3^{0,q} G_j^{0,q} - P_4^{0,q} E_{q+1-j}^{-1,q}, && j = 0, \cdots, q+1, \\
G_j^{0,q+1} &= G_j^{0,q} + S_4^{0,q} E_{q+1-j}^{-1,q}, && j = 1, \cdots, q+1, \\
E_j^{-1,q+1} &= E_j^{-1,q} + T_3^{0,q} G_{q+1-j}^{0,q}, && j = 1, \cdots, q+1,
\end{aligned} \tag{3.55}$$

where the variable q varies from q_0 up to $q^* - 1$.

Calculator of partial correlation coefficients:

$$
\begin{aligned}
P_4^{0,q} &= -\gamma(0,q)[V^b(-1,q) - \kappa^T(0,q)[V^f(0,q)]^{-1}\kappa(0,q)]^{-1}, & P_4^{0,q} &\in \mathcal{R}^{m\times l},\\
P_3^{0,q} &= -P_4^{0,q}\kappa^T(0,q)[V^f(0,q)]^{-1}, & P_3^{0,q} &\in \mathcal{R}^{m\times l},\\
S_4^{0,q} &= -\kappa(0,q)[V^b(0-1,q)]^{-1}, & S_4^{0,q} &\in \mathcal{R}^{l\times l},\\
T_3^{0,q} &= -\kappa^T(0,q)[V^f(0,q)]^{-1},
\end{aligned}
\tag{3.56}
$$

where the variable q varies from q_0 up to $q^* - 1$.

Calculator of intermediate variables:

$$
\begin{aligned}
R^f(0,q+1) &= R^f(0,q) + P_3^{0,q}\eta(0,q) + P_4^{0,q}\mu(-1,q), & q &= q_0,\cdots,q^*-1,\\
V^f(0,q+1) &= V^f(0,q) + S_4^{0,q}\kappa(0,q), & q &= q_0,\cdots,q^*-1,\\
V^b(-1,q+1) &= V^b(-1,q) + T_3^{0,q}\kappa(0,q), & q &= q_0,\cdots,q^*-1,\\
\eta(0,q+1) &= \eta(0,q) + S_4^{0,q}\mu(-1,q), & q &= q_0,\cdots,q^*-2,\\
\kappa(0,q) &= \sum_{j=0}^{q} G_j^{p,q}R_{uu}(q+1-j), & q &= q_0,\cdots,q^*-1,\\
\mu(p-1,q) &= \sum_{j=0}^{q} E_{q-i}^{p-1,q}R_{uy}(-j-1), & q &= q_0,\cdots,q^*-1,\\
\gamma(p,q) &= R_{yu}(q+1) - \sum_{j=0}^{q} C_j^{p,q}R_{uu}(q+1-j), & q &= q_0,\cdots,q^*-1,
\end{aligned}
\tag{3.57}
$$

3.4.3 The Lattice Algorithm of LMMSE MAX Predictors

As shown in (3.48), the estimation errors defined in (3.10) can be expressed in terms of polynomials. Substituting (3.50) into (3.48) produces immediately the following lattice algorithm of LMMSE MAX predictors.

$$
\begin{aligned}
\mathbf{e}_{n,p,q+1}^{f,y} &= \mathbf{e}_{n,p,q}^{f,y} + P_3^{p,q}\mathbf{e}_{n,p,q}^{f,u} + P_4^{p,q}\mathbf{e}_{n-1,p-1,q}^{b,u},\\
\mathbf{e}_{n,p,q+1}^{f,u} &= \mathbf{e}_{n,p,q}^{f,u} + S_4^{p,q}\mathbf{e}_{n-1,p-1,q}^{b,u},\\
\mathbf{e}_{n,p-1,q+1}^{b,u} &= \mathbf{e}_{n-1,p-1,q}^{b,u} + T_1^{p,q}\mathbf{e}_{n,p,q}^{f,y}, + T_2^{p,q}\mathbf{e}_{n,p,q+1}^{b,y}, + T_3^{p,q}\mathbf{e}_{n,p,q}^{f,u},\\
\mathbf{e}_{n,p,q+2}^{b,y} &= \mathbf{e}_{n,p,q+1}^{b,y} + Q_4^{p,q}\mathbf{e}_{n,p-1,q+1}^{b,u},
\end{aligned}
\tag{3.58}
$$

where the involved partial correlation coefficients are calculated according to Eqs.(3.51) and (3.52).

3.5 The Whole Order-Recursive Algorithm

As we can see from Table 3.1 and Table 3.3, neither the ORA in p nor the ORA in q can efficiently solve problem (3.15) in general cases because both p and q could be large. However, the problem can be efficiently solved by using the ORA's in p and in q alternately. An incorporation of the order-recursion in p and the order-recursion in q results in the desired fast algorithm, which we call the order recursive algorithm (ORA). The form of the ORA depends on the path of order increment, which is defined below.

Definition 3.4 A set of ARX model orders $\mathbf{P} \subset \mathcal{O}_{0,-1}^{p^*,q^*}$ forms an order increment path if and only if

$$\mathbf{P} = \{(p_i, q_i) \in \mathcal{O}_{0,-1}^{p^*,q^*}, i = 0, \cdots, M | p_{i+1} \geq p_i, q_{i+1} \geq q_i, p_{i+1} - p_i + q_{i+1} - q_i = 1\}.$$

∎

In fact, an order increment path implies a directed trajectory in the order space which goes only in the up direction or in the right direction with step length of one. Obviously, for any order (p^*, q^*), there could be many different order increment paths from an initial order (p_0, q_0) to order (p^*, q^*). Along any order increment path starting from (p_0, q_0) and ending at (p^*, q^*), the ORA can efficiently solve all LMMSE prediction problems of order (p, q) lying on the path in a successive manner, where the ORA in p and the ORA in q are used alternately. When the order increment path goes in the up direction, the ORA in p is performed; otherwise, the ORA in q is executed. Hence, once the order increment path is chosen, the switching procedure between the ORA in p and the ORA in q is fixed. Before discussing the ORA further, one point should be made clear. How do you obtain the initial conditions of the ORA in p and the ORA in q ?

3.5.1 Calculation of Initial Conditions

The initial conditions involved in the ORA are described in Eqs. (3.42) and (3.53). Computing the initial conditions is trivial in computation when the initial order (p_0, q) or (p, q_0) is small. However, this is not the case when the ORA in p is switched to, or back from, the ORA in q. This is because the order increment path could change its direction even when the order (p, q) has been large. So, a natural question is: Can we compute an initial condition efficiently ? Alternatively, can we use results computed by the ORA in p to determine the initial condition of the ORA in q, or vice versa ? The answer is yes.

Determination of the Initial Condition of the ORA in P

Suppose that we have the solutions to Eq. (3.32) of order $(p, q), q = q_0, \cdots, q_1$, which are computed via the ORA in q. It follows from (3.50) and (3.51) that the solutions

$$(A_1^{p,q_1} \quad \cdots \quad A_p^{p,q_1} \quad -C_0^{p,q_1} \quad \cdots \quad -C_{q_1}^{p,q_1}) \tag{3.59}$$

$$(B_p^{p,q_1} \quad \cdots \quad B_1^{p,q_1} \quad -D_{q_1}^{p,q_1} \quad \cdots \quad -D_1^{p,q_1}) \tag{3.60}$$

$$(-H_1^{p,q_1} \quad \cdots \quad -H_p^{p,q_1} \quad G_1^{p,q_1} \quad \cdots \quad G_{q_1}^{p,q_1}) \tag{3.61}$$

$$(-F_{p-1}^{p-1,q_1} \quad \cdots \quad -F_0^{p-1,q_1} \quad E_{q_1}^{p-1,q_1} \quad \cdots \quad E_1^{p-1,q_1}) \tag{3.62}$$

and $R^b(p, q)$ are known. Thus, $\lambda(p-1, q_1)$ can be computed through (3.41). Substituting the matrices $\lambda(p-1, q_1)$ and $R^b(p, q)$ into (3.40) yields N_2^{p-1,q_1} and then produces, via (3.39),

$$(-F_p^{p,q_1} \quad \cdots \quad -F_0^{p,q_1} \quad E_{q_1}^{p,q_1} \quad \cdots \quad E_1^{p,q_1}) . \tag{3.63}$$

Consequently, the initial condition of the ORA in p at the initial order (p, q_1) is completely attained.

Determination of the Initial Condition of the ORA in Q

Suppose that we have the solutions to Eq. (3.30) of order $(p, q), p = p_0, \cdots, p_1$, which are computed via the ORA in p. It follows from (3.39) and (3.40) that the solutions

$$(A_1^{p_1,q} \quad \cdots \quad A_{p_1}^{p_1,q} \quad -C_0^{p_1,q} \quad \cdots \quad -C_q^{p_1,q}) \tag{3.64}$$

$$(B_{p_1}^{p_1,q} \quad \cdots \quad B_1^{p_1,q} \quad -D_q^{p_1,q} \quad \cdots \quad -D_1^{p_1,q}) \tag{3.65}$$

$$(-H_1^{p_1,q} \quad \cdots \quad -H_{p_1}^{p_1,q} \quad G_1^{p_1,q} \quad \cdots \quad G_q^{p_1,q}) \tag{3.66}$$

$$(-F_{p_1-1}^{p_1-1,q} \quad \cdots \quad -F_0^{p_1-1,q} \quad E_q^{p_1-1,q} \quad \cdots \quad E_1^{p_1-1,q}) \tag{3.67}$$

and $V^b(p_1 - 1, q)$ are known. Thus, $\nu(p_1, q)$ can be computed through (3.52) and then $Q_4^{p_1,q-1}$ can be calculated. As a result,

$$(B_{p_1}^{p_1,q+1} \quad \cdots \quad B_1^{p_1,q+1} \quad -D_{q+1}^{p_1,q+1} \quad \cdots \quad -D_1^{p_1,q+1}) \tag{3.68}$$

can be determined via (3.50). Consequently, the initial condition of the ORA in q at the initial order (p_1, q) is obtained.

3.5.2 The Generic Form of the ORA

An order increment path followed by the ORA decides the switching procedure between the ORA in p and the ORA in q. A different switching procedure leads to a different incorporation of the ORA in p and the ORA in q, and then a different version of the ORA. The generic form of the ORA can be illustrated in Figure 3.3. In the ORA, the ORA in p and ORA in q may share some information. For instance, as discussed before, the solutions generated by the ORA in p can be used as the initial condition for the ORA in q when the execution of the ORA in p stops and the order-recursion in q is to be started. The buffer in Figure 3.3 is used to store the information which may be used for some purposes including computing initial conditions. The switching procedure could affect the storage requirement and computational complexity/throughput of the whole ORA. In this thesis, we are not going to discuss this issue in detail. Instead, we present two versions of the ORA for two kinds of typical problems.

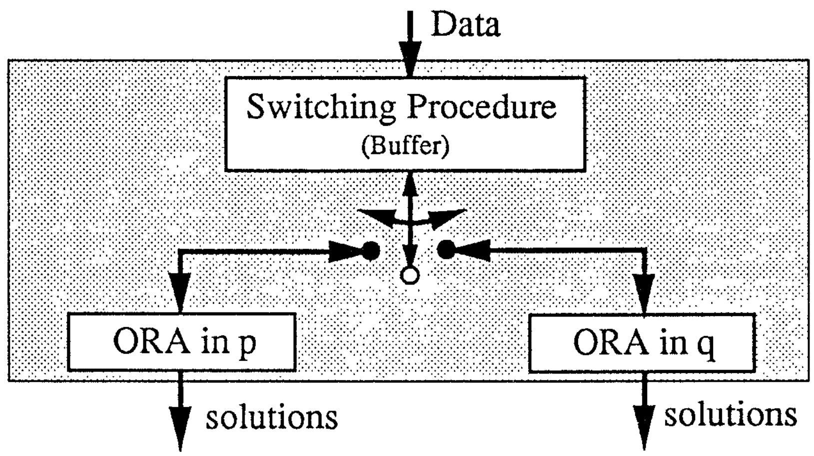


Figure 3.3: The generic form of the order-recursive algorithm

An LMMSE problem: Suppose that we consider an LMMSE problem of order (p^*, q^*). The corresponding order increment path could be chosen as

$$\mathbf{P}(p^*, q^*) = \{(p_i, q_i) | (p_i, q_i) = \begin{cases} (i-1, -1), & i = 1, \cdots, p^*+1, \\ (p^*, i-p^*-2) & i = p^*+2, \cdots, p^*+2+q^* \end{cases} \}$$

which is depicted in Figure 3.4. For such an order increment path, the ORA can be described as:

Procedure ORA-of-Type-I

Data $\{R_{yy}(i)\}_{i=0}^{p^*}$, $\{R_{uu}(j)\}_{j=0}^{q^*}$, $\{R_{yy}(i)\}_{i=-q^*}^{p^*}$

start

call order-recursion-in-p $(0, p^*, -1, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$[6]

call order-recursion-in-q $(0, q^*, p^*, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$[7]

stop

[6]See the ORA in p for details.
[7]See the ORA in q for details.

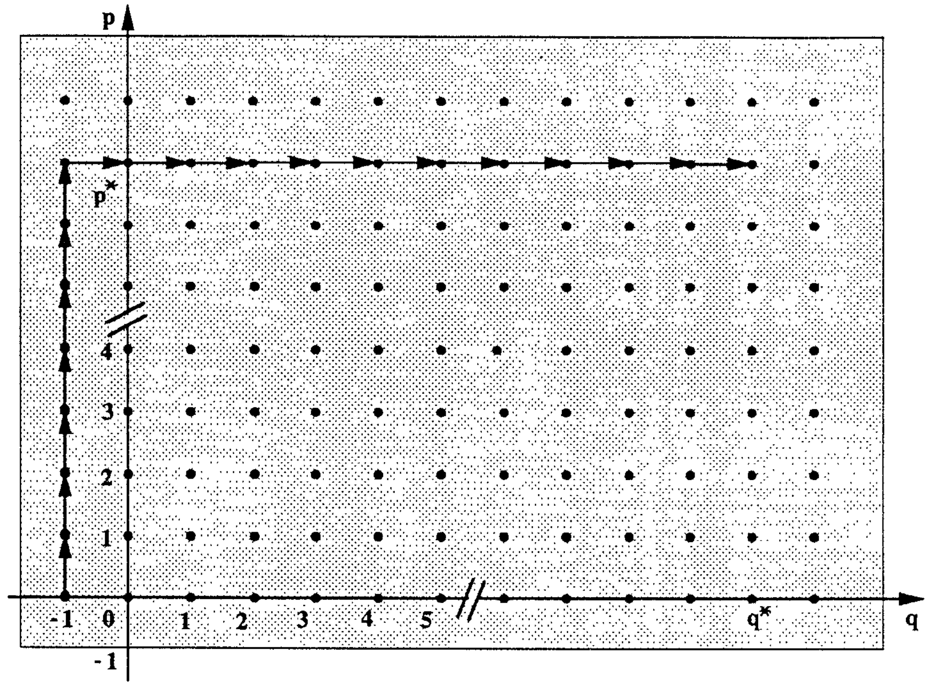


Figure 3.4: Illustration of the order increment path $\mathbf{P}(p^*, q^*)$

Computational Complexity

From Theorems 3.2 and 3.3, we have the following conclusion.

Theorem 3.4 *The ORA can efficiently determine LMMSE parameter estimates for an ARX model of order* (p^*, q^*). *The form of an ORA depends on the selection of an order increment path. Computationally, an ORA actually produces all LMMSE parameter estimates for all ARX models of order lying on the order increment path. The computational complexity of the ORA of type I is*

multiplication:

$$\tfrac{1}{2}(4m^3 + 3m^2 l)p^{*^2} + \tfrac{1}{2}(4l^3 + 7l^2 m)q^{*^2} + (7m^2 l + 4ml^2)q^* p^* + \mathbf{O}(p^* \vee q^*)$$

addition:

$$\tfrac{1}{2}(4m^3 + 3m^2 l)p^{*^2} + \tfrac{1}{2}(4l^3 + 7l^2 m)q^{*^2} + (7m^2 l + 4ml^2)q^* p^* + \mathbf{O}(p^* \vee q^*)$$

∎

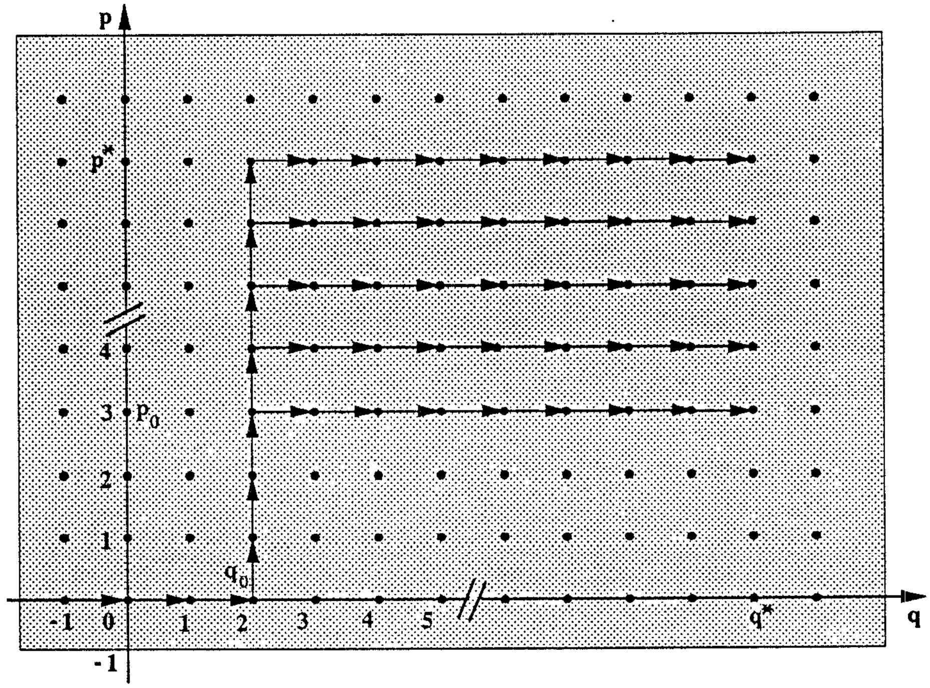


Figure 3.5: The pathes of order increment for a set of LMMSE problems

A family of LMMSE problems: Suppose that we consider a finite family of LMMSE problems of order (p,q) within a rectangle $\mathbf{R} \triangleq \{(p,q)|p = p_0, \cdots, p^*, q = q_0, \cdots, q^*\}$. In this case, we need a set of order increment paths, which may be described as:

$$\mathbf{P}(p^*,q^*) = \mathbf{P}_{1i}(-1,q_0;0) \bigcup \mathbf{P}_{2i}(0,p^*;q_0) \bigcup (\bigcup_{j=0}^{p^*-p_0} \mathbf{P}_j(q_0+1,q^*;p_0+j))$$

where

$$\mathbf{P}_{1i}(-1,q_0;0) = \{(p_i,q_i)|p_i = 0, q_i = i-1, i = 0,1,\cdots,q_0+1\}$$

$$\mathbf{P}_{2i}(0,p^*;q_0) = \{(p_i,q_i)|p_i = i, q_i = q_0, i = 0,1,\cdots,p^*\}$$

$$\mathbf{P}_j(q_0+1,q^*;p_0+j) = \{(p_i,q_i)|p_i = j, q_i = i+q_0+1, i = 0,\cdots,q^*-q_0-1\}.$$

The above set of order increment paths is also depicted in Figure 3.5.

In this case, the order recursive algorithm is expressed as:

Procedure ORA-of-Type-II

Data $\{R_{yy}(i)\}_{i=0}^{p^*}$, $\{R_{uu}(j)\}_{j=0}^{q^*}$, $\{R_{yy}(i)\}_{i=-q^*}^{p^*}$

start

call order-recursion-in-q $(0, q_0, 0, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$

call order-recursion-in-p $(0, p^*, q_0, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$

for $p := p_0$ up to p^*

begin

call order-recursion-in-q $(q_0 + 1, q^*, p, R_{yy}(\cdot), R_{yu}(\cdot), R_{uu}(\cdot))$

end

Hence, Theorem 3.2 and Theorem 3.3 lead to the following conclusion.

Theorem 3.5 The order-recursive algorithm can also be applied to solving LMMSE parameter estimation problems for all ARX models of order (p, q), where $p = p_0, p_0 + 1, \cdots, p^*$, and $q = q_0, q_0 + 1, \cdots, q^*$. The form of an ORA depends on the selection of order increment pathes. The computational complexity of the ORA of type II is

multiplication:

$$\frac{1}{2}(4l^3 + 7l^2 m)q^{*^2}(p^* - p_0) + (7m^2 l + 4ml^2)q^*(p^{*^2} - p_0^2)$$
$$+ \frac{1}{2}(4m^3 + 7m^2 l)p^{*^2} + (3m^2 l + 7ml^2)p^* q_0$$
$$+ \frac{1}{2}(7l^3 + 4lm^2)q_0^2 + \mathbf{O}(p^* \vee q^*)$$

addition:

$$\frac{1}{2}(4l^3 + 7l^2 m)q^{*^2}(p^* - p_0) + (7m^2 l + 4ml^2)q^*(p^{*^2} - p_0^2)$$
$$+ \frac{1}{2}(4m^3 + 7m^2 l)p^{*^2} + (3m^2 l + 7ml^2)p^* q_0$$
$$+ \frac{1}{2}(7l^3 + 4lm^2)q_0^2 + \mathbf{O}(p^* \vee q^*)$$

∎

From Corollaries 3.2 and 3.3, we can determine the computational complexity of the unparallelizable processes within the order recursive algorithm.

Corollary 3.4 The computational complexity of the unparallelizable processes within the ORA is

multiplication: $(l \vee m)^3(p^* + q^*) + \mathbf{O}(1)$

addition: $(p^* + q^*)\log_2(l \vee m) + p^* \log_2(p^* + q_0 + 2)$
$$+ (q^* - q_0)\log_2(p^* + q^* + 3) + q_0 \log_2(q_0 + 3) + \mathbf{O}(1)$$

∎

Remark 3.8 In the ORA of type I or type II, the parameters of the involved LMMSE predictors of the initial order are assumed to be calculated by using the Cholesky method. When the initial order is chosen as $(0, -1)$, we can easily determine the initial condition, from Definition 3.3 and (3.6)–(3.9), without any numerical computation. Some of the initial conditions are expressed below:

$$R^f(0,-1) = R^b(0,-1) = R^b(0,0) = R_{yy}(0)$$
$$V^f(0,-1) = V^b(0,-1) = R_{uu}(0).$$

The others are equal to zero matrices of appropriate dimension. ∎

3.6 Numerical Examples

Example 1: Consider a zero-mean jointly stationary process $\begin{pmatrix} y_n \\ u_n \end{pmatrix}$ associated with a stable and causal system:

$$y_n + 1.5y_{n-1} + 0.66y_{n-2} + 0.08y_{n-3} = u_n + 0.5u_{n-1} + \omega_n,$$

where

- $\{u_n\}$ is a white stationary process with $R_{uu}(0) = 1.0$,
- $\{\omega_n\}$ is a zero-mean white noise with $R_{\omega\omega}(0) = 1.0$,
- $Eu_{n+k}\omega_n = 0 \ \forall n{+}k,\ n \in Z_+$ and $Ey_{n+k}\omega_n = \begin{cases} 1 & k = 0 \ \ \forall n + k,\ n \in Z \\ 0 & k \neq 0 \ \ \forall n + k,\ n \in Z. \end{cases}$

By the assumption of joint stationarity and causality, we have :

$$\Gamma(3,1) = E(\ y_n\ y_{n-1}\ y_{n-2}\ y_{n-3}\ u_n\ u_{n-1}\ )^T(\ y_n\ y_{n-1}\ y_{n-2}\ y_{n-3}\ u_n\ u_{n-1}\ )$$

$$
= \left(\begin{array}{cccc|cc} 17.03961 & -15.75360 & 13.64455 & -11.43261 & 1. & -1. \\ -15.75360 & 17.03961 & -15.75360 & 13.64455 & 0. & 1. \\ 13.64455 & -15.75360 & 17.03961 & -15.75360 & 0. & 0. \\ -11.43261 & 13.64455 & -15.75360 & 17.03961 & 0. & 0. \\ & & & & & \\ \hline 1. & 0. & 0. & 0. & 1. & 0. \\ -1. & 1. & 0. & 0. & 0. & 1. \end{array} \right).
$$

The LMMSE estimation problem of order (p, q) with $q = 1$ and $1 \leq p \leq 3$ is solved by using the ORA in p. The computational results are expressed as follows:

Initial Condition: the case of initial order (p_0, q), $p_0 = 1$, $q = 1$:

Get the predictors of order (p_0, q) by solving some Yule-Walker equations:

$y_{n|n-1}(1,1) = -(0.919823 y_{n-1} - 1.000000 u_n + 0.080177 u_{n-1})$

$y_{n-1|n}(1,1) = -(0.982169 y_n - 0.982169 u_n)$

$u_{n|n-1}(1,1) = -(0.000000 y_{n-1} + 0.000000 u_{n-1})(= 0)$

$u_{n-1|n}(1,1) = -(0.0511687 y_n - 0.0113799 y_{n-1} - 0.0511687 u_n)$

Order Recursion in p

(0). Find the PEV's and intermediate variables:

$$
\begin{pmatrix} R^f(1,1) & R^b(1,1) & V^f(1,1) & V^b(1,1) & \alpha(1,1) \\ \beta(1,1) & \delta(1,1) & \eta(1,1) & \zeta(1,1) & \lambda(1,1) \end{pmatrix}
$$

$$
= \begin{pmatrix} 1.468909 & 1.566917 & 1.000000 & 0.937451 & -0.845976 \\ -0.846976 & 0.000000 & 1.000000 & 0.000000 & 0.877448 \end{pmatrix}
$$

(1). The case of order $(p_0 + 1, q)$:

(1.1). Find the partial correlation coefficients:

$(K_2(1,1) \quad K_3(1,1)) = (0.539899 \quad 0.000000)$

$(L_1(1,1) \quad L_3(1,1) \quad L_4(1,1)) = (0.575922 \quad 0.00 \quad 0.935993)$

$M_2(1,1) = 0.000000$

$N_2(1,1) = -0.461575$

(1.2). Obtain the LMMSE predictors:

$y_{n|n-1}(2,1) = -(1.450095y_{n-1} + 0.539899y_{n-2} - 1.000000u_n - 0.450095u_{n-1})$

$y_{n-2|n}(2,1) = -(0.623816y_n + 1.501263y_{n-1} - 0.623816u_n)$

$u_{n|n-1}(2,1) = -(0.000y_{n-1} + 0.000y_{n-2} + 0.000u_{n-1})(= 0)$

$u_{n-1|n}(2,1) = -(-0.236769y_n - 0.704326y_{n-1} - 0.461575y_{n-2} + 0.236769u_n)$

(1.3). Find the PEV's and auxiliary variables:

$$\begin{pmatrix} R^f(2,1) & R^b(2,1) & V^f(2,1) & V^b(2,1) & \alpha(2,1) \\ \beta(2,1) & \delta(2,1) & \eta(2,1) & \zeta(2,1) & \lambda(2,1) \end{pmatrix}$$
$$= \begin{pmatrix} 1.012167 & 1.900985 & 1.000000 & 0.532442 & -0.152079 \\ -0.152079 & 0.000000 & 1.000000 & 0.000000 & 0.368154 \end{pmatrix}$$

(2). The case of order $(p_0 + 2, q)$:

(2.1). Find the partial correlation coefficients:

$(\, K_2(2,1) \quad K_3(2,1)\,) = (\, 0.080000 \quad 0.000000\,)$

(2.2). Obtain the LMMSE predictor:

$y_{n|n-1}(3,1) = -(1.500000y_{n-1} + 0.660000y_{n-2} + 0.080000y_{n-3} - 1.000000u_n - 0.500000u_{n-1})$.

This example reflects how the ORA in p operates. Meanwhile, it confirms the correctness of the algorithm because the computation results are exactly the same as those computed by using the Gauss elimination method and the parameter estimates are equal to the parameters of the true plant in the case of $p = 3$. Furthermore, this example shows that the LMMSE forward and backward output predictors of order $(p, 1), p = 1, 2, 3$, are stable because they have the roots $\lambda_{1,1}(1) = -0.919823$, $\lambda_{2,1}(1) = -0.7250 + j0.1192$, $\lambda_{2,1}(2) = -0.7250 - j0.1192$, and $\lambda_{3,1}(1) = -0.2$, $\lambda_{3,1}(2) = -0.5$, $\lambda_{3,1}(3) = -0.8$.

Example 2: Consider a block-Toeplitz submatrix system of order (3,1) having the same form as Eq. (3.19). The matrix $\Gamma(3,1)$ is read as follows:

$$\Gamma(3,1) = \left(\begin{array}{cccc|cc} R_{yy}(0) & R_{yy}(1) & R_{yy}(2) & R_{yy}(3) & R_{yu}(0) & R_{yu}(1) \\ R_{yy}(-1) & R_{yy}(0) & R_{yy}(1) & R_{yy}(2) & R_{yu}(-1) & R_{yu}(0) \\ R_{yy}(-2) & R_{yy}(-1) & R_{yy}(0) & R_{yy}(1) & R_{yu}(-2) & R_{yu}(-1) \\ R_{yy}(-3) & R_{yy}(-2) & R_{yy}(-1) & R_{yy}(0) & R_{yu}(-3) & R_{yu}(-2) \\ \hline R_{uy}(0) & R_{uy}(1) & R_{uy}(2) & R_{uy}(3) & R_{uu}(0) & R_{uu}(1) \\ R_{uy}(-1) & R_{uy}(0) & R_{uy}(1) & R_{uy}(2) & R_{uu}(-1) & R_{uu}(0) \end{array}\right).$$

The block elements of the matrix $\Gamma(3,1)$ are known as

$$R_{yy}(0) = \begin{pmatrix} 9 & 0 & 1 \\ 0 & 6 & 0 \\ 1 & 0 & 3 \end{pmatrix}, \ R_{yy}(1) = \begin{pmatrix} 5 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 0 & 3 \end{pmatrix},$$

$$R_{yy}(2) = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \ R_{yy}(3) = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix};$$

$$R_{uu}(0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ R_{uu}(k) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad k \neq 0;$$

$$R_{yu}(1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, \ R_{yu}(0) = \begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \ R_{yu}(-k) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \ 1 \leq k \leq 3.$$

Note that the above blocks are not any ACM and CCM because the matrix $\Gamma(3,1)$ is not positive definite.

The computation results obtained by using the ORA in p are shown in Table 3.5. They are the same as those obtained by using Pro-Matlab[109]. This confirms that the new algorithm is valid for block-Toeplitz submatrix systems of linear equations. In addition, The simplified version of the ORA in p described in (3.45) – (3.47) is also confirmed by the results in Examples 1 and 2 since they are calculated by using the general version of the ORA in p.

Example 3: Suppose that we are given the auto-correlation and cross-correlation functions of a jointly stationary process $(y\ u)$:

$E(\ y_n\ y_{n-1}\ u_n\ u_{n-1}\ u_{n-2}\ u_{n-3})^T(\ y_n\ y_{n-1}\ u_n\ u_{n-1}\ u_{n-2}\ u_{n-3})$

solutions	i=0	i=1
$A_{1,i+1}$	−.57143 −.20000 −.14286 −.14285 −.60000 −.28571 .42857 .00000 −1.14285	-
$A_{2,i+1}$	−.26452 .00000 .07742 .09677 .00000 .48387 1.42581 1.00000 −1.27097	−.28065 −.12258 −.22903 −.12903 −.51613 −.80645 −.76774 −1.07097 .05161
$A_{3,i+1}$	−.21397 −.14231 −.65931 −.39111 −.99967 −1.87958 1.83480 .82966 −2.83463	−.37867 −.65931 −.29225 −.01310 −.87958 −.00400 −.77144 −1.83463 .04412
$A_{3,i+2}$	-	−.00231 .71936 .86399 .17250 1.13352 1.96069 −.57854 .93681 1.68568
$-C_{1,i}$	−2.00000 .00000 .00000 −1.00000 .00000 .00000	.14285 .20000 .28571 −.40000 −1.85714 .00000
$-C_{2,i}$	−2.00000 .00000 .00000 −1.00000 .00000 .00000	−.47097 .00000 −.19355 −1.00000 −3.85161 −1.00000
$-C_{3,i}$	−2.00000 .00000 .00000 −1.00000 .00000 .00000	−.57207 .14231 .78222 −.00033 −4.66960 −.82966
$B_{1,i+1}$	−1.00000 −.20000 .00000 −.21428 −.80000 .07143 .00000 −.20000 −1.00000	-
$B_{2,i+1}$	.17419 .69677 1.03871 −.61613 −1.46452 −1.52581 .40000 .60000 1.20000	−1.180645 −1.00 −.70323 .78710 1.00 1.13548 −.60000 −1.00 −1.80000
$-D_{1,i+1}$	2.00000 .20000 .42857 .80000 .00000 .20000	-
$-D_{2,i+1}$	2.36129 1.00000 −1.57419 −1.00000 1.20000 1.00000	-
$G_{1,i}$	.00000 .00000 .00000 .00000 .00000 .00000	-
$G_{2,i}$	-	.00000 .00000 .00000 .00000 .00000 .00000
$G_{2,i+1}$	-	.00000 .00000 .00000 .00000 .00000 .00000
$-H_{1,i}$	-	.00000 .00000 .00000 .00000
$-H_{2,i}$	-	.00000 .00000 .00000 .00000
$F_{1,i}$	−.69677 −.78710 −1.15484 −.04516 −.18065 −.23225	.72258 1.00000 .8120 .08387 .00000 .21936
$F_{2,i}$	.17350 .68237 1.30470 .05905 .56494 .74424	−.56510 −.88273 −.45016 −.08525 −.52032 −.13982
$F_{2,i+1}$	-	.27202 .20418 −.88273 .01227 −.23835 −.52032
$-E_{1,i+1}$	−1.44516 −1.00000 −.16774 .00000	-
$-E_{2,i+1}$	−.54404 −.20418 −.02455 .23835	-

Table 3.5: The solutions of a block-Toeplitz submatrix system of order $(p, 1)$ with $p = 1, 2, 3$

$$
= \left(\begin{array}{cc|cccc}
8.7510 & -6.7377 & 1.0187 & -1.9350 & 1.4684 & -0.7472 \\
-6.7377 & 8.7510 & -0.0346 & 1.9187 & -1.9350 & 1.4684 \\
\hline
1.0187 & -0.0346 & 1.0000 & -0.0183 & 0.0046 & -0.0017 \\
-1.9350 & 1.0187 & -0.0183 & 1.0000 & -0.0183 & 0.0046 \\
1.4684 & -1.9350 & 0.0046 & -0.0183 & 1.0000 & -0.0183 \\
-0.7472 & 1.4684 & -1.9350 & 0.0046 & -0.0183 & 1.0000
\end{array}\right).
$$

The LMMSE estimation problem of order (p,q) with $p=1$ and $0 \le q \le 2$ is solved by using the ORA in q. The computation results are expressed as follows

(1). The initial case of (p, q_0), $p=1$, $q_0=0$:

(1.1). Get the estimators of order (p_0, q):

$y_{n|n-1}(1,0) = -(0.7660 y_{n-1} - 0.9922 u_n)$,

$y_{n-0|n}(1,0) = -(0.8690 y_n - 0.8506 u_n)$,

$u_{n|n-1}(1,0) = -(0.0040 y_{n-1})$,

$u_{n-0|n}(0,0) = -(-0.1164 y_n)$

(1.2). Find the PEV's and intermediate variables:

$$
\begin{pmatrix}
R^f(1,0) & R^b(1,1) & V^f(1,0) & V^b(0,0) & \gamma(1,0) \\
\nu(1,1) & \eta(1,0) & \kappa(1,0) & \mu(0,0) & \xi(0,0)
\end{pmatrix}
= \begin{pmatrix}
2.5791 & 2.9257 & 0.9999 & 0.8814 & -1.1365 \\
-0.6472 & 0.9921 & -0.0143 & -1.1507 & -0.0143
\end{pmatrix}
$$

(2). The case of order (p, q_0+1) :

(2.1). Find the partial correlation coefficients:

$(\, P_3(1,0) \quad P_4(1,0) \,) = (\, 0.0184 \quad 1.2897 \,)$

$S_4(1,0) = 0.0162$

$Q_4(1,0) = 1.2361$

(2.2). Obtain the LMMSE estimators:

$y_{n|n-1}(1,1) = -(0.6159 y_{n-1} - 0.9738 u_n + 1.2897 u_{n-1})$

$u_{n|n-1}(1,1) = -(0.0021 y_{n-1} + 0.0162 u_{n-1})$

$u_{n-1|n}(0,1) = -(0.2485 y_n - 0.2348 u_n)$

$y_{n-1|n}(1,1) = -(1.1761 y_n - 1.1408 u_n + 1.2361 u_{n-1})$

(2.3). Find the PEV's and intermediate variables:

$$\begin{pmatrix} R^f(1,1) & R^b(1,2) & V^f(1,1) & V^b(0,1) & \gamma(1,1) \\ \nu(1,1) & \eta(1,1) & \kappa(1,1) & \mu(0,1) & \xi(0,1) \end{pmatrix}$$
$$= \begin{pmatrix} 1.1133 & 2.1257 & 0.9996 & 0.5235 & 0.2485 \\ -0.2359 & 0.9734 & 0.0003 & 0.2487 & 0.0003 \end{pmatrix}$$

(3). The case of order $(p, q_0 + 2)$:

(3.1). Find the partial correlation coefficients:

$$(P_3(1,1) \quad P_3(1,1)) = (0.00014 \quad -0.4746)$$

(3.2). Obtain the LMMSE predictor:

$$y_{n|n-1}(1,2) = -(0.4980 y_{n-1} - 0.9736 u_n + 1.4011 u_{n-1} - 0.4746 u_{n-2})$$
$$R^f(1,2) = 0.9954.$$

3.7 Summary

Motivated by the need for fast solutions to the LMMSE estimation problem of ARX systems, we have developed a new algorithm which we call the order-recursive algorithm for solving a block-Toeplitz submatrix system. Its main advantage over previous methods is that without over-parametrization of ARX models, it can significantly reduce the computational complexity in solving a set of LMMSE estimation problems of successively increasing order of either the autoregressive part or the moving-average part of exogenous inputs, and then those of order (p, q) within a rectangle $1 \leq p_0 \leq p \leq p^*, 0 \leq q_0 \leq q \leq q^*$. Based on the similarity between the LWWR algorithm[21] and the ORA and Cybenko's analysis of the stability of the Levinson-Durbin algorithm [38], we believe that the ORA is numerically stable.

The ORA is a generalization of the LWWR algorithm. Instead of only solving (block-)Toeplitz systems, the ORA can solve block-Toeplitz submatrix systems. During the derivation of the ORA in p and ORA in q, two extra input estimators are introduced and the Toeplitz structure of the submatrices is directly exploited. Due to the frequency with which Toeplitz submatrices appear in applications, the ORA is expected to have applications in many fields, both directly and indirectly, as the fundamental Levinson algorithm did. Some examples are briefly discussed below. (1) *Model Reduction:* The utilization of the ORA in model reduction can be seen from the relation between the L_2-norm of a transfer matrix and the ACM of its impulse response [39]. (2) *Least Squares Estimation of ARX and ARMA systems:* A least squares estimation problem of an ARX system is equivalent to the following minimization problem:

$$\min_{A_i, B_j} \sum_{n=1}^{N} \|\mathbf{y}_n + \sum_{i=1}^{p} A_i \mathbf{y}_{n-i} - \sum_{j=0}^{q} C_j \mathbf{u}_{n-j}\|^2, \tag{3.69}$$

where $\mathbf{y}_n$ and $\mathbf{u}_n, n = 1, 2, \cdots$, are output and input observations. As discussed before, problem (3.69) can be approximately solved by using the ORA and the approximation error will be very small when the length of the data sequences N is long. On the other hand, problem (3.69) also appears in the three-stage least squares algorithm, which is a popular and successful method for estimation of ARMA systems[63]. In this case, the $\mathbf{u}_n, n = 1, 2, \cdots$, are some estimates of unmeasurable noise. Consequently, the ORA's can be applied to the linear filtering, spectral estimation, identification of ARX and ARMA systems, and self-tuning control as an efficient computation algorithm. In fact, this is the main topic of Chapter 4. (3) *Development of Other Computation Algorithms:* Often, an appropriate algorithm requires modification to fit the special application. Thus, the marriage of the ORA and its applications could stimulate users to extend and generalize it. On the other hand, linear filtering of ARX and ARMA systems and general Toeplitz submatrix systems of linear equations lack efficient algorithms featuring order-recursion. A way to overcome the difficulty has been suggested by the ORA through the introduction of some extra forward and backward predictors as the auxiliary solutions.

Chapter 4

A Fast Time & Order Recursive Algorithm for Parameter Estimation of a Set of ARX Models

In the previous chapter we assumed that exact values for the second order statistics were known, and the system output and input were jointly-stationary. The first assumption is not normally true in practice. It is much more realistic to assume that only measurements of system input $\mathbf{u}$ and output $\mathbf{y}$ are available. The second assumption usually does not hold in adaptive control and adaptive signal processing. In this chapter, a fast time and order recursive algorithm (TORA) is developed for parameter estimation of a family of ARX models based on measurements of system outputs and inputs. The algorithm is actually an extension of the fast order-recursive algorithm developed in Chapter 3 for solving Toeplitz submatrix systems. For uniformly bounded input and output processes, the strong consistency of the parameter estimates generated by the recursive LS algorithm guarantees that the TORA produces strongly consistent parameter estimates. The TORA can be useful, by combining it with other relevant techniques, in stochastic modeling, adaptive IIR filtering, quantification of unmodeled dynamics, and self-tuning control.

4.1 Preliminaries

In this chapter we consider the development of a least-squares (LS) parameter estimation algorithm for *a family* of MIMO ARX models, given observed time series of l-dimensional input $\{\mathbf{u}_n\}_{n=0}^{N}$ and m-dimensional output $\{\mathbf{y}_n\}_{n=0}^{N}$ but not given any statistics of these measurements. The ARX models represent autoregressive models with exogenous inputs described as

$$\mathbf{y}_n + \sum_{i=1}^{p} A_i \mathbf{y}_{n-i} = \sum_{j=0}^{q} C_j \mathbf{u}_{n-j} + w_n, \tag{4.1}$$

where $w_n, n \geq 0$, are noise vectors and the pair of integers (p, q) is defined as the model order[1]. Specifically, we intend to develop an algorithm which is able to efficiently and simultaneously determine all minimizers of *the family* of minimization problems described below:

$$\left\{ \min_{A_i, C_j} \{ \frac{1}{N} \sum_{k=1}^{N} \|\mathbf{e}_n(p,q)\|^2 \} \;\middle|\; (p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*} \right\} \tag{4.2}$$

for any $N \geq 1$, where

$$\mathbf{e}_n(p,q) \triangleq \mathbf{y}_n + \sum_{i=1}^{p} A_i \mathbf{y}_{n-i} - \sum_{j=0}^{q} C_j \mathbf{u}_{n-j}, \tag{4.3}$$

and the model complexity set $\mathcal{O}_{0,-1}^{p^*,q^*}$ is defined as

$$\mathcal{O}_{0,-1}^{p^*,q^*} \triangleq \{(p,q); 0 \leq p \leq p^*, \; -1 \leq q \leq q^*\}. \tag{4.4}$$

The norm used in (4.2) is the l_2-norm. Obviously, problem (4.2) degenerates into a standard LS parameter estimation problem when the model complexity set is a singleton.

[1] Here, we have made the convention, which will be used throughout this chapter, that if $a > b$, then $\sum_{i=a}^{b} f(i) = 0$ regardless of $f(i)$. Thus, an ARX model reduces to a pure AR model when $q = -1$ or an all-zero ARX model when $p = 0$.

Motivation: Within the framework of exact modeling of linear systems order estimation by minimization of the CIC criterion or the criterion of accumulated prediction error has been shown to be strongly consistent even when the input and output processes of the true system are neither stationary or ergodic [29][71]. Thus, this technique shows great potential as a solution to this identification problem. This approach essentially proceeds by minimizing the criteria over a finite set of linear predictors of different order which are obtained by some optimization techniques, say the LS technique. The computation of a performance measure usually does not form a major computational load for each time iteration. But the computation of the set of predictors, say LS moving average predictors with exogenous inputs (MAX predictors), can dominate the computation time if there are many candidate models. This is often the case since the prior knowledge of the model is often poor.

Within the framework of approximate system modeling for control system design, much research is being done on the integration of performance objectives of the control system into the modeling process. This effort mainly focuses on the quantification of the model uncertainty and its acceptable level. The model uncertainty, depending on the model order, consists of the unmodeled dynamics and the variance of the estimated parameters. Goodwin et al. developed the stochastic embedding approach [56], where the uncertainty bound for a model of given order is computed by solving a recursive least squares (RLS) problem. Alternatively, the effect from unmodeled dynamics can be described as bounded disturbances and studied by incorporating the measured data into prior information [42][88][97]. Some effective tools for this approach are set membership identification. For a model with given order, bounds on the disturbances are obtained by solving a (weighted) RLS problem. The solution to the problem gives out the parameters of a model which has the smallest bound on disturbances. Consequently, the availability of all solutions of a set of LS problems of different order would help the trade-off between controller requirements and the level of the model uncertainty.

In addition to the concerns about controls, as shown in Section 2.2, the proposed TORA can be very helpful in on-line order estimation for adaptive IIR filters as the well-known Levinson algorithm for all-pole filters[138].

Background: The most previous results related to the subject in this chapter have been reviewed in Chapter 2. Here, we discuss some extra things.

Problem (4.2) can be efficiently solved by using the Levinson-type algorithms in the special case that all models in question satisfy $p = q$ [62][148]. However, they cannot be applied to determining LS parameter estimates for a set of ARX models with different p and q. For this more general case, people may suggest some algorithms with suboptimal performance but low computational complexity. Indeed, an over-parametrization method has been implicitly suggested in [62][148]: (1) Use *over-parameterized* ARX models with $p = q$ for parameter estimation; (2) Determine the LS parameter estimates for these models via the Levinson-type algorithms; (3) Truncate the LS parameter estimates to arrive at parameter estimates with different p and q. However, as will be shown in Section 4.5, the performance of the algorithm is not good because the over-parametrization causes the parameter estimates to have large variance.

Another effort for ARX modeling and IIR filtering has been directed towards the development of fast parallel algorithms in lattice filter structures [84][85]. These algorithms can take advantage of parallel processing to obtain all LS ARX models of order (p, q) lower than some order. But these algorithms can only process data from SISO systems and use more recursion terms than the algorithm proposed here does.

Approach: It is well known that determining LS parameter estimates of the ARX model (4.1), denoted by $(\hat{A}^{p,q}_{N,1} \ \ldots \ \hat{A}^{p,q}_{N,p} \ -\hat{C}^{p,q}_{N,0} \ \ldots \ -\hat{C}^{p,q}_{N,q})$, is mathematically equivalent to solving an augmented normal matrix[102]. For instance, consider a SISO ARX system of order $(3, 1)$. The corresponding augmented normal matrix is

$$
\left(1 \quad \hat{a}_1^{3,1} \quad \hat{a}_2^{3,1} \quad \hat{a}_3^{3,1} \quad -\hat{c}_0^{3,1} \quad -\hat{a}_1^{3,1} \right) \times
$$

$$
\frac{1}{N}\sum_{n=1}^{N} \left(\begin{array}{cccc|cc}
y_n y_n & y_n y_{n-1} & y_n y_{n-2} & y_n y_{n-3} & y_n u_n & y_n u_{n-1} \\
y_{n-1} y_n & y_{n-1} y_{n-1} & y_{n-1} y_{n-2} & y_{n-1} y_{n-3} & y_{n-1} u_n & y_{n-1} u_{n-1} \\
y_{n-2} y_n & y_{n-2} y_{n-1} & y_{n-2} y_{n-2} & y_{n-2} y_{n-3} & y_{n-2} u_n & y_{n-2} u_{n-1} \\
y_{n-3} y_n & y_{n-3} y_{n-1} & y_{n-3} y_{n-2} & y_{n-3} y_{n-3} & y_{n-3} u_n & y_{n-3} u_{n-1} \\
\hline
u_n y_n & u_n y_{n-1} & u_n y_{n-2} & u_n y_{n-3} & u_n u_n & u_n u_{n-1} \\
u_{n-1} y_n & u_{n-1} y_{n-1} & u_{n-1} y_{n-2} & u_{n-1} y_{n-3} & u_{n-1} u_n & u_{n-1} u_{n-1}
\end{array} \right)
$$

$$
= \left(r^f(3,1) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right).
$$

Carefully checking the augmented normal matrix, we can see that the following *submatrix-block-Toeplitz matrix:*

$$
\tilde{\Gamma}_N(3,1) \triangleq \frac{1}{N}\sum_{n=1}^{N} \left(\begin{array}{cccc|cc}
y_n y_n & y_n y_{n-1} & y_n y_{n-2} & y_n y_{n-3} & y_n u_n & y_n u_{n-1} \\
y_{n-1} y_n & y_n y_n & y_n y_{n-1} & y_n y_{n-2} & y_{n-1} u_n & y_n u_n \\
y_{n-2} y_n & y_{n-1} y_n & y_n y_n & y_n y_{n-1} & y_{n-2} u_n & y_{n-1} u_n \\
y_{n-3} y_n & y_{n-2} y_n & y_{n-1} y_n & y_n y_n & y_{n-3} u_n & y_{n-2} u_n \\
\hline
u_n y_n & u_n y_{n-1} & u_n y_{n-2} & u_n y_{n-3} & u_n u_n & u_n u_{n-1} \\
u_{n-1} y_n & u_n y_n & u_n y_{n-1} & u_n y_{n-2} & u_{n-1} u_n & u_n u_n
\end{array} \right)
$$

is very close to the augmented normal matrix. Then, it is reasonable to believe that the corresponding solution

$$
\left(1 \quad \tilde{a}_1^{3,1} \quad \tilde{a}_2^{3,1} \quad \tilde{a}_3^{3,1} \quad -\tilde{c}_0^{3,1} \quad -\tilde{c}_1^{3,1} \right) \tilde{\Gamma}(3,1) = \left(\tilde{r}^f(3,1) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right)
$$

is also close to the desired solution $\left(1 \quad \hat{a}_1^{3,1} \quad \hat{a}_2^{3,1} \quad \hat{a}_3^{3,1} \quad -\hat{c}_0^{3,1} \quad -\hat{c}_1^{3,1} \right)$. Actually, this observation is the idea of how we develop the TORA.

This chapter is an extension of Chapter 3. We will develop, in three steps, a *fast* algorithm, which we call TORA (for Time and Order Recursive Algorithm). At the first step, an LS parameter estimation problem of order (p^*, q^*) will be transformed into a problem of solving a sequence of special systems of linear equations of order (p^*, q^*), which are *very close* to the augmented normal equations associated with the LS problem. For each time instant, the coefficient matrix in the system can be partitioned into a submatrix-block-Toeplitz matrix defined in Definition 3.1. As a result, the special systems are actually block-Toeplitz submatrix systems, which are also defined in Definition 3.1. At the second step, we simply use the order-recursive algorithm (ORA) developed in Chapter 3 to solve each block-Toeplitz submatrix system in an order-recursive manner. At the last step, a simple time-recursion will be built up for updating the blocks in the block-Toeplitz submatrix system as time advances. Consequently, the incorporation of the time-recursion and ORA results in the *fast* TORA.

Often, algorithms with suboptimal performance are used in practice to reduce computational burden. The TORA is such an algorithm, but it is also an asymptotically optimal algorithm. For uniformly bounded input and output processes, the parameter estimates generated by the TORA will asymptotically converge to the parameter estimates produced by the ordinary recursive least squares algorithm (RLS) with probability one if some mild conditions are satisfied. In other words, the approximation error converges to zero. Thus, the TORA asymptotically preserves many useful properties of the RLS. As a result, computing a set of LS ARX models simultaneously and recursively is no longer a prohibitive task.

Outline: The TORA is composed of the time-recursion and the ORA. This chapter is organized as follows: Section 4.2 introduces a parameter estimation problem whose solution is equivalent to the solutions to a sequences of Toeplitz submatrix systems. Then, some properties of the systems are explored. Section 4.3 presents the TORA and explains its computational complexity, parallelism, and the physical and mathematical meaning of the computed results. Section 4.4 contains the asymptotic properties of the TORA. Section 4.5 expresses some numerical examples. Section 4.6 provides some conclusions and potential generalizations.

4.2 LS Problems And Toeplitz Submatrix Systems

In this section, we formulate the TORA and LS parameter estimation problems for ARX systems and then establish a quantitative relation between the solutions to these two problems. The resulting formulation will be used in Section 4.3 for deriving the TORA. The quantitative relation will be used in Section 4.4 for investigating asymptotical properties of the TORA.

4.2.1 The TORA estimation Problem

Consider the following predictor described as

$$\mathbf{y}_{\mathrm{N|N\text{-}1}}(p,q) = -\sum_{i=1}^{p} A_{\mathrm{N,i}}^{\mathrm{p,q}}\mathbf{y}_{\mathrm{N\text{-}i}} + \sum_{j=0}^{q} C_{\mathrm{N,j}}^{\mathrm{p,q}}\mathbf{u}_{\mathrm{N\text{-}j}}, \qquad (p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}. \tag{4.5}$$

The predictor parameters $A_{\mathrm{N,i}}^{\mathrm{p,q}} \in R^{m\times m}$ and $C_{\mathrm{N,j}}^{\mathrm{p,q}} \in R^{m\times l}$ are determined by solving the following block-Toeplitz submatrix system

$$\begin{pmatrix} I & \tilde{A}_{\mathrm{N,1}}^{\mathrm{p,q}} & \cdots & \tilde{A}_{\mathrm{N,p}}^{\mathrm{p,q}} & 0 & -\tilde{C}_{\mathrm{N,0}}^{\mathrm{p,q}} & -\tilde{C}_{\mathrm{N,1}}^{\mathrm{p,q}} & \cdots & -\tilde{C}_{\mathrm{N,q}}^{\mathrm{p,q}} \\ 0 & \tilde{B}_{\mathrm{N,p}}^{\mathrm{p,q}} & \cdots & \tilde{B}_{\mathrm{N,1}}^{\mathrm{p,q}} & I & 0 & -\tilde{D}_{\mathrm{N,q}}^{\mathrm{p,q}} & \cdots & -\tilde{D}_{\mathrm{N,1}}^{\mathrm{p,q}} \\ 0 & -\tilde{H}_{\mathrm{N,1}}^{\mathrm{p,q}} & \cdots & -\tilde{H}_{\mathrm{N,p}}^{\mathrm{p,q}} & 0 & I & \tilde{G}_{\mathrm{N,1}}^{\mathrm{p,q}} & \cdots & \tilde{G}_{\mathrm{N,q}}^{\mathrm{p,q}} \\ -\tilde{F}_{\mathrm{N,p}}^{\mathrm{p,q}} & -\tilde{F}_{\mathrm{N,p\text{-}1}}^{\mathrm{p,q}} & \cdots & -\tilde{F}_{\mathrm{N,0}}^{\mathrm{p,q}} & 0 & -\tilde{F}_{\mathrm{N,q}}^{\mathrm{p,q}} & -\tilde{F}_{\mathrm{N,q\text{-}1}}^{\mathrm{p,q}} & \cdots & I \end{pmatrix} \times \tilde{\Gamma}_{\mathrm{N}}(p+1,q)$$

$$= \begin{pmatrix} \tilde{R}_{\mathrm{N}}^{\mathrm{f}}(p,q) & \overbrace{0 \;\cdots\; 0}^{p} & \tilde{\alpha}_{\mathrm{N}}(p,q) & 0 & \overbrace{0 \;\cdots\; 0}^{q} \\ \tilde{\beta}_{\mathrm{N}}(p,q) & 0 \;\cdots\; 0 & \tilde{R}_{\mathrm{N}}^{\mathrm{b}}(p,q) & \tilde{\delta}_{\mathrm{N}}(p,q) & 0 \;\cdots\; 0 \\ \tilde{\eta}_{\mathrm{N}}(p,q) & 0 \;\cdots\; 0 & \tilde{\zeta}_{\mathrm{N}}(p,q) & \tilde{V}_{\mathrm{N}}^{\mathrm{f}}(p,q) & 0 \;\cdots\; 0 \\ 0 & 0 \;\cdots\; 0 & \tilde{\lambda}_{\mathrm{N}}(p,q) & 0 & 0 \;\cdots\; \tilde{V}_{\mathrm{N}}^{\mathrm{b}}(p,q) \end{pmatrix}, \tag{4.6}$$

where the superscript of the unknowns on the left side is used to reflect the effect of the order of the predictors in question. The first subscript represents the time instant and the second is an index. The matrix $\tilde{\Gamma}_{\mathrm{N}}(p+1,q)$ is the same as the matrix $\Gamma(p+1,q)$ defined in (3.19) except that the block elements there, $R_{yy}(\cdot)$, $R_{yu}(\cdot)$, and $R_{uu}(\cdot)$, are replaced by the new block elements $R_{\mathrm{N}}^{yy}(\cdot)$, $R_{\mathrm{N}}^{yy}(\cdot)$, and $R_N^{yy}(\cdot)$. These block elements are defined in the following way:

$$
\begin{aligned}
R_N^{yy}(n) &\triangleq \frac{1}{N}\sum_{k=1}^{N} \mathbf{y}_k\mathbf{y}_{k-n}^T + \delta(n)\frac{1}{Nr_0}I = (R_N^{yy}(-n))^T,\\
R_N^{yu}(n) &\triangleq \frac{1}{N}\sum_{k=1}^{N} \mathbf{y}_k\mathbf{u}_{k-n}^T = (R_N^{uy}(-n))^T,\\
R_N^{yu}(-n) &\triangleq \frac{1}{N}\sum_{k=1}^{N} \mathbf{y}_{k-n}\mathbf{u}_k^T = (R_N^{uy}(n))^T,\\
R_N^{uu}(n) &\triangleq \frac{1}{N}\sum_{k=1}^{N} \mathbf{u}_k\mathbf{u}_{k-n}^T + \delta(n)\frac{1}{Nr_0}I = (R_N^{uu}(-n))^T,
\end{aligned}
\tag{4.7}
$$

where $n = 0, 1, 2, \cdots$. $\delta(\cdot)$ is the δ-function and $r_0(> 0)$ is an initial value. The extra term involving $\delta(\cdot)$ is introduced to guarantee matrix $\tilde{\Gamma}_N(p+1)$ is invertible for each time instant and this is also sufficient to guarantee that the TORA is able to perform parameter estimation properly (See Theorem 3.1 for details).

Remark 4.1 Notice that the blocks of matrix $\tilde{\Gamma}_N(p+1)$ can be viewed as estimates of autocorrelation and cross-correlation matrices when $\{\mathbf{y}_n\}_{n=1}^{\infty}$ and $\{\mathbf{u}_n\}_{n=1}^{\infty}$ are a realization of some jointly-stationary processes $\mathbf{y}$ and $\mathbf{u}$. But, generally, they cannot be interpreted in this way. ∎

Remark 4.2 Notice that it is to reduce the computational complexity of problem (4.2) that we introduce some extra unknowns on the left side of Eq. (4.6). Referring to Eqs. (3.7) to (3.9) and Eq. (3.19), we can see that they are actually the parameters of the following auxiliary estimators:

$$
\mathbf{y}_{N-p|N}(p,q) = -\sum_{i=1}^{p} B_{N,i}^{p,q}\mathbf{y}_{N-p+i} + \sum_{j=1}^{q} D_{N,j}^{p,q}\mathbf{u}_{N-q+j}, \qquad (p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}, \tag{4.8}
$$

$$
\mathbf{u}_{N|N-1}(p,q) = \sum_{i=1}^{p} H_{N,i}^{p,q}\mathbf{y}_{N-i} - \sum_{j=1}^{q} G_{N,j}^{p,q}\mathbf{u}_{N-j}, \qquad (p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}, \tag{4.9}
$$

$$
\mathbf{u}_{N-q|N}(p,q) = \sum_{i=0}^{p} F_{N,i}^{p,q}\mathbf{y}_{N-p+i} - \sum_{j=1}^{q} E_{N,j}^{p,q}\mathbf{u}_{N-q+j}, \qquad (p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}, \tag{4.10}
$$

They can be useful in some other applications, as discussed in section 3.2.1. ∎

To compare the TORA parameter estimates with LS parameter estimates, we need to explore some properties of the block-Toeplitz submatrix systems described in (4.6). From Eq.(4.6), one can deduce four normal equations:

$$\begin{aligned}
\tilde{\theta}^{\mathrm{T}}_{\mathrm{f,N}}(p,q)\tilde{R}^{1}_{\mathrm{N}}(p,q) &= -(\ \tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(1)\ \cdots\ \tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(p)\ \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(0)\ \cdots\ \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(q)\ ),\\
\tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)\tilde{R}^{2}_{\mathrm{N}}(p,q) &= -(\ \tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(-p)\ \cdots\ \tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(-1)\ \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(-p)\ \cdots\ \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(q-p-1)\ ),\\
\tilde{\chi}^{\mathrm{T}}_{\mathrm{f,N}}(p,q)\tilde{R}^{3}_{\mathrm{N}}(p,q) &= -(\ \tilde{R}^{\mathrm{uy}}_{\mathrm{N}}(1)\ \cdots\ \tilde{R}^{\mathrm{uy}}_{\mathrm{N}}(p)\ \tilde{R}^{\mathrm{uu}}_{\mathrm{N}}(1)\ \cdots\ \tilde{R}^{\mathrm{uu}}_{\mathrm{N}}(q)\ ),\\
\tilde{\chi}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)\tilde{R}^{4}_{\mathrm{N}}(p,q) &= -(\ \tilde{R}^{\mathrm{uy}}_{\mathrm{N}}(-q)\ \cdots\ \tilde{R}^{\mathrm{uy}}_{\mathrm{N}}(-q+p)\ \tilde{R}^{\mathrm{uu}}_{\mathrm{N}}(-q)\ \cdots\ \tilde{R}^{\mathrm{uu}}_{\mathrm{N}}(-1)\ ),
\end{aligned} \tag{4.11}$$

where the normal matrices $\tilde{R}^{1}_{\mathrm{N}}(p,q)$, $\tilde{R}^{2}_{\mathrm{N}}(p,q)$, $\tilde{R}^{3}_{\mathrm{N}}(p,q)$, and $\tilde{R}^{4}_{\mathrm{N}}(p,q)$ are defined in Definition 3.2, with a small modification that the matrix $\Gamma(p,q)$ there is replaced by the matrix $\tilde{\Gamma}_{\mathrm{N}}(p,q)$.

Remark 4.3 It follows from Definition 3.1 that

$$\tilde{R}^{4}(p,q) = \tilde{\Gamma}(p,q-1). \tag{4.12}$$

∎

The solutions to the normal equations, $\tilde{\theta}^{\mathrm{T}}_{\mathrm{f,N}}(p,q), \tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q), \tilde{\chi}^{\mathrm{T}}_{\mathrm{f,N}}(p,q)$, and $\tilde{\chi}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)$, can be expressed as

$$\begin{aligned}
\tilde{\theta}^{\mathrm{T}}_{\mathrm{f,N}}(p,q) &\triangleq (\ \tilde{A}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \ldots\ \tilde{A}^{\mathrm{p,q}}_{\mathrm{N,p}}\ -\tilde{C}^{\mathrm{p,q}}_{\mathrm{N,0}}\ -\tilde{C}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \ldots\ -\tilde{C}^{\mathrm{p,q}}_{\mathrm{N,q}}\ ),\\
\tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q) &\triangleq (\ \tilde{B}^{\mathrm{p,q}}_{\mathrm{N,p}}\ \ldots\ \tilde{B}^{\mathrm{p,q}}_{\mathrm{N,1}}\ -\tilde{D}^{\mathrm{p,q}}_{\mathrm{N,q}}\ \ldots\ -\tilde{D}^{\mathrm{p,q}}_{\mathrm{N,1}}\ ),\\
\tilde{\chi}^{\mathrm{T}}_{\mathrm{f,N}}(p,q) &\triangleq (\ -\tilde{H}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \ldots\ -\tilde{H}^{\mathrm{p,q}}_{\mathrm{N,p}}\ \tilde{G}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \ldots\ \tilde{G}^{\mathrm{p,q}}_{\mathrm{N,q}}\ ),\\
\tilde{\chi}^{\mathrm{T}}_{\mathrm{b,N}}(p,q) &\triangleq (\ -\tilde{F}^{\mathrm{p,q}}_{\mathrm{N,p}}\ -\tilde{F}^{\mathrm{p,q}}_{\mathrm{N,p-1}}\ \ldots\ -\tilde{F}^{\mathrm{p,q}}_{\mathrm{N,0}}\ \tilde{E}^{\mathrm{p,q}}_{\mathrm{N,q}}\ \ldots\ \tilde{E}^{\mathrm{p,q}}_{\mathrm{N,1}}\ ),
\end{aligned} \tag{4.13}$$

which we call the TORA parameter estimates or the parameters of TORA estimators in (4.5) and (4.8) - (4.10).

4.2.2 The LS Parameter Estimation Problem

When the parameters of the estimators in (4.5) and (4.8)–(4.10) are determined by using the LS algorithm, the resulting estimators are called the LS estimators. To express the parameters of LS estimators, define the following regression vectors,

$$
\begin{aligned}
\phi^{\mathrm{T}}_{\mathrm{f,N\text{-}1}}(p,q) &\triangleq (\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}p}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}q}}\ ),\\
\phi^{\mathrm{T}}_{\mathrm{b,N\text{-}1}}(p,q) &\triangleq (\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}(p\text{-}1)}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}(q\text{-}1)}}\ ),\\
\psi^{\mathrm{T}}_{\mathrm{f,N\text{-}1}}(p,q) &\triangleq (\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}p}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}q}}\ ) \equiv \phi^{\mathrm{T}}_{\mathrm{b,N\text{-}2}}(p,q),\\
\psi^{\mathrm{T}}_{\mathrm{b,N\text{-}1}}(p,q) &\triangleq (\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}p}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}(q\text{-}1)}}\ ) \equiv \phi^{\mathrm{T}}_{\mathrm{b,N\text{-}1}}(p+1,q).
\end{aligned}
\tag{4.14}
$$

Thus, the parameters of the LS estimators, obtained by using the RLS[102], can be described as follows:

$$
\begin{aligned}
\hat{\theta}^{\mathrm{T}}_{\mathrm{f,N}}(p,q) &\triangleq (\ \hat{A}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \cdots\ \hat{A}^{\mathrm{p,q}}_{\mathrm{N,p}}\ \ -\hat{C}^{\mathrm{p,q}}_{\mathrm{N,0}}\ \cdots\ -\hat{C}^{\mathrm{p,q}}_{\mathrm{N,q}}\ )\\
&= -(\textstyle\sum_{k=1}^{N} \mathbf{y}_{\mathrm{k}}\phi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q))\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q) + \frac{1}{r_0}\hat{\theta}^{\mathrm{T}}_{\mathrm{f,0}}(p,q)\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q),\\
\\
\hat{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q) &\triangleq (\ \hat{B}^{\mathrm{p,q}}_{\mathrm{N,p}}\ \cdots\ \hat{B}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \ -\hat{D}^{\mathrm{p,q}}_{\mathrm{N,q}}\ \cdots\ -\hat{D}^{\mathrm{p,q}}_{\mathrm{N,1}}\ )\\
&= -(\textstyle\sum_{k=1}^{N} \mathbf{y}_{\mathrm{k\text{-}p}}\phi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q))\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}(p,q) + \frac{1}{r_0}\hat{\theta}^{\mathrm{T}}_{\mathrm{b,0}}(p,q)\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}(p,q),\\
\\
\hat{\chi}^{\mathrm{T}}_{\mathrm{f,N}}(p,q) &\triangleq (\ -\hat{H}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \cdots\ -\hat{H}^{\mathrm{p,q}}_{\mathrm{N,p}}\ \ \hat{G}^{\mathrm{p,q}}_{\mathrm{N,1}}\ \cdots\ \hat{G}^{\mathrm{p,q}}_{\mathrm{N,q}}\ )\\
&= -(\textstyle\sum_{k=1}^{N} \mathbf{u}_{\mathrm{k}}\psi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q))\hat{P}^{\mathrm{u}}_{\mathrm{f,N}}(p,q) + \frac{1}{r_0}\hat{\chi}^{\mathrm{T}}_{\mathrm{f,0}}(p,q)\hat{P}^{\mathrm{u}}_{\mathrm{f,N}}(p,q),\\
\\
\hat{\chi}^{\mathrm{T}}_{\mathrm{b,N}}(p,q) &\triangleq (\ -\hat{F}^{\mathrm{p,q}}_{\mathrm{N,p}}\ \cdots\ -\hat{F}^{\mathrm{p,q}}_{\mathrm{N,0}}\ \ \hat{E}^{\mathrm{p,q}}_{\mathrm{N,q}}\ \cdots\ \hat{E}^{\mathrm{p,q}}_{\mathrm{N,1}}\ )\\
&= -(\textstyle\sum_{k=1}^{N} \mathbf{u}_{\mathrm{k\text{-}q}}\psi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q))\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}(p,q) + \frac{1}{r_0}\hat{\chi}^{\mathrm{T}}_{\mathrm{b,0}}(p,q)\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}(p,q),
\end{aligned}
\tag{4.15}
$$

where the gain matrices are

$$
\begin{aligned}
\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q) &= (\textstyle\sum_{k=1}^{N} \phi_{\mathrm{f,k\text{-}1}}(p,q)\phi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q) + \frac{1}{r_0}I)^{-1},\\
\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}(p,q) &= (\textstyle\sum_{k=1}^{N} \phi_{\mathrm{b,k\text{-}1}}(p,q)\phi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q) + \frac{1}{r_0}I)^{-1},\\
\hat{P}^{\mathrm{u}}_{\mathrm{f,N}}(p,q) &= (\textstyle\sum_{k=1}^{N} \psi_{\mathrm{f,k\text{-}1}}(p,q)\psi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q) + \frac{1}{r_0}I)^{-1},\\
\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}(p,q) &= (\textstyle\sum_{k=1}^{N} \psi_{\mathrm{b,k\text{-}1}}(p,q)\psi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q) + \frac{1}{r_0}I)^{-1}.
\end{aligned}
\tag{4.16}
$$

4.2.3 The Relation Between TORA and LS Parameter Estimates

The difference between the parameters of TORA estimators and those of LS estimators is actually due to the deliberate perturbation vectors defined as, for

$i = 0, 1, 2, \ldots,$

$$f_{N,1}(i+1,p,q) = \begin{pmatrix} Z_p \otimes I_m & 0 \\ 0 & Z_{q+1} \otimes I_l \end{pmatrix}^i f_{N,1}(i,p,q),$$

$$f_{N,2}(i+1,p,q) = \begin{pmatrix} Z_p \otimes I_m & 0 \\ 0 & Z_q \otimes I_l \end{pmatrix}^i f_{N,2}(i,p,q),$$

$$f_{N,3}(i+1,p,q) = \begin{pmatrix} Z_p \otimes I_m & 0 \\ 0 & Z_q \otimes I_l \end{pmatrix}^i f_{N,3}(i,p,q),$$

$$f_{N,1}(i+1,p,q) = \begin{pmatrix} Z_{p+1} \otimes I_m & 0 \\ 0 & Z_q \otimes I_l \end{pmatrix}^i f_{N,4}(i,p,q),$$

and

$$f^T_{N,1}(0,p,q) \triangleq (\mathbf{y}^T_N \ \mathbf{y}^T_{N-1} \ \cdots \ \mathbf{y}^T_{N-(p-1)} \ \vdots \ \mathbf{0}^T \ \mathbf{u}^T_N \ \mathbf{u}^T_{N-1} \ \cdots \ \mathbf{u}^T_{N-(q-1)}),$$

$$f^T_{N,2}(0,p,q) \triangleq (\mathbf{0}^T \ \mathbf{y}^T_N \ \mathbf{y}^T_{N-1} \ \cdots \ \mathbf{y}^T_{N-(p-2)} \ \vdots \ \mathbf{0}^T \ \mathbf{u}^T_N \ \mathbf{u}^T_{N-1} \ \cdots \ \mathbf{u}^T_{N-(q-2)}),$$

$$f^T_{N,3}(0,p,q) \triangleq (\mathbf{y}^T_N \ \mathbf{y}^T_{N-1} \ \cdots \ \mathbf{y}^T_{N-(p-1)} \ \vdots \ \mathbf{u}^T_N \ \mathbf{u}^T_{N-1} \ \cdots \ \mathbf{u}^T_{N-(q-1)}),$$

$$f^T_{N,4}(0,p,q) \triangleq (\mathbf{0}^T \ \mathbf{y}^T_N \ \mathbf{y}^T_{N-1} \ \cdots \ \mathbf{y}^T_{N-(p-1)} \ \vdots \ \mathbf{0}^T \ \mathbf{u}^T_N \ \mathbf{u}^T_{N-1} \ \cdots \ \mathbf{u}^T_{N-(q-2)}),$$

where Z_k is the $k \times k$ lower shift matrix, zero everywhere except for 1's on the first subdiagonal, $\otimes$ represents the Kronecker product, and I_k is a $k \times k$ identity matrix.

The effect of the perturbation vectors on the parameters of TORA estimators is explicitly expressed in Lemma 4.1, Lemma 4.2, and Lemma 4.3. (Their proofs will be given in the Appendix B.)

Lemma 4.1 Suppose that system output and input measurements are one-sided sequences. That is, $\mathbf{y}_{-k} = 0$ if $0 \le k \le p^*$, and $\mathbf{u}_{-k} = 0$ if $0 \le k \le q^*$. Thus, the normal matrices in Eq.(4.6) can be decomposed into a sum of outer products of perturbation vectors and the properly scaled inverses of the gain matrices in (4.16). The decomposition is described below:

$$\begin{aligned} \tilde{R}^1_N(p,q) &= \tfrac{1}{N}(\hat{P}^y_{f,N}(p,q))^{-1} + M_{N,1}(p,q) \\ &= \tfrac{1}{N}\sum_{k=1}^{N} \phi_{f,k-1}(p,q)\phi^T_{f,k-1}(p,q) + \tfrac{1}{Nr_0} I + M_{N,2}(p,q) \end{aligned} \tag{4.17}$$

$$\begin{aligned} \tilde{R}^2_N(p,q) &= \tfrac{1}{N}(\hat{P}^y_{b,N}(p,q))^{-1} + M_{N,2}(p,q) \\ &= \tfrac{1}{N}\sum_{k=1}^{N} \phi_{b,k-1}(p,q)\phi^T_{b,k-1}(p,q) + \tfrac{1}{Nr_0} I + M_{N,2}(p,q) \end{aligned} \tag{4.18}$$

$$\begin{aligned}\tilde{R}^3_{\mathrm{N}}(p,q) &= \tfrac{1}{N}(\hat{P}^{\mathrm{u}}_{\mathrm{f,N}}(p,q))^{-1} + M_{\mathrm{N},3}(p,q)\\ &= \tfrac{1}{N}\sum_{k=1}^{N}\psi_{\mathrm{f,k-1}}(p,q)\psi^{\mathrm{T}}_{\mathrm{f,k-1}}(p,q) + \tfrac{1}{Nr_0}I + M_{\mathrm{N},3}(p,q)\end{aligned} \tag{4.19}$$

$$\begin{aligned}\tilde{R}^4_{\mathrm{N}}(p,q) &= \tfrac{1}{N}(\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}(p,q))^{-1} + M_{\mathrm{N},4}(p,q)\\ &= \tfrac{1}{N}\sum_{k=1}^{N}\psi_{\mathrm{b,k-1}}(p,q)\psi^{\mathrm{T}}_{\mathrm{b,k-1}}(p,q) + \tfrac{1}{Nr_0}I + M_{\mathrm{N},4}(p,q)\end{aligned} \tag{4.20}$$

where

$$\begin{aligned}M_{\mathrm{N},1}(p,q) &= \tfrac{1}{N}\sum_{i=0}^{\max(p,q)-1} f_{\mathrm{N},1}(i,p,q)f^{\mathrm{T}}_{\mathrm{N},1}(i,p,q)\\ M_{\mathrm{N},2}(p,q) &= \tfrac{1}{N}\sum_{i=0}^{\max(p,q)-2} f_{\mathrm{N},2}(i,p,q)f^{\mathrm{T}}_{\mathrm{N},2}(i,p,q)\\ M_{\mathrm{N},3}(p,q) &= \tfrac{1}{N}\sum_{i=0}^{\max(p,q)-1} f_{\mathrm{N},3}(i,p,q)f^{\mathrm{T}}_{\mathrm{N},3}(i,p,q)\\ M_{\mathrm{N},4}(p,q) &= \tfrac{1}{N}\sum_{i=0}^{\max(p-1,q-2)} f_{\mathrm{N},4}(i,p,q)f^{\mathrm{T}}_{\mathrm{N},4}(i,p,q)\end{aligned} \tag{4.21}$$

∎

Lemma 4.2 The perturbation vectors occurring in the modified normal equations (4.11) cause the parameters of TORA estimators to deviate from those of LS estimators provided system output and input measurements are one-sided sequences. This is shown by the following facts:

$$\begin{aligned}\tilde{\theta}^{\mathrm{T}}_{\mathrm{f,N}}(p,q)\tilde{R}^1_{\mathrm{N}}(p,q) &= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k}}\phi^{\mathrm{T}}_{\mathrm{f,k-1}}(p,q)\\ \tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)\tilde{R}^2_{\mathrm{N}}(p,q) &= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k-p}}\phi^{\mathrm{T}}_{\mathrm{b,k-1}}(p,q) - \tfrac{1}{N}\sum_{i=0}^{p-1}\mathbf{y}_{\mathrm{N-(p-1)+i}}f^{\mathrm{T}}_{\mathrm{N},2}(i,p,q)\\ \tilde{\chi}^{\mathrm{T}}_{\mathrm{f,N}}(p,q)\tilde{R}^3_{\mathrm{N}}(p,q) &= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{u}_{\mathrm{k}}\psi^{\mathrm{T}}_{\mathrm{f,k-1}}(p,q)\\ \tilde{\chi}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)\tilde{R}^4_{\mathrm{N}}(p,q) &= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{u}_{\mathrm{k-q}}\psi^{\mathrm{T}}_{\mathrm{b,k-1}}(p,q) - \tfrac{1}{N}\sum_{i=0}^{q-1}\mathbf{u}_{\mathrm{N-(q-1)+i}}f^{\mathrm{T}}_{\mathrm{N},4}(i,p,q),\end{aligned} \tag{4.22}$$

where $\tilde{R}^1_{\mathrm{N}}(p,q)$, $\tilde{R}^2_{\mathrm{N}}(p,q)$, $\tilde{R}^3_{\mathrm{N}}(p,q)$, and $\tilde{R}^4_{\mathrm{N}}(p,q)$ are presented in (4.17) -(4.20).

∎

Lemma 4.3 Suppose that system output and input measurements are one-sided sequences. Then, the right sides of the modified normal equations in (4.11) can be expressed in the following way so that the effect of the perturbation vectors on the right sides can be explicitly seen:

$$
\begin{aligned}
&(\tilde{R}_N^{yy}(1) \cdots \tilde{R}_N^{yy}(p)\ \tilde{R}_N^{yu}(0) \cdots \tilde{R}_N^{yu}(q)) = \tfrac{1}{N}\sum_{k=1}^{N} \mathbf{y}_k \phi_{f,k-1}^T(p,q) \\
&(\tilde{R}_N^{yy}(-p) \cdots \tilde{R}_N^{yy}(-1)\ \tilde{R}_N^{yu}(-p) \cdots \tilde{R}_N^{yu}(-p+q-1)) \\
&= \tfrac{1}{N}\sum_{k=1}^{N} \mathbf{y}_{k-p} \phi_{b,k-1}^T(p,q) + \tfrac{1}{N}\sum_{i=0}^{p-1} \mathbf{y}_{N-(p-1)+i} f_{N,2}^T(i,p,q) \\
&(\tilde{R}_N^{uy}(1) \cdots \tilde{R}_N^{uy}(p)\ \tilde{R}_N^{uu}(1) \cdots \tilde{R}_N^{uu}(q)) = \tfrac{1}{N}\sum_{k=1}^{N} \mathbf{u}_k \psi_{f,k-1}^T(p,q) \\
&(\tilde{R}_N^{uy}(-q) \cdots \tilde{R}_N^{uy}(-q+p)\ \tilde{R}_N^{uu}(-q) \cdots \tilde{R}_N^{uu}(-1)) \\
&= \tfrac{1}{N}\sum_{k=1}^{N} \mathbf{u}_{k-q} \psi_{b,k-1}^T(p,q) + \tfrac{1}{N}\sum_{i=0}^{q-1} \mathbf{u}_{N-(q-1)+i} f_{N,4}^T(i,p,q).
\end{aligned}
\tag{4.23}
$$

■

It follows from applying the generalized formula for the matrix inverse to (4.17) –(4.20) that

$$
\begin{aligned}
[\tilde{R}_N^1]^{-1} &= N\hat{P}_{f,N}^y - N\hat{P}_{f,N}^y M_{N,1}\{N\hat{P}_{f,N}^y M_{N,1} + I\}^{-1}\hat{P}_{f,N}^y \\
[\tilde{R}_N^2]^{-1} &= N\hat{P}_{b,N}^y - N\hat{P}_{b,N}^y M_{N,2}\{N\hat{P}_{b,N}^y M_{N,2} + I\}^{-1}\hat{P}_{b,N}^y \\
[\tilde{R}_N^3]^{-1} &= N\hat{P}_{f,N}^u - N\hat{P}_{f,N}^u M_{N,3}\{N\hat{P}_{f,N}^u M_{N,3} + I\}^{-1}\hat{P}_{f,N}^u \\
[\tilde{R}_N^4]^{-1} &= N\hat{P}_{b,N}^u - N\hat{P}_{b,N}^u M_{N,4}\{N\hat{P}_{b,N}^u M_{N,4} + I\}^{-1}\hat{P}_{b,N}^u .
\end{aligned}
\tag{4.24}
$$

Substituting the above four identities into (4.22) and then using (4.15) and (4.23) yield the following relation between the TORA and RLS estimates.

Theorem 4.1 Under the assumption that system output and input measurements are one-sided sequences, the TORA estimates are explicitly related to the RLS parameter estimates in the following manner:

$$
\tilde{\theta}_{f,N}^T = \hat{\theta}_{f,N}^T - \frac{1}{r_0}\hat{\theta}_{f,0}^T\hat{P}_{f,N}^y + (\frac{1}{r_0}\hat{\theta}_{f,0}^T\hat{P}_{f,N}^y - \hat{\theta}_{f,N}^T)M_{N,1}\{N\hat{P}_{f,N}^y M_{N,1} + I\}^{-1}\hat{P}_{f,N}^y \tag{4.25}
$$

$$
\begin{aligned}
\tilde{\theta}_{b,N}^T = \hat{\theta}_{b,N}^T - \tfrac{1}{r_0}\hat{\theta}_{b,0}^T\hat{P}_{b,N}^y \quad & -[\sum_{i=0}^{p-1} \mathbf{y}_{N-(p-1)+i} f_{N,2}^T(i)]\tfrac{1}{N}[\tilde{R}_N^2]^{-1} \\
& +(\tfrac{1}{r_0}\hat{\theta}_{b,0}^T\hat{P}_{b,N}^y - \hat{\theta}_{b,N}^T)M_{N,2}\{N\hat{P}_{b,N}^y M_{N,2} + I\}^{-1}\hat{P}_{b,N}^y ,
\end{aligned}
\tag{4.26}
$$

$$\tilde{\chi}^{\mathrm{T}}_{\mathrm{f,N}} = \hat{\chi}^{\mathrm{T}}_{\mathrm{f,N}} - \frac{1}{r_0}\hat{\chi}^{\mathrm{T}}_{\mathrm{f,0}}\hat{P}^{\mathrm{u}}_{\mathrm{f,N}} + (\frac{1}{r_0}\hat{\chi}^{\mathrm{T}}_{\mathrm{f,0}}\hat{P}^{\mathrm{u}}_{\mathrm{f,N}} - \hat{\chi}^{\mathrm{T}}_{\mathrm{f,N}})M_{\mathrm{N,3}}\{N\hat{P}^{\mathrm{u}}_{\mathrm{f,N}}M_{\mathrm{N,3}} + I\}^{-1}\hat{P}^{\mathrm{u}}_{\mathrm{f,N}} \quad (4.27)$$

$$\begin{aligned}\tilde{\chi}^{\mathrm{T}}_{\mathrm{b,N}} = \hat{\chi}^{\mathrm{T}}_{\mathrm{b,N}} - \tfrac{1}{r_0}\hat{\chi}^{\mathrm{T}}_{\mathrm{b,0}}\hat{P}^{\mathrm{u}}_{\mathrm{b,N}} \quad & -[\sum_{i=0}^{q-1} \mathbf{u}_{\mathrm{N-(q-1)+i}} f^{\mathrm{T}}_{\mathrm{N,4}}(i)]\tfrac{1}{N}[\tilde{R}^{4}_{\mathrm{N}}]^{-1} \\ & +(\tfrac{1}{r_0}\hat{\chi}^{\mathrm{T}}_{\mathrm{b,0}}\hat{P}^{\mathrm{u}}_{\mathrm{b,N}} - \hat{\chi}^{\mathrm{T}}_{\mathrm{b,N}})M_{\mathrm{N,4}}\{N\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}M_{\mathrm{N,4}} + I\}^{-1}\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}.\end{aligned} \quad (4.28)$$

∎

4.3 Time and Order Recursive Algorithm

Note from Eqs.(4.12), (4.16), and (4.20) that for any finite time N, all the matrices $\tilde{\Gamma}_N(p,q), (p,q) \in \mathcal{O}^{p^*,q^*}_{0,-1}$, are positive definite. This fact plus Theorem 3.1 implies the following conclusion.

Theorem 4.2 Suppose that system output and input measurements are one-sided sequences. For any finite time instant $N \geq 1$, the block-Toeplitz submatrix systems in (4.6) and (4.6) have the following properties:

(i) the matrices
$$\begin{pmatrix} \tilde{R}^b(p,q) & \tilde{\delta}(p,q) \\ \tilde{\delta}^T(p,q) & \tilde{V}^f(p,q) \end{pmatrix}, \begin{pmatrix} \tilde{V}^f(p,q) & \tilde{\kappa}(p,q) \\ \tilde{\kappa}^T(p,q) & \tilde{V}^b(p-1,q) \end{pmatrix}, \begin{pmatrix} \tilde{R}^f(p,q) & 0 \\ \tilde{\eta}(p,q) & \tilde{V}^f(p,q) \end{pmatrix}$$
are nonsingular.

(ii) the matrices$\tilde{R}^1(p,q)$, $\tilde{R}^2(p,q)$, $\tilde{R}^3(p,q)$, and $\tilde{R}^4(p,q)$ are nonsingular.

(iii) all the symmetric matrices in (i) and (ii) are positive definite.[2] ∎

This indicates that for each finite instant N, equations (4.6) and (4.6) can be solved by using the fast order-recursive algorithm (ORA). The remaining problem is how to update the block elements of the matrix $\tilde{\Gamma}_N(p,q)$ involved in (4.6).

[2]This implies that the matrices $\tilde{R}^f(p,q)$, $\tilde{R}^b(p,q)$, $\tilde{R}^f(p,q)$, $\tilde{R}^b(p,q)$, are positive definite.

From Eq.(4.7), the blocks of matrices $\tilde{\Gamma}_{\mathrm{N}}(p,q)$ can be updated in a time-recursive manner:

$$\begin{aligned}
\tilde{R}_{\mathrm{N}}^{yy}(n) &= \tfrac{N-1}{N}\tilde{R}_{\mathrm{N\text{-}1}}^{yy}(n) + \tfrac{1}{N}\mathbf{y}_{\mathrm{N}}\mathbf{y}_{\mathrm{N\text{-}n}}^{\mathrm{T}} + \delta(n)\delta(N-1)\tfrac{1}{r_0}I, && n=0,\dots,p^*,\\
\tilde{R}_{\mathrm{N}}^{yu}(n) &= \tfrac{N-1}{N}\tilde{R}_{\mathrm{N\text{-}1}}^{yu}(n) + \tfrac{1}{N}\mathbf{y}_{\mathrm{N}}\mathbf{u}_{\mathrm{N\text{-}n}}^{\mathrm{T}}, && n=0,\dots,q^*,\\
\tilde{R}_{\mathrm{N}}^{yu}(-n) &= \tfrac{N-1}{N}\tilde{R}_{\mathrm{N\text{-}1}}^{yu}(-n) + \tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}n}}\mathbf{u}_{\mathrm{N}}^{\mathrm{T}}, && n=p^*,\dots,1,\\
\tilde{R}_{\mathrm{N}}^{uu}(n) &= \tfrac{N-1}{N}\tilde{R}_{\mathrm{N\text{-}1}}^{uu}(n) + \tfrac{1}{N}\mathbf{u}_{\mathrm{N}}\mathbf{u}_{\mathrm{N\text{-}n}}^{\mathrm{T}} + \delta(n)\delta(N-1)\tfrac{1}{r_0}I, && n=0,\dots,q^*,
\end{aligned} \tag{4.29}$$

where $\tilde{R}_0^{yy}(n), \tilde{R}_0^{yu}(n), \tilde{R}_0^{yu}(-n), \tilde{R}_0^{yu}(n)$ are zero matrices of appropriate dimensions. So, the time-recursion can be written in the following way:

Procedure time-recursion $(N, \{\mathbf{y}_n\}_{n=\mathrm{N}}^{\mathrm{N}-p^*}, \{\mathbf{u}_n\}_{n=\mathrm{N}}^{\mathrm{N}-q^*})$

Data $\tilde{R}_{\mathrm{N\text{-}1}}^{yy}(\cdot), \tilde{R}_{\mathrm{N\text{-}1}}^{yu}(\cdot), \tilde{R}_{\mathrm{N\text{-}1}}^{uu}(\cdot)$

begin

update $\tilde{R}_{\mathrm{N}}^{yy}(\cdot), \tilde{R}_{\mathrm{N}}^{yu}(\cdot), \tilde{R}_{\mathrm{N}}^{uu}(\cdot)$ via (4.29)

Solve Eq.(4.6) or Eq. (4.6) of order (p_0, q_0)[3]

end

Time and Order Recursive Algorithm

Combining the time recursion with the ORA, we have the TORA for solving all block-Toeplitz submatrix systems of order $(p,q) \in \mathcal{O}_{p_0,q_0}^{p^*,q^*}$.

Algorithm TORA

Data p_0, q_0, p^*, q^*, N_f

start

for $N := 1$ up to N_f

begin

call time-recursion $(N, \{\mathbf{y}_n\}_{n=\mathrm{N}}^{\mathrm{N}-p^*}, \{\mathbf{u}_n\}_{n=\mathrm{N}}^{\mathrm{N}-q^*})$

call the ORA to determine all the TORA estimates (described in (4.13)) of order $(p,q) \in \mathcal{O}_{p_0,q_0}^{p^*,q^*}$

end

[3]This represents the calculation of the initial condition of the ORA in p and ORA in q.

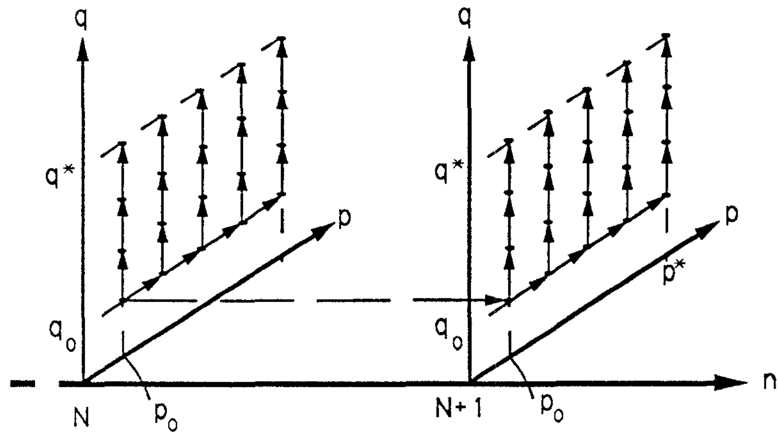


Figure 4.1: Illustration of the time and order recursions in the TORA

end

stop

The TORA is also illustrated in Figure 4.1. It can be seen there that the long-dashed line represents the recursive calculation of the initial condition for the TORA, which is done right after the new data are received. Then, all the TORA estimates of all order $(p, q) \in \mathcal{O}^{p^*,q^*}_{p_0,q_0}$ are determined by the ORA.

Remark 4.4 Mathematically, the TORA generates the solution of Eq.(4.6). As will be shown, the solution is a good approximation of the exact solution to problem (4.2). ∎

Remark 4.5 The TORA generates parameter estimates for all input/output forward/backward predictors. Thus, the inputs and outputs of the ARX models receive symmetric treatments in the TORA, which is useful in econometrics and in control when actuators have some imperfections [104]. ∎

The Computational Complexity of the TORA

It can see from (4.29) that the cost of updating the block elements of the matrix $\tilde{\Gamma}_N(p^*, q^*)$ is $\mathbf{O}(mp^* + lq^*)$ multiplications and additions. Note that the TORA consists of the time-recursion and the ORA. The computational complexity of the whole TORA can be found easily from Theorem 3.4 and the description of the time-recursion.

Theorem 4.3 The TORA can efficiently compute all solutions to Eq.(4.6) of order $(p,q) \in \mathcal{O}^{p^*,q^*}_{p_0,q_0}$. For each time iteration, it just takes $\mathbf{O}((q^*p^{*^2}) \vee (p^*q^{*^2}))$ flops, which are contributed by

$$\textbf{Time-recursion: } \tfrac{1}{6}(mp_0 + lq_0 + l)^3 + \tfrac{1}{3}(mp_0 + lq_0)^3 + \tfrac{1}{6}(mp_0 + m + lq_0)^3 + \mathbf{O}(p^* + q^* + (mp_0 + lq_0)^2)$$

$$\textbf{ORA: } \mathbf{O}((p^{*^2}q^*) \vee (p^*q^{*^2}))$$

Moreover, the unparallelizable computational task takes only $\mathbf{O}(mp^* + lq^*)$ multiplication operations and $\mathbf{O}((mp^* + lq^*)\log_2(mp^* + lq^*))$ addition operations. ∎

Remark 4.6 For problem (4.2), if the RLS is used, then the number of required multiplication-addition operations could be as high as $\mathbf{O}((q^*p^{*^3}) \vee (p^*q^{*^3}))$. So, the TORA significantly reduces the computational complexity of solving *a set* of LS parameter estimation problems of order $(p,q) \in \mathcal{O}^{p^*,q^*}_{p_0,q_0}$, particularly in the case that p^* or q^* is large. ∎

Remark 4.7 From the viewpoint of computation, the initial order (p_0, q_0) should be chosen to be as small as possible. For the TORA illustrated in Figure 2, the initial order of $(0, -1)$ is suggested. ∎

4.4 Convergence Analysis

The TORA is more efficient computationally than the RLS. But it just provides an approximation of the RLS parameter estimates. Notice that the asymptotic properties of the RLS have been thoroughly investigated [21]. We devote our main efforts in this section to studying the relation between the RLS estimates and the TORA estimates. In Section 4.2, we have expressed the RLS estimates and the TORA estimates in a form useful in the study of the asymptotics of the TORA. We have also established the expression for the difference between the parameter estimates generated by the two algorithms, where the key factors of the perturbation vectors are specified explicitly. Now we will show that the TORA preserves the consistency properties of the RLS. Since the argument (p, q) is not important in this section, it will be omitted except when it is needed.

Before expressing the main results, we present the following fact [73]: Let A_N, $N = 1, 2, \cdots$, be a sequence of invertible matrices. If $\lim_{N\to\infty} A_N = I$, then $\lim_{N\to\infty} A_N^{-1} = I$.

Theorem 4.4 Consider the forward/backward output/input MAX predictors of order (p, q) described in (4.5)-(4.10). Let $\{\mathbf{y}_N\}_{N=1}^{\infty}$ and $\{\mathbf{u}_N\}_{N=1}^{\infty}$ be two uniformly bounded output and input sequences. For each of the four kinds of estimators in (4.5) -(4.10), say the output predictor in (4.5), if for order $(p, q) \in \mathcal{O}_{p_0,q_0}^{p^*,q^*}$,

1. the gain matrix satisfies: $\|\hat{P}_{f,N}^{y}\| \to 0$, as $N \to \infty$,
2. the RLS estimates of the output predictor are uniformly bounded:

$$\|\hat{\theta}_{f,N}\| < K < \infty \quad \text{for } \forall N \geq 1,$$

then the difference between the RLS estimates and the TORA estimates has the following relationship:

$$\|\hat{\theta}_{f,N}(p,q) - \tilde{\theta}_{f,N}\| \leq \mathbf{O}(\|\hat{P}_{f,N}^{y}\|). \tag{4.30}$$

∎

Corollary 1: If the assumptions in Theorem 4.4 hold almost surely and the parameter estimates produced by the RLS are strongly consistent, then the TORA generates strongly consistent parameter estimates. ∎

Proof: Applying the fact mentioned just before Theorem 4.4 to (4.24) yields:

$$\begin{aligned} \|\tfrac{1}{N}[\tilde{R}^2_{\mathrm{N}}]^{-1}\| &\leq \|\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}\| + \mathbf{O}(\|\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}\|^2) \\ \|\tfrac{1}{N}[\tilde{R}^4_{\mathrm{N}}]^{-1}\| &\leq \|\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}\| + \mathbf{O}(\|\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}\|^2) \end{aligned} \tag{4.31}$$

since $\{N\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}M_{\mathrm{N,2}} + I\}$ and $\{N\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}M_{\mathrm{N,4}} + I\}$ converge to identity matrices as N goes to infinity. Therefore, by assumption (2) and Theorem 4.1, we have the conclusion from (4.25)-(4.28)

$$\begin{aligned} &\|\tilde{\theta}_{\mathrm{f,N}} - \hat{\theta}_{\mathrm{f,N}}\| = \mathbf{O}(\|\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}\|) \\ &\|\tilde{\theta}_{\mathrm{b,N}} - \hat{\theta}_{\mathrm{b,N}}\| = \mathbf{O}(\|\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}\|) + \mathbf{O}(\tfrac{1}{N}\|[\tilde{R}^2_{\mathrm{N}}]^{-1}\|) = \mathbf{O}(\|\hat{P}^{\mathrm{y}}_{\mathrm{b,N}}\|) \\ &\|\tilde{\chi}_{\mathrm{f,N}} - \hat{\chi}_{\mathrm{f,N}}\| = \mathbf{O}(\|\hat{P}^{\mathrm{u}}_{\mathrm{f,N}}\|) \\ &\|\tilde{\chi}_{\mathrm{b,N}} - \hat{\chi}_{\mathrm{b,N}}\| = \mathbf{O}(\|\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}\|) + \mathbf{O}(\tfrac{1}{N}\|[\tilde{R}^4_{\mathrm{N}}]^{-1}\|) = \mathbf{O}(\|\hat{P}^{\mathrm{u}}_{\mathrm{b,N}}\|) \end{aligned} \tag{4.32}$$

Corollary 1 comes from the triangle inequality of matrix norms and (4.32).

∎

The on-line parameter estimation problem for feedback control systems is a problem of long-standing interest. An important result on this problem is from the paper by Lai and Wei[92]. Following the presentation of Caines[21], we give the result below for future reference.

Theorem 4.5 [Lai and Wei, [92]] Let the processes $\mathbf{z} = \begin{pmatrix} \mathbf{y} \\ \mathbf{u} \end{pmatrix}$ and $\mathbf{e}$ be related by the following ARX system:

$$\begin{aligned} \mathbf{y}_n &= -A_1\mathbf{y}_{n-1} - \cdots - A_p\mathbf{y}_{n-p} + C_0\mathbf{u}_n + \cdots + C_q\mathbf{u}_{n-q} + \mathbf{e}_n \\ &= -\theta^T\phi_{f,n-1} + \mathbf{e}_n \end{aligned}$$

where

$$\theta^T \triangleq (A_1 \cdots A_p \; - C_0 \cdots - C_q)$$

and

$$\phi^T_{f,n-1} = (\mathbf{y}^T_{n-1} \cdots \mathbf{y}^T_{n-p}\mathbf{u}^T_n \cdots \mathbf{u}^T_{n-q}).$$

Assume that

(C1) $\mathbf{e}$ is a martingale difference process with respect to an increasing family of σ-fields $\mathcal{F}_k; k \in Z_1$.

(H1) $\sup_{k\in Z_1} \mathrm{E}\|\mathbf{e}_k\|^\alpha|\mathcal{F}_{k-1}) < \infty$ a.s. for some $\alpha > 2$.

Denote by $\lambda^y_{min}(N)$ and $\lambda^y_{max}(N)$ the minimum and maximum eigenvalues, respectively, of $\sum_{k=1}^N \phi_{f,k-1}\phi^T_{f,k-1}$ and assume that

(H2) $\lambda^y_{min}(N) \to \infty$ a.s. and $\log \lambda^y_{max}(N) = \mathbf{o}(\lambda^y_{min}(N))$ *a.s.*

Then the least squares estimate $\hat{\theta}_N$ is a strongly consistent estimate of θ and, further,

$$\|\hat{\theta}_N - \theta\| = \mathbf{o}((\frac{\log(\lambda^y_{max}(N))}{\lambda^y_{min}(N)})^{1/2}) \quad a.s.$$

∎

Theorem 4.6 Suppose that the assumptions of Theorem 4.5 hold and the true order $(p,q) \in \mathcal{O}^{p^*,q^*}_{p_0,q_0}$. If the processes $\mathbf{y}$ and $\mathbf{u}$ are uniformly bounded for each realization, the TORA generates a strongly consistent estimate of θ. Further

$$\|\tilde{\theta}_N - \theta\| = \mathbf{o}((\frac{\log(\lambda^y_{max}(N))}{\lambda^y_{min}(N)})^{1/2}) \quad a.s.. \tag{4.33}$$

∎

Proof: It follows from (4.16) that for large N,

$$\|\hat{P}^y_{f,N}\| \cong (\lambda^y_{min}(N))^{-1} \quad a.s.. \tag{4.34}$$

Therefore, under the assumptions of Theorem 4.5 and boundedness of the input and output processes, (4.32) holds almost surely. That implies

$$\|\tilde{\theta}_N - \theta\| \leq \|\hat{\theta}_N - \theta\| + \mathbf{O}(\|\hat{P}^y_{f,N}\|) \quad a.s..$$

Using Theorem 4.4 and (4.34), we have

$$\begin{aligned}\|\tilde{\theta}_N - \theta\| &= \mathbf{o}((\frac{\log(\lambda^y_{max}(N))}{\lambda^y_{min}(N)})^{1/2}) + \mathbf{O}(\frac{1}{\lambda^y_{min}(N)}) \quad a.s.\\ &= \mathbf{o}((\frac{\log(\lambda^y_{max}(N))}{\lambda^y_{min}(N)})^{1/2}) \quad a.s..\end{aligned}$$

∎

Theorem 4.5 means that the TORA preserves the convergence property of the RLS provided the input and output processes are almost surely and uniformly bounded. In other words, we can use the TORA to estimate the parameters of ARX systems without losing the convergence property of the RLS.

Remark 4.8 In Lemmas 4.1 – 4.3 and Theorems 4.1 and 4.2, we assume that the system output and input measurements are one-sided. That is, $\mathbf{y}_{-k} = 0$ if $0 \leq k \leq p^*$, and $\mathbf{u}_{-k} = 0$ if $0 \leq k \leq q^*$. It can be seen from the proofs of the above conclusions in Appendix B that there is no problem essentially if this assumption is not true. To preserve Theorem 4.2 with this assumption, a small enough constant r_0 should be chosen in (4.7)–(4.13) instead of an arbitrary positive number. The other conclusions hold, with the difference that the matrices $M_{N,i}, i = 1, 2, 3, 4$, have much more expression. But they still converge to zero as N goes to infinity if the measurements are uniformly bounded. As a result, all the conclusions in Section 4.4 are true no matter whether the one-sided assumption is made.

4.5 Error Analysis and Compensation

The TORA estimates converge to the desired LS parameter estimates under some mild conditions. Section 4.4 provides us with an upper bound on the convergence rate of the TORA estimates to the corresponding LS estimates when the number of data points is large. This is not good enough in certain cases because the transient performance of parameter estimation is also important in practice and the unspecified proportionality coefficient in the bound described in (4.30) and (4.33) could affect the convergence rate significantly for a while. As derived in Section 4.2.3, the difference between the TORA and LS parameter estimates is

$$\tilde{\theta}^{T}_{f,N} - \hat{\theta}^{T}_{f,N} = -\frac{1}{r_0}\hat{\theta}^{T}_{f,0}\hat{P}^{y}_{f,N} + (\frac{1}{r_0}\hat{\theta}^{T}_{f,0}\hat{P}^{y}_{f,N} - \hat{\theta}^{T}_{f,N})M_{N,1}\{N\hat{P}^{y}_{f,N}M_{N,1} + I\}^{-1}\hat{P}^{y}_{f,N},$$

$$\begin{aligned}\tilde{\theta}^{T}_{b,N} - \hat{\theta}^{T}_{b,N} = -\frac{1}{r_0}\hat{\theta}^{T}_{b,0}\hat{P}^{y}_{b,N} &-[\sum_{i=0}^{p-1} \mathbf{y}_{N-(p-1)+i}f^{T}_{N,2}(i)]\frac{1}{N}[\tilde{R}^{2}_{N}]^{-1} \\ &+(\frac{1}{r_0}\hat{\theta}^{T}_{b,0}\hat{P}^{y}_{b,N} - \hat{\theta}^{T}_{b,N})M_{N,2}\{N\hat{P}^{y}_{b,N}M_{N,2} + I\}^{-1}\hat{P}^{y}_{b,N},\end{aligned}$$

$$\tilde{\chi}^{T}_{f,N} - \hat{\chi}^{T}_{f,N} = -\frac{1}{r_0}\hat{\chi}^{T}_{f,0}\hat{P}^{u}_{f,N} + (\frac{1}{r_0}\hat{\chi}^{T}_{f,0}\hat{P}^{u}_{f,N} - \hat{\chi}^{T}_{f,N})M_{N,3}\{N\hat{P}^{u}_{f,N}M_{N,3} + I\}^{-1}\hat{P}^{u}_{f,N},$$

$$\begin{aligned}\tilde{\chi}^{T}_{b,N} - \hat{\chi}^{T}_{b,N} = -\frac{1}{r_0}\hat{\chi}^{T}_{b,0}\hat{P}^{u}_{b,N} &-[\sum_{i=0}^{q-1} \mathbf{u}_{N-(q-1)+i}f^{T}_{N,4}(i)]\frac{1}{N}[\tilde{R}^{4}_{N}]^{-1} \\ &+(\frac{1}{r_0}\hat{\chi}^{T}_{b,0}\hat{P}^{u}_{b,N} - \hat{\chi}^{T}_{b,N})M_{N,4}\{N\hat{P}^{u}_{b,N}M_{N,4} + I\}^{-1}\hat{P}^{u}_{b,N}.\end{aligned}$$

To be easy to get some insight into how to conduct error compensation, we only consider here the TORA and LS output predictor and assume zero initial condition, i.e., $\hat{\theta}^{T}_{f,0} = 0$. Thus, the difference between the parameters of the TORA and LS output predictor has the simple description

$$\tilde{\theta}^{T}_{f,N} - \hat{\theta}^{T}_{f,N} = \underbrace{-\hat{\theta}^{T}_{f,N}M_{N,1}\{N\hat{P}^{y}_{f,N}M_{N,1} + I\}^{-1}\hat{P}^{y}_{f,N}}_{\text{error term}}. \tag{4.35}$$

Note that

$$M_{N,1} = \frac{1}{N}\sum_{i=0}^{\max(p,q)-1} f_{N,1}(i,p,q)f^{T}_{N,1}(i,p,q)$$

and $f^{T}_{N,1}(i,p,q)$ is composed from the latest $p+1$ data points of $\mathbf{y}$ and the latest q data points of $\mathbf{u}$. This indicates that the error could explode when the latest measurements become large suddenly. This is the case when the system to be identified is not well damped. Therefore, some error compensation might be required in such cases. For example, the error term on the left side of Eq. (4.35) could be partially cancelled.

Remark 4.9 To avoid some possible confusion, we emphasize that the error term for a particular order at a particular time instant has effect on neither the TORA estimates with higher order at that time instant nor the TORA estimates at the future time instant. In this sense, the error term does not propagate either in time or in order. ∎

There might be many ways to compensate the error term. Each of them has some sort of tradeoff between the desired performance of parameter estimates and the computational burden for accomplishing the compensation. Here, we suggest a method. Substituting (4.15) into (4.35) yields

$$\tilde{\theta}^{\mathrm{T}}_{\mathrm{f,N}} - \hat{\theta}^{\mathrm{T}}_{\mathrm{f,N}} = \Big(\sum_{k=1}^{N} \mathbf{y}_{\mathrm{k}} \phi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q)\Big)\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q) M_{\mathrm{N,1}}\{N\hat{P}^{\mathrm{y}}_{\mathrm{f,N}} M_{\mathrm{N,1}} + I\}^{-1}\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}. \tag{4.36}$$

Note from (4.23) and (4.17) that

$$\Big(\sum_{k=1}^{N} \mathbf{y}_{\mathrm{k}} \phi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q)\Big) = \begin{pmatrix} \tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(1) & \cdots & \tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(p) & \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(0) & \cdots & \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(q) \end{pmatrix} \tag{4.37}$$

and

$$\{N\hat{P}^{\mathrm{y}}_{\mathrm{f,N}} M_{\mathrm{N,1}} + I\}^{-1} = I - [\tilde{R}^{1}_{\mathrm{N}}]^{-1} M_{\mathrm{N,1}}. \tag{4.38}$$

So, the desired error compensation hinges on efficiently computing $\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)$ and $[\tilde{R}^{1}_{\mathrm{N}}]^{-1}$ for any $(p,q) \in \mathcal{O}^{p^*,q^*}_{p_0,q_0}$.

We may replace $\hat{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)$ and $[\tilde{R}^{1}_{\mathrm{N}}]^{-1}$ in (4.36) and (4.38) by $\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)$ and $\check{R}^{1}_{\mathrm{N}}$. The matrices $\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)$ and $\check{R}^{1}_{\mathrm{N}}$ are defined as

$$\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q) = \begin{pmatrix} \big(\sum_{k=1}^{N} \phi^{\mathrm{y}}_{\mathrm{f,k\text{-}1}}(p,q)\phi^{\mathrm{y}^{\mathrm{T}}}_{\mathrm{f,k\text{-}1}}(p,q) + \frac{1}{r_0}I\big)^{-1} & \mathbf{0} \\ \mathbf{0} & \big(\sum_{k=1}^{N} \phi^{\mathrm{u}}_{\mathrm{f,k\text{-}1}}(p,q)\phi^{\mathrm{u}^{\mathrm{T}}}_{\mathrm{f,k\text{-}1}}(p,q) + \frac{1}{r_0}I\big)^{-1} \end{pmatrix} \tag{4.39}$$

and

$$\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q) = \left(\begin{array}{ccc|ccc} R^{yy}_{\mathrm{N}}(0) & \cdots & R^{yy}_{\mathrm{N}}(p-1) & & & \\ \vdots & \ddots & \vdots & & \mathbf{0} & \\ R^{yy}_{\mathrm{N}}(1-p) & \cdots & R^{yy}_{\mathrm{N}}(0) & & & \\ & & & R^{uu}_{\mathrm{N}}(0) & \cdots & R^{uu}_{\mathrm{N}}(q) \\ & \mathbf{0} & & \vdots & \ddots & \vdots \\ & & & R^{uu}_{\mathrm{N}}(-q) & \cdots & R^{uu}_{\mathrm{N}}(0) \end{array}\right), \tag{4.40}$$

where

$$\phi^{\mathrm{y}}_{\mathrm{f,k\text{-}1}}(p,q) \triangleq \begin{pmatrix} \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}} & \cdots & \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}p}} \end{pmatrix}$$

and

$$\phi^{\mathrm{u}}_{\mathrm{f,k\text{-}1}}(p,q) \triangleq (\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N}}\ \ldots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}q}}\ ).$$

The resulting estimates, denoted by $\check{\theta}_{\mathrm{f,N}}$, are equal to

$$\begin{aligned}\check{\theta}_{\mathrm{f,N}} &= \tilde{\theta}^{\mathrm{T}}_{\mathrm{f,N}} \\ &-[(\tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(1)\cdots\tilde{R}^{\mathrm{yy}}_{\mathrm{N}}(p)\ \tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(0)\cdots\tilde{R}^{\mathrm{yu}}_{\mathrm{N}}(q))\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)M_{\mathrm{N,1}}\{I-[\tilde{R}^{1}_{\mathrm{N}}]^{-1}M_{\mathrm{N,1}}\}\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)]\end{aligned} \tag{4.41}$$

and they will be called the *modified TORA estimates.* The matrices $\check{P}^{\mathrm{y}}_{\mathrm{f,N}}(p,q)$ and $[\tilde{R}^{1}_{\mathrm{N}}]^{-1}$ in the compensation term of (4.41) can be efficiently calculated by using the algorithm of Friedlander et al. [47] and the Levinson-Durbin algorithm [138]. When these algorithms are used, the computational cost of determining the error compensation in (4.41) is of order of $p^{*^2} + q^{*^2}$. Consequently, the conclusion can be drawn from Theorem 4.3 that the modified TORA has about the same computational complexity as the TORA.

4.6 Numerical Simulations

A large number of simulations has been done. Here, we report a couple of them. A comprehensive report on system order and parameter estimation will be presented in the next chapter.

Example 4.1 Consider an ARX system:

$$y_n + 0.7 \cdot y_{n-1} - 0.4975 \cdot y_{n-2} - 0.8483 \cdot y_{n-3} - 1. \cdot u_n = w_n \tag{4.42}$$

where the input u_n is a sinusoidal signal with two frequencies, $u_n = sin(3n) + sin(0.1n)$. The model noise w_n is a zero mean pseudo white noise with a variance of 0.5. The parameter estimation of the ARX system has been done, respectively, by using the RLS, TORA, and over-parametrization method which is discussed in Section 4.1. All these algorithms are programmed using Pro-Matlab[109]. The system output is drawn in Subplot 1 of Figure 4.2, where the horizontal axis represents the iteration number. The LS parameter estimates are plotted in both Subplot 2 and Subplot 3 of Figure 4.2 as a reference for comparing the

LS estimates with the TORA estimates or with the estimates determined by the over-parametrization method, which will be denoted by the OP estimates. The OP estimates are drawn in Subplot 2 and the TORA estimates in Subplot 3. Subplot 2 shows that the OP estimates have a slow convergence rate. This is predictable in this example because without the model noise the output and input measurements are not persistently exciting signals for an ARX model of order $(3,2)$ which is implicitly used in the over-parametrization method [138]. This implies that the minimum eigenvalue of the normal matrix $\hat{P}^{y}_{f,N}(3,2)$ is uniformly bounded instead of approaching infinity. As shown in Theorem 4.5, the minimum eigenvalue is inversely proportional to the magnitude of the LS parameter estimation error. Therefore, the OP estimates converge to the true model parameters slowly because it is the pseudo white noise that drives the minimum eigenvalue to approach infinity. In contrast, the TORA can generate consistent parameter estimates even without the model noise. In conclusion, this example shows that OP estimates have larger variance of parameter estimation than the TORA estimates because the former requires some kind of over-parametrization.

Example 4.2 Consider an ARX system:

$$y_n + 2.4y_{n-1} + 1.91y_{n-2} + 0.504y_{n-3} = u_n + \omega_n \tag{4.43}$$

where ω_n is a pseudo white noise with variance of 0.5 and the model input u_n is also a pseudo white noise but has a variance of 1.5. As shown by Figure 4.3, the system is not well damped because the maximum magnitude of the transfer function of the system can reach more than 100. The parameter estimation of the ARX system has been done, respectively, by using the RLS, TORA, modified TORA and over-parametrization (OP) method. All these algorithms are programmed by using Pro-Matlab[109]. The system has three poles of -0.9, -0.8, and -0.7 and the output is plotted in Subplot 1 in Figure 4.4. The simulation results in this example show that the LS parameter estimation algorithm has its superority in terms of convergence rate over the other three algorithms. The superority is significant for the TORA and OP algorithms, but is slight compared

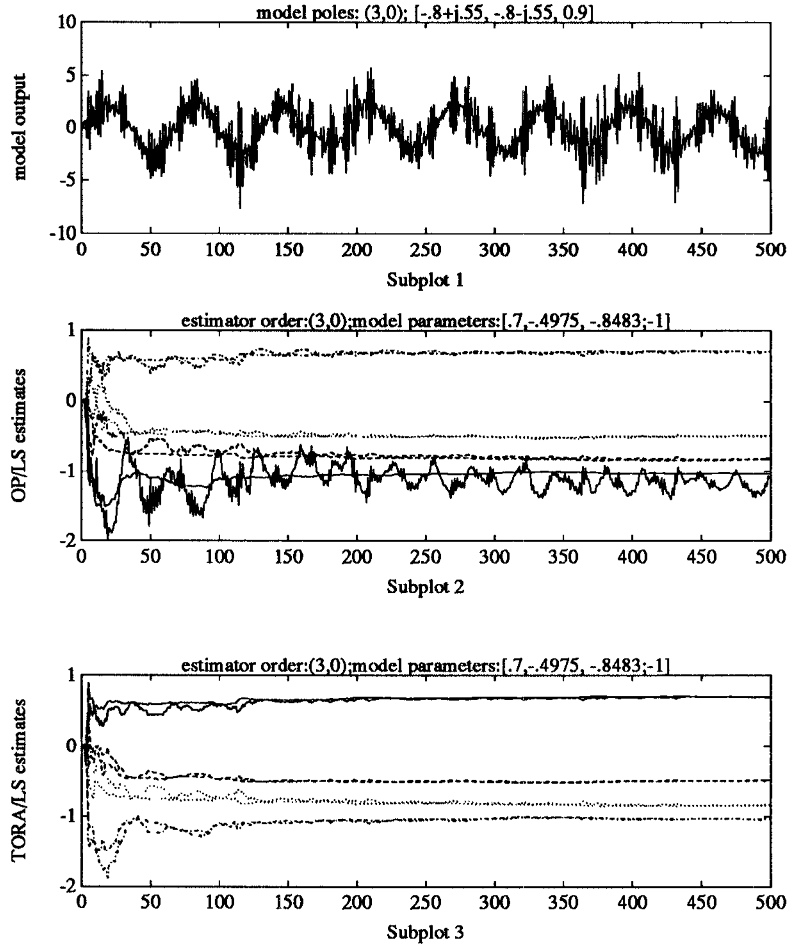


Figure 4.2: The parameter estimates via the TORA and over-parametrization method are compared with each other in Figures 4.2, where the much "jumpier" curves in Subplot 2 represent the parameter estimates obtained by using the *over-parametrization* method. The much "jumpier" curves in Subplot 3 represent the TORA estimates. The corresponding LS estimates are drawn in Subplot 2 and are plotted again in Subplot 3, which provides with a comparison of the OP or TORA estimates with the LS estimates. As shown in Subplots 2 and 3, the over-parametrization causes the parameter estimates to have much larger variance than the TORA parameter estimates.

with the modified TORA. As pointed out before, the deviation of the TORA estimates from the corresponding LS estimates is proportional to the squared model output and inversely proportional to the minimum eigenvalues of the normal matrices involved in the LS parameter estimation. Subplot 1 in Figure 4.4 shows that the model output changes dramatically. The magnitude of the model output varies from more than 100 to 5 within 50 iterations. Meanwhile, as shown in Subplot 2 in Figure 4.3, the associated minimum eigenvalues increase almost linearly with a small slope of about 0.54 as data points get more and more. Consequently, the small minimum eigenvalues cannot prevent the uprising of the model output in magnitude from degenerating the TORA and OP estimates, which is shown in Subplots 2 and 3 in Figure 4.4. As we expect, Subplots 2 and 3 in Figure 4.4 indicate that the OP parameter estimates are worse than the TORA estimates. This is because two extra parameters have to be introduced in this example to equalize the numbers of the feedback coefficients and feedforward parameters. The modified TORA shows its remarkable improvement on the TORA in Subplot 4 in Figure 4.4, especially when the iteration number is large. For instance, as shown in Subplot 4 in figure 4.4, only a very small deviation is caused by the jumping of the output magnitude to more than 60 when the iteration number is close to 500.

4.7 Summary

In this chapter, a fast time and order recursive algorithm (TORA) for computing a family of ARX models for given input-output data has been developed. We believe that the full exploitation of this technique could bring improvements in system identification algorithms. The TORA, including the modified TORA, can well approximate all LS parameter estimates for ARX models of different order with computational complexity significantly less than that for the RLS. Structurally, the algorithm has a good physical interpretation and its parallelism can be easily visualized. Computationally, it operates like a linear minimum mean

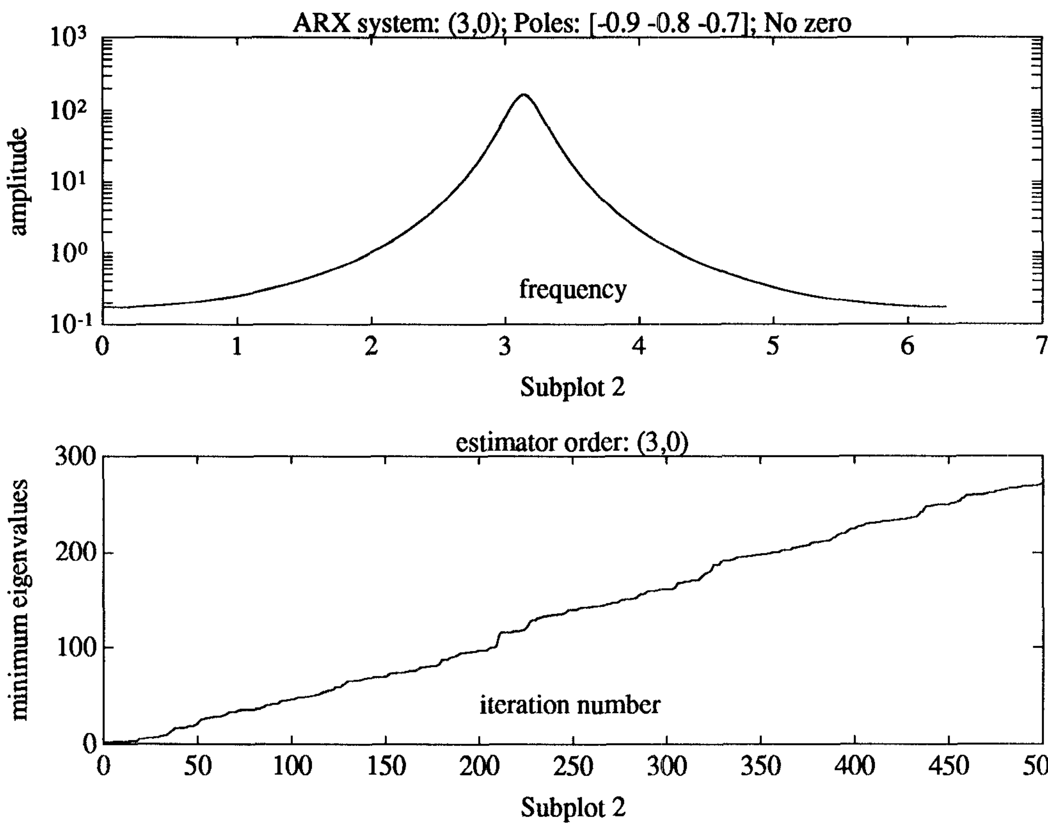


Figure 4.3: The amplitude of the transfer function of the system (4.45) is shown in Subplot 1, which indicates that the system is not well damped. For understanding the simulation results of parameter estimation, the minimum eigenvalues of the normal matrices involved are plotted in Subplot 2.

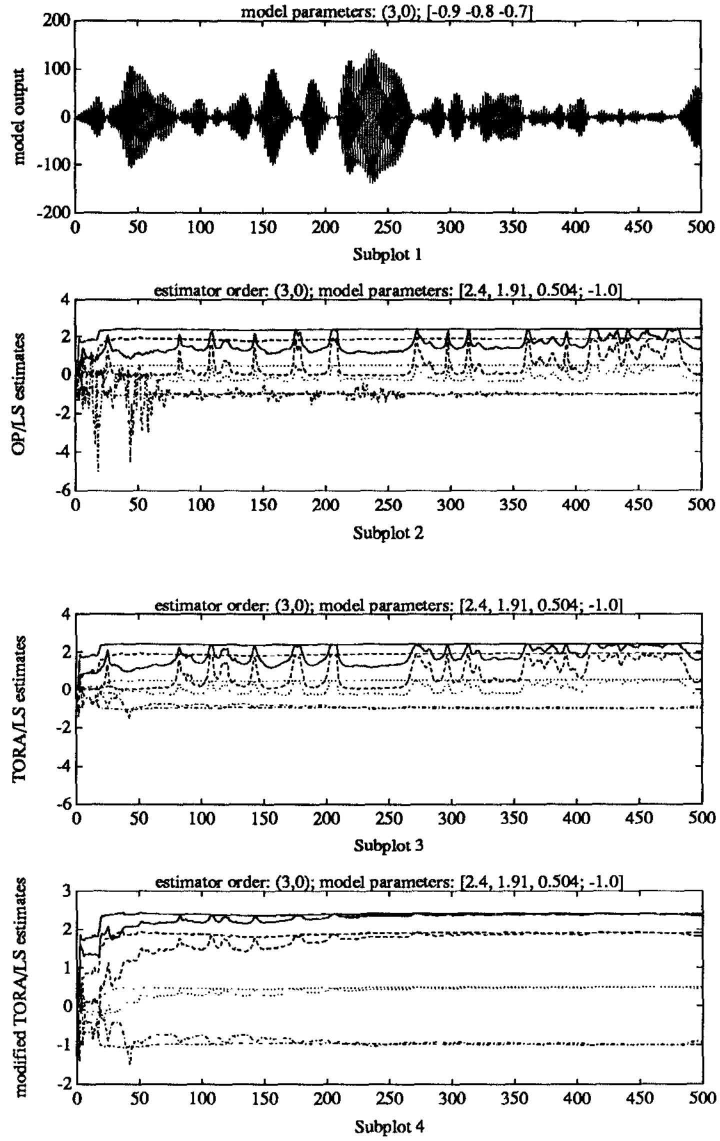


Figure 4.4: A comparison of LS, TORA, modified TORA, and OP parameter estimates are made here. In each of Subplots 2, 3, and 4, four pairs of curves are presented. The "jumpier" curve in each pair represents a TORA, modified TORA, or OP parameter estimate. The smoother curves are always LS parameter estimates. Another expresses an LS parameter estimate. Sudden bursts of model output push TORA and OP estimates to deviate from the desired LS estimates. This is worsened when the minimum eigenvalue is small. However, the modified TORA estimates converge to the LS estimates at a satisfactory rate. As we expect, the OP estimates have larger variance than the TORA parameter estimates.

square error parameter estimator with inputs of time-varying blocks instead of autocorrelation matrices and cross-correlation matrices. These blocks may be interpreted as estimates of the auto-correlation matrices and cross-correlation matrices if the input and output processes are jointly-stationary [53]. For uniformly bounded input and output processes, the strong consistency of the parameter estimates generated by the RLS guarantees that the TORA generates strongly consistent parameter estimates. The TORA can be used in modeling of autoregressive moving-average systems with exogenous inputs (ARMAX) simply by using the TORA to build an ARMA model of the residuals or by applying it to the three-stage LS identification algorithm [102]. The TORA can be useful in system identification, linear filtering, adaptive signal processing, and adaptive control by combining it with other relevant techniques such as those for order estimation. It could also be useful in data analysis of two-dimensional systems because the TORA does not require causality of input and output data.

The TORA itself is a fast least-squares modeling technique. Its performance cannot be better than that of least-squares estimators. There are some places within the TORA for further modification or extension. Some inner products are involved in computation of some intermediate variables and some Schur-type algorithms are needed to eliminate them. In addition, the condition number for the TORA is worse than that for the QR-based algorithms. So, for some applications, a QR version of the TORA would be desirable. We believe that the QR version of the TORA could be established by applying the QR decomposition to the matrix inversion occurring in computation of the partial correlation coefficients and using the techniques developed in reference [100].

Chapter 5

A Fast Method for Strongly Consistent Estimation of ARX System Order And Parameters

In this chapter a fast algorithm for strongly consistent order and parameter estimation is proposed for MIMO ARX systems with martingale difference noise processes. This algorithm proceeds by minimizing the accumulated prediction error over a family of ARX models of different order. The parameters of all ARX models within the family are obtained by using the fast time and order recursive algorithm developed in Chapter 4. The strong consistency of the proposed algorithm is proved by extending the results of Hemmerly and Davis[72]. This algorithm could become an efficient and practical tool for linear modeling based on input and output data.

5.1 Preliminaries

Consider two data sequences $\{\mathbf{y}_t\}_{t=0}^{n}$ and $\{\mathbf{u}_t\}_{t=0}^{n}$, which are realizations of zero-mean processes $\mathbf{y}$ and $\mathbf{u}$. The processes are related by an unknown discrete-time system with disturbances. A basic step in the analysis and use of the data is the determination of a model for the system. The most popular approach for determining a system model is linear modeling because many theories and tools are available. Linear modeling is concerned with the problem of determining a linear model of a dynamic system, i.e., estimating the model order and parameters, based on the observations. Good estimates of the model order are important in adaptive control system design and adaptive IIR filtering because both stability and performance of a closed-loop system are strongly affected by the number of poles and zeros of the system model. A model-searching approach has been suggested in Chapter 4. This approach is composed of two steps: (a) determining a family of least squares (LS) ARX models of different order (p, q). The LS ARX models are described as

$$\hat{A}_n(z^{-1})\mathbf{y}_n = \hat{C}_n(z^{-1})\mathbf{u}_n + \mathbf{e}_n \tag{5.1}$$

where $\mathbf{e}_n$ is the output prediction error. $\hat{A}_n(z^{-1}) = I + \hat{A}_{n,1}z^{-1} + \cdots + \hat{A}_{n,p}z^{-p}$ and $\hat{C}_n(z^{-1}) = \hat{C}_{n,0} + \hat{C}_{n,1}z^{-1} + \cdots + \hat{C}_{n,q}z^{-q}$ are matrix polynomials in the backward shift operator z^{-1}. Parameters of each LS ARX model within the family are determined by solving a minimization problem described below:

$$\min_{\hat{A}_{n,i},\hat{C}_{n,j}} \{\sum_{t=0}^{n} \|\hat{A}_n(z^{-1})\mathbf{y}_t - \hat{C}_n(z^{-1})\mathbf{u}_t\|^2\} \tag{5.2}$$

(b) selecting the best model order according to some criterion for evaluating the resulting models (See [122] and the references therein for details).

Specifically, when the accumulated prediction error (APE) or the CIC (for identification criterion for control systems) is used as an order estimation criterion, the above approach is a very promising technique [29][72]. As shown by Hemmerly and Davis [72], this approach could generate strongly consistent estimation of system order and parameters if the unknown system can be modeled as an ARX model and a persistent excitation condition is satisfied. There, the noise disturbance of the system is assumed to be a martingale difference process. *No assumptions of stationary and ergodic output and input processes are made.* As shown in Appendix C, the assumption of martingale stochastic noise is true for many closed-loop systems. More significantly, the APE, combined with the adaptive control strategy devised in [25], has been used in self-tuning control of ARX systems with martingale difference noise[72]. It is shown there that, under some mild assumptions, the parameters and order are estimated in a strongly consistent way while the optimal cost of the whole adaptive control system is achieved asymptotically.

The order estimation approach by Hemmerly and Davis or by Chen et al. essentially proceeds by minimizing the APE over a finite set of LS ARX models of different order. The computation of the APE of all models usually does not form a major computational load for each time iteration. But the determination of all the LS ARX models could be prohibitive for on-line computation if there are many candidate models. This is often the case since the prior knowledge of the model is often poor. A set of ARX models can be efficiently obtained by using the Levinson-type algorithms in the special case that all models in question satisfy $p = q$ [62][148], where p and q represent the number of model feedback coefficients and model feedforward coefficients, respectively. However, they cannot be applied to determining LS parameter estimates for a set of ARX models with different p and q. For this more general case, people may suggest algorithms with suboptimal performance but low computational complexity. Indeed, an over-parametrization method has been implicitly suggested in [62][148]. However, as will be shown, the performance of the algorithm is not good because the over-parametrization causes the parameter estimates to have large variance and could delay the time when the right order estimate arrives.

In Chapter 4, a *fast* algorithm for parameter estimation of *a family* of ARX models has been developed. The algorithm is called the time and order recursive algorithm (TORA). The family of ARX models is described below

$$\mathbf{y}_n + \sum_{i=1}^{p} A_i \mathbf{y}_{n-i} = \sum_{j=0}^{q} C_j \mathbf{u}_{n-j} + w_n, \tag{5.3}$$

where $p = p_0, \cdots, p^* - 1, p^*$; $q = q_0, \cdots, q^* - 1, q^*$. $\mathbf{y}_n$ and $\mathbf{u}_n$ are model output and input and w_n is noise disturbance.

The TORA deals with the parameter estimation problem based upon the current and past measurements of output and input, $\{\mathbf{y}_t\}_{t=1}^n$ and $\{\mathbf{u}_t\}_{t=1}^n$. When new output and input measurements are received, the TORA updates the *data* autocorrelation coefficients and the *data* cross-correlation coefficients time-recursively. Then, it computes two sets of partial correlation coefficients in a partially order-recursive way. The first set of partial correlation coefficients is used for updating parameter estimates recursively in the order of the autoregressive part denoted by p. The second set is used for updating parameter estimates recursively in the order of the moving-average part denoted by q. Note that the partial correlation coefficients are dependent on model order (p, q). To complete the parameter recursion, three extra auxiliary estimators are introduced such that four sets of parameter estimates have to be determined for each order (p, q). When p (or q) is increased by 1, the corresponding parameter estimates of order (p+1,q) (or (p, q+1)) are produced as a weighted summation of the four sets of parameter estimates of order (p,q). The weighting factors are the partial correlation coefficients of the first set (or the second set).

The main advantage of the TORA over other relevant algorithms is its high efficiency and great parallelism (See Chapter 4 for details).

Fact 1 [*Computational Complexity:*] Consider parameter estimation for a family of ARX models of order $(p, q), p = p_0, \cdots, p^*-1, p^*;\ \ q = q_0, \cdots, q^*-1, q^*$. The TORA can efficiently compute all parameter estimates. At each time iteration, the number of multiplication operations is $\frac{1}{2}(4l^3+7l^2m)q^{*^2}(p^*-p_0)+(7m^2l+4ml^2)q^*(p^{*^2}-p_0^2)+\frac{1}{2}(4m^3+7m^2l)p^{*^2}+(3m^2l+7mlo2)p^*q_0\ \frac{1}{2}(7l^3+4lm^2)q_0^2+\mathbf{O}(p^*\vee q^*)$. And the number of addition operations is $\frac{1}{2}(4l^3+7l^2m)q^{*^2}(p^*-p_0)+(7m^2l+4ml^2)q^*(p^{*^2}-p_0^2)+\frac{1}{2}(4m^3+7m^2l)p^{*^2}+(3m^2l+7ml^2)p^*q_0+\frac{1}{2}(7l^3+4lm^2)q_0^2+\mathbf{O}(p^*\vee q^*)$ ∎

Fact 2 [*Parallelism:*] For each time iteration, the unparallelizable computations within the TORA take $(l\vee m)^3(p^*+q^*)+\mathbf{O}(1)$ multiplications and $(p^*+q^*)\log_2(l\vee m)+p^*\log_2(p^*+q_0+2)+(q^*-q_0)\log_2(p^*+q^*+3)+q_0\log_2(q_0+3)+\mathbf{O}(1)$ additions. ∎

What will happen to consistency if we replace LS parameter estimates in the Hemmerly and Davis approach by parameter estimates generated by the TORA? The primary purpose of this chapter is to show that this replacement preserves the strong consistency of ARX system order and parameter estimation under the assumptions made in [72]. In section 5.2, we investigate two well-known results about martingale analysis of consistent parameter estimation. This is done by showing that these results can be proved by using a unified approach suggested by Lai and Wei [92][149]. We also show that the TORA parameter estimates[1] preserve both the strong consistency of LS parameter estimation and the bound for the convergence rate. In section 5.3, we illustrate a fast method for simultaneous estimation of system order and parameters by combining the APE with the TORA together. Following the previous results, we show that the proposed method generates strongly consistent estimation of both ARX system order and parameters. Another highlight of this chapter is the simulation study presented in Section 5.4. The simulations are systematically conducted to investigate (1) the transient performance of parameter and order estimation, (2) how the system characteristics of stability, stability margin, controllability/observability, and fast dynamics affect the performance of parameter and order estimation. The final section provides some conclusions and suggestions for future work.

5.2 Strongly Consistent Parameter Estimation

Consider an ARX system

$$\mathbf{y}_n + A_1\mathbf{y}_{n-1} + \cdots + A_{p_t}\mathbf{y}_{n-p_t} = C_0\mathbf{u}_n + C_1\mathbf{u}_{n-1} + \cdots + C_{q_t}\mathbf{u}_{n-q_t} + \omega_n,$$

where A_{p_t} and C_{q_t} are of full row-rank and (p_t, q_t) is called the true order of the system. Consistent parameter estimation can occur only in the overmodeled

[1]Here, the TORA parameter estimates represent the parameter estimates generated by the TORA.

case. That is, the ARX model which is used for parameter estimation must have order (p,q) with $p \geq p_t$ and $q \geq q_t$. Mathematically, an ARX system of order (p_t, q_t) can be described by using a linear regression model:

$$\mathbf{y}_n = \theta^T(p,q)\phi_{n-1}(p,q) + \omega_n, \quad p \geq p_t, \ q \geq q_t, \tag{5.4}$$

where the regression model parameter (matrix) is defined as

$$\theta^T(p,q) \triangleq (-A_1 \cdots - A_{p_t} \ 0 \cdots \ 0 \ C_0 \cdots C_{q_t} \ 0 \cdots \ 0)$$

and the regression vector is expressed below

$$\phi_{n-1}(p,q) = (\mathbf{y}_{n-1}^T \cdots \mathbf{y}_{n-p_t}^T \cdots \mathbf{y}_{n-p}^T \ \mathbf{u}_n^T \cdots \mathbf{u}_{n-q_t}^T \cdots \mathbf{u}_{n-q_t}^T)^T.$$

For simplicity of notation, we will omit the arguments (p,q) except when confusion may occur.

Propagation Relation of LS Parameter Estimates

Define the *normal matrix* as

$$V_n \triangleq \sum_{t=1}^{n} \phi_{t-1}\phi_{t-1}^T + P_0^{-1}, \tag{5.5}$$

where P_0 is a positive definite matrix. Thus, the least-squares (LS) estimate $\hat{\theta}_n$ of parameter matrix θ at time instant n is the solution to the normal equation:

$$\underbrace{(\sum_{t=1}^{n} \phi_{t-1}\phi_{t-1}^T + P_0^{-1})}_{V_n}\hat{\theta}_n = \sum_{t=1}^{n} \phi_{t-1}\mathbf{y}_t^T. \tag{5.6}$$

The LS parameter estimate is often computed in a recursive manner[24]:

$$\hat{\theta}_n = \hat{\theta}_{n-1} + P_n\phi_{n-1}(\mathbf{y}_n^T - \phi_{n-1}^T\hat{\theta}_{n-1}) \tag{5.7}$$

where $P_n, n \geq 0$, which we call the *gain matrices*, are defined as

$$\begin{aligned} P_n &\triangleq V_n^{-1} \\ &= P_{n-1} - a_{n-1}P_{n-1}\phi_{n-1}\phi_{n-1}^T P_{n-1} \end{aligned} \tag{5.8}$$

with

$$a_n = (1 + \phi_{n-1}^T P_{n-1} \phi_{n-1})^{-1}. \tag{5.9}$$

The initial values $\hat{\theta}_0$ and P_0 in the recursive calculation are given by some rule. (5.5) and (5.7) yield $P_n^{-1}\hat{\theta}_n = P_{n-1}^{-1}\hat{\theta}_{n-1} + \phi_{n-1}y_n^T$. Summing the above identity from 1 to n generates $P_n^{-1}\hat{\theta}_n = P_0^{-1}\hat{\theta}_0 + \sum_{t=1}^n \phi_{t-1}\mathbf{y}_t^T$. This and (5.4) produce the propagation relation of $\hat{\theta}_n$ starting from $n = 0$:

$$\hat{\theta}_n = \theta + P_n P_0^{-1}(\hat{\theta}_0 - \theta) + P_n \sum_{t=1}^{n} \phi_{t-1}\omega_t^T, \tag{5.10}$$

where θ is defined right below Eq. (5.4). Eq. (5.10) represents the propagation relation of LS parameter estimates. As will be seen, it is useful in proving consistency of LS parameter estimation.

Consistency of LS Parameter Estimation

Using (5.8), (5.10) and the triangle inequality of induced norms generates:

$$\begin{aligned}\|\hat{\theta}_n - \theta\|^2 &\leq \|V_n^{-1}P_0^{-1}(\hat{\theta}_0 - \theta)\|^2 + \|V_n^{-1}\textstyle\sum_{t=1}^n \phi_{t-1}\omega_t^T\|^2 \\ &\leq \|V_n^{-1}\|^2 \cdot \|P_0^{-1}(\hat{\theta}_0 - \theta)\|^2 + \|V_n^{-1/2}\|^2 \cdot \|V_n^{-1/2}\textstyle\sum_{t=1}^N \phi_{t-1}\omega_t^T\|^2,\end{aligned}$$

where the norm for vectors is the l_2 norm and the norm for matrices is the spectral norm, i.e., $\|M\| \triangleq$ the root square of the maximum eigenvalue of $M^T M$.

Define a new quantity Q_n as

$$Q_n \triangleq (\sum_{t=1}^{n} \omega_t \phi_{t-1}^T) P_n (\sum_{t=1}^{n} \phi_{t-1}\omega_t^T). \tag{5.11}$$

Denote by $\lambda_{min}(n)$ and $\lambda_{max}(n)$ the minimum and maximum eigenvalues of the normal matrix V_n. Thus, we have

$$\|\hat{\theta}_n - \theta\|^2 \leq \|P_0^{-1}(\hat{\theta}_0 - \theta)\|^2 / \lambda_{min}^2(n) + \mathrm{trace}(Q_n)/\lambda_{min}(n). \tag{5.12}$$

Inequality (5.12) implies that $\lim_{n\to\infty} \lambda_{min}(n) = \infty$ is a necessary condition for consistent parameter estimation; otherwise there would be some bias of parameter estimates resulting from initial conditions. We can also see from (5.12) that to obtain consistent least-squares parameter estimates, we just need to have $\lambda_{min}(n)$ go to infinity and the ratio of trace(Q_n) over $\lambda_{min}(n)$ go to zero. The consistency of parameter estimation is only decided by trace$(Q_n)/\lambda_{min}(n)$ if $\lim_{n\to\infty}\lambda_{min}(n) = \infty$. Because of the physical meaning of Q_n, we will call it the *convergence factor* (of LS parameter estimates).

So far, two kinds of assumptions have been proposed to guarantee the strong consistency of parameter estimation. In both cases, the basic tool is the martingale local convergence theory. The first kind of assumptions, proposed by Lai and Wei[92], has the advantages that, "subject to a bound on the supremum of an αth moment (for $\alpha > 2$) of the disturbance process, one can permit an almost exponential rate of growth of the condition numbers," $\kappa(n) = \lambda_{max}(n)/\lambda_{min}(n), n \in Z_1$ [21]. The second kind, proposed by Chen [23], permits "a specific, unbounded, growth in the variance of the disturbance process" if "the growth of the condition numbers to be polynomial in a certain technical sense" is allowed [21].

Lai and Wei first derived an upper bound on trace(Q_n) and then proved consistency. Chen, however, used a direct method. In fact, some variations of Chen's proof can yield an upper bound on trace(Q_n), which then results in consistency immediately.

Lemma 5.1 Consider an ARX system in (5.4) with driving noise $\{\omega_t\}_{t=0}^{\infty}$ which is a martingale difference process w.r.t. an increasing family of σ-fields $\mathcal{F}_t, t \in Z_1$.

Write $r_t = \text{trace}(V_t)$. Assume

(G1) there exists $k_0 > 0$ and $\epsilon \in [0, 1)$ such that

$$E\|\omega_t\|^2|\mathcal{F}_{t-1}) \leq k_0 r_t^{\epsilon} \quad a.s. \quad \forall t \in Z_+$$

(G2)

$$\lambda_{min}(t) \to \infty \quad a.s. \quad \text{as} \quad t \to \infty \tag{5.13}$$

and there is an a.s. finite random variable γ and a constant $\alpha \in [0, \frac{1}{2})$ such that for all $n \in Z_1$,

$$[\text{trace}(V_n)]^{1-\alpha} = r_n^{1-\alpha} \leq \gamma\lambda_{min}(n) \;\; a.s. \quad \alpha < \frac{1}{2} - \frac{\epsilon}{2} \tag{5.14}$$

where $\epsilon \in [0,1)$ is the constant appearing in (**G**1). Then,

$$\text{trace}(Q_n) = \mathbf{o}(r_n^{2\delta-\alpha}) \tag{5.15}$$

where $2\delta - \alpha > 0$ and δ and α are the constants appearing in (**G**1) and (**G**2). ∎

Proof:

$$\begin{aligned} \tfrac{1}{r_n^{2\delta-\alpha}} \cdot \text{trace}(Q_n) &\leq \tfrac{1}{r_n^{2\delta-\alpha}} \lambda_{min}^{-1}(n) \|\textstyle\sum_{k=1}^n \phi_{k-1}\omega_k^T\|^2 \quad \text{(by (5.11))} \\ &\leq \gamma \tfrac{1}{r_n^{2\delta-\alpha}} \tfrac{1}{r_n^{1-\alpha}} \|\textstyle\sum_{k=1}^n \phi_{k-1}\omega_k^T\|^2 \quad \text{(by (5.14))} \\ &= \gamma(\tfrac{\|\sum_{k=1}^n \phi_{k-1}\omega_k^T\|}{r_n^{1/2+\delta-\alpha}})^2. \end{aligned} \tag{5.16}$$

It follows from Remark C.1 that $\{\frac{\phi_{t-1}\omega_t^T}{r_t^{(1/2)+\delta-\alpha}}, \mathcal{F}_t\}$ is a martingale difference process and hence, from (**G1**),

$$\sum_{t=1}^{\infty} E(\frac{\|\phi_{t-1}\omega_t^T\|}{r_t^{(1/2)+\delta-\alpha}})^2 |\mathcal{F}_{t-1}) \leq k_0 \sum_{t=1}^{\infty} \frac{\|\phi_{t-1}\|^2}{r_t^{1+2\delta_0}} < \infty, \tag{5.17}$$

where $\delta_0 = \delta - \alpha - \epsilon/2 > 0$. The second inequality in (5.17) comes from (**G2**) and Lemma C.2 because $r_t = \sum_{k=1}^t \|\phi_{t-1}\|^2 + \text{trace}(P_0^{-1})$ is divergent. Therefore, we have, from (C.11) in Lemma C.4,

$$\sum_{t=1}^{\infty} \frac{\phi_{t-1}\omega_t^T}{r_t^{(1/2)+\delta-\alpha}} \quad \text{converges} \quad a.s. \tag{5.18}$$

and then, it follows from Lemma C.2 that

$$\frac{1}{r_n^{1/2+\delta-\alpha}} \sum_{k=1}^{n} \phi_{k-1}\omega_k^T \to 0 \quad a.s. \qquad \text{as } n \to \infty. \tag{5.19}$$

∎

Lemma 5.1 and (5.12) immediately yield the following theorem.

Theorem 5.1 [Chen, [23]]: Under the assumptions of Lemma 1, $\hat{\theta}_n \to \theta$ $a.s.$ as $n \to \infty$ with convergence rate

$$\|\hat{\theta}_n - \theta\| = \mathbf{O}(r_n^{\delta-1/2}) \quad \forall\delta \in (\alpha + \frac{\epsilon}{2}, \frac{1}{2}]. \tag{5.20}$$

∎

Consequently, we now see that Chen's result can be obtained by deriving an upper bound on trace(Q_n) and then applying it to (5.12). In fact, along the same path, Lai and Wei proved another well known result which is presented in Theorem 5.2. Therefore, estimating an upper bound of the convergence rate trace(Q_n) becomes the core of a unified approach for proving consistency and determining the convergence rate of LS parameter estimates. For completeness, we state the result of Lai and Wei.

Theorem 5.2 [Lai and Wei, [92]]: Consider an ARX system in (5.4) with driving noise $\{\omega_t\}_{t=0}^{\infty}$ which is a martingale difference process w.r.t. an increasing family of σ-fields $\mathcal{F}_t, t \in Z_1$.

(H1) $\sup_{t\in Z_1} E\|\omega_t\|^{\alpha}|\mathcal{F}_{t-1}) < \infty \;\; a.s.$ for some $\alpha > 2$.

Assume that

(H2)

$$\begin{aligned} &\lambda_{min}(n) \to \infty \;\; a.s. \;\; \text{as} \;\; n \to \infty \\ &\log \lambda_{max}(n) = \mathbf{o}(\lambda_{min}(n)) \;\; a.s. \end{aligned} \tag{5.21}$$

Then the least-squares parameter estimate $\hat{\theta}_n$ is a strongly consistent estimate of θ, and further,

$$\|\hat{\theta}_n - \theta\| = \mathbf{O}((\frac{\log(\lambda_{max}(n))}{\lambda_{min}(n)})^{1/2}) \quad a.s.. \tag{5.22}$$

∎

Remark 5.1 (Comparison of the convergence rates in Theorem 5.1 and Theorem 5.2) When an LS parameter estimation problem is well posed, the condition number of the problem must be finite; otherwise, the solution of the problem could have infinite computation error. How to maintain the finiteness of the condition number has been extensively studied (See [93] and the references therein). Thus, for a well-posed LS parameter estimation problem, assumption (5.14) is always true with $\alpha = 0$ if the number of parameters to be estimated is finite. Hence, it follows from (5.20) that

$$\|\hat{\theta}_n - \theta\| = \mathbf{O}(\lambda_{min}^{\delta-1/2}(n)), \quad \forall \delta \in (\frac{\epsilon}{2}, \frac{1}{2}) \tag{5.23}$$

Furthermore, assume that stochastic noise ω_t has finite second moment, which is true in most real situations. Thus, (**G1**) holds with $\epsilon = 0$ and then, from (5.23), the bound on the convergence rate (of LS parameter estimation) is very close to $\mathbf{O}(\lambda_{min}^{-1/2}(n))$. Under the above assumptions, the bound given in Theorem 5.2 is $\mathbf{O}((\frac{\log \lambda_{min}(n)}{\lambda_{min}(n)})^{1/2})$, which is faster than $\mathbf{O}(\lambda_{min}^{\delta-1/2}(n))$ for any $\delta \in (0, 1/2)$. ∎

Remark 5.2 (Constraints on convergence rate) According to Remark 5.1, the larger $\lambda_{min}(n)$ is, the faster the convergence rate of parameter estimation could be. However, $\lambda_{min}(n)$ can not be arbitrarily large. For example, the almost sure boundness of input and output processes implies that $\lambda_{min}(n) = \mathbf{O}(n)$ $a.s.$. In addition, a big growth rate of $\lambda_{min}(n)$ is not always permitted because $\lambda_{min}(n)$ is bounded above by a sum of a constant and the "total energy" of input signals:

$$\lambda_{min}(n) = \lambda_{min}(p, q, n) \leq \sum_{t=1}^{n} \|\mathbf{u}_t\|^2 + \text{trace}(P_0^{-1}(0,0)), \quad \forall p > 0,\ q > 0, \tag{5.24}$$

for each realization of the output and input processes. Remember that $\lambda_{min}(p, q, n)$, $n \in Z_1$, are the minimum eigenvalues of normal matrices

$$V_n(p,q) = \sum_{k=1}^{n} \phi_{k-1}(p,q)\phi_{k-1}^T(p,q) + P_0^{-1}(p,q)$$

and $\phi_{k-1}(p,q)$, $k \in Z_1$, are regression vectors defined as

$$\phi_{k-1}(p,q) \triangleq (\mathbf{y}_{k-1}^T \cdots \mathbf{y}_{k-p}^T\ \mathbf{u}_k^T\ \mathbf{u}_{k-1}^T\ \mathbf{u}_{k-q}^T)^T,\ p \geq 0,\ q \geq 0.$$

Specifically,

$$V_n(0,0) = \sum_{k=1}^{n} \mathbf{u}_k \mathbf{u}_k^T + P_0^{-1}(0,0),$$

where $P_0^{-1}(0,0)$ is a positive definite matrix of proper dimension. To see (5.24), recall that for a symmetric matrix R the following relationships hold:

$$\lambda_{min} = \min_{\|\mathbf{x}\|=1} \mathbf{x}^T R \mathbf{x} \qquad \text{and} \qquad \lambda_{max} = \max_{\|\mathbf{x}\|=1} \mathbf{x}^T R \mathbf{x}. \tag{5.25}$$

So, if $\{R_m\}$ is a sequence of nested symmetric matrices such that $R_{m+1} = \begin{pmatrix} R_m & * \\ * & * \end{pmatrix}$, where the asterisks represent the elements of R_{m+1} which are not interesting to us, it follows from (5.25) that

$$\lambda_{min}(R_{m+1}) \leq \lambda_{min}(R_m) \qquad \text{and} \qquad \lambda_{max}(R_{m+1}) \geq \lambda_{max}(R_m) \tag{5.26}$$

and the condition number satisfies:

$$\kappa(R_{m+1}) \geq \kappa(R_m). \tag{5.27}$$

Obviously, $V_n(p,q),\ n \in Z_1$, are nested matrices when either p or q is decreased. It follows from (5.26) that

$$\sum_{k=1}^{n} \|\mathbf{u}_k\|^2 + \mathrm{trace}(P_0^{-1}(0,0)) = \mathrm{trace}(V_n(0,0)) \geq \lambda_{min}(0,0,n) \geq \lambda_{min}(p,q,n). \tag{5.28}$$

∎

Remark 5.3 **(Effect of extra parameter estimates on convergence rate)** The order of least squares parameter estimators is of great importance for the consistency and convergence rate of the corresponding parameter estimates. Obviously, any parameter estimator is not consistent in any sense if the corresponding parameter estimator is strictly undermodeled , i.e., $p < p_t$ or $q < q_t$. Now we consider the overmodeled case, i.e., parameter estimators of order (p,q), $p \geq p_t$, $q \geq q_t$. Recall from (5.4) that an ARX system with order (p_t, q_t) can be described by a linear regression model of order (p,q) satisfying $p \geq p_t$ and $q \geq q_t$. And the corresponding regression model parameter matrix is equal to

$$\theta^T(p,q) = (-A_1 \ \cdots - A_{p_t} \ \underbrace{0 \cdots 0}_{p-p_t} \ \ C_0 \ \cdots C_{q_t} \ \underbrace{0 \cdots 0}_{q-q_t}).$$

For such a linear regression model, the least-squares parameter estimates, $\hat{\theta}_N(p,q)$, $N \in Z_1$, of the parameter matrix $\theta(p,q)$ are equal to the solutions to the following normal matrices:

$$\underbrace{(\sum_{k=1}^{N} \phi_{k-1}(p,q)\phi_{k-1}^T(p,q) + P_0^{-1})}_{V_N(p,q)} \hat{\theta}_N(p,q) = \sum_{k=1}^{N} \phi_{k-1}(p,q)\mathbf{y}_k^T.$$

Thus, (5.22) can be rewritten as

$$\|\hat{\theta}_N(p,q) - \theta(p,q)\| = \mathbf{O}((\frac{\log(\lambda_{max}(p,q,N))}{\lambda_{min}(p,q,N)})^{1/2}) \quad a.s.$$

provided that the assumptions of Theorem 5.2 hold, where $\lambda_{min}(p,q,N)$ and $\lambda_{max}(p,q,N)$ denote the minimum and maximum eigenvalues of $V_N(p,q)$. Note from the structure of the matrix $V_N(p,q)$ that in the strictly overmodeled case: $p \geq p_t$, $q \geq q_t$, $p+q > p_t+q_t$, $V_N(p_t,q_t)$ is always a nested symmetric matrix of $V_N(p,q)$. By (5.26) we have an expected result that in the strictly overmodeled case, the convergence rates of parameter estimates are at best equal to the convergence rate of the LS parameter estimator of order (p_t,q_t). In fact, overparametrization usually causes parameter estimates to have large variance. This will be discussed more deeply in Section 5.5.

In addition, it follows from (5.27) that $\kappa(V_n(p_t,q_t)) \leq \kappa(V_n(p,q))$. So, assumptions (5.21) and (**G**1) could become more restrictive to consistent least-squares parameter estimators. ∎

Consistency of TORA Parameter Estimates

TORA parameter estimates are a good approximation of the corresponding LS parameter estimates. This can be seen from Theorem 4.4 which can be generalized as follows.

Lemma 5.2 Suppose that $\mathbf{y}$ and $\mathbf{u}$ are almost surely and uniformly bounded. If (i) $\lambda_{min}(p,q,n) \rightarrow \infty$ a.s. as $n \rightarrow \infty$, (ii) the LS parameter estimates $\hat{\theta}_n(p,q), n \leq 1$, are uniformly bounded; i.e., $\exists K_1$ such that $\|\hat{\theta}_n(p,q)\| < K_1$ $\forall n \leq 1$, then

$$\|\hat{\theta}_n(p,q) - \tilde{\theta}_n(p,q)\| = \mathbf{O}(\lambda_{min}^{-1}(p,q,n)) \quad a.s. \tag{5.29}$$

where $\tilde{\theta}_n(p,q)$, $n \leq 1$, are the corresponding TORA parameter estimates. ∎

Obviously, Assumption 2 in Lemma 5.2 is implied by the strong consistency of LS parameter estimates. Therefore, we have the following theorem.

Theorem 5.3 Consider an ARX system in (5.4). Suppose that the output and input processes are uniformly bounded for each realization. If (1) the true model order is in the model complexity set, i.e., $(p_t,q_t) \in \mathcal{O}_{p_0,q_0}^{p^*,q^*}$, (2) the ARX system satisfies the assumption of either Theorem 5.2 or Theorem 5.1 for any parameter estimator order $(p,q) \in \mathcal{O}^{p^*,q^*}_{p_t,q_t}$, then the TORA estimates $\tilde{\theta}_n(p,q), (p,q) \in \mathcal{O}^{p^*,q^*}_{p_t,q_t}$, are strongly consistent. Furthermore,

$$\|\tilde{\theta}_n(p,q) - \theta(p,q)\| = \mathbf{o}((\frac{\log(\lambda^y_{max}(p,q,n))}{\lambda^y_{min}(p,q,n)})^{1/2}) \quad a.s. \quad \forall (p,q) \in \mathcal{O}^{p^*,q^*}_{p_t,q_t}. \tag{5.30}$$

∎

5.3 Strongly Consistent Order Estimation

Recently, Hemmerly and Davis[72] developed an approach for strongly consistent order estimation for ARX systems. They consider an m-output and l-input ARX system:

$$\mathbf{y}_n + A_1\mathbf{y}_{n-1} + \cdots + A_{p_t}\mathbf{y}_{n-p_t} = C_0\mathbf{u}_n + C_1\mathbf{u}_{n-1} + \cdots + C_{q_t}\mathbf{u}_{n-q_t} + \omega_n \tag{5.31}$$

where the order (p_t, q_t) and parameter matrix

$$\theta^T(p_t,q_t) \triangleq [-A_1 \ \cdots \ -A_{p_t} \ \ C_0 \ \cdots \ C_{q_t}]$$

are unknown. System (5.31) is assumed to satisfy:

A-I The noise $\{\omega_n\}$ is a martingale difference process satisfying

$$E\|\omega_n\|^2|\mathcal{F}_{n-1}) = \sigma^2 \quad a.s. \tag{5.32}$$

$$\sup_n E\|\omega_n\|^\alpha|\mathcal{F}_{n-1}) < \infty \quad a.s., \quad \text{for some } \alpha > 2. \tag{5.33}$$

A-II The true order (p_t, q_t) belongs to a known finite set, which we call the model complexity set,

$$\mathcal{O}^{p^*,q^*}_{0,0} \triangleq \{(p,q); 0 \le p \le p^*, 0 \le q \le q^* \text{ for some positive integers } p^* \text{and } q^*\} \tag{5.34}$$

A-III The matrices A_{p_t} and C_{q_t} are of row-full rank.

The order estimate in the approach by Hemmerly and Davis is determined by minimizing the LS accumulated "honest" prediction error (LS–APE for short)

$$(\hat{p}_n, \hat{q}_n) = \arg \min_{(p,q)\in\mathcal{O}^{p^*,q^*}_{0,0}} \sum_{k=1}^{n} \text{LS–APE}(p,q)$$

where

$$\text{LS–APE}(p,q) \triangleq \|y_k - \hat{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2. \tag{5.35}$$

As we discussed before, the computational bottleneck of the approach is the determination of LS parameter estimates $\hat{\theta}_n(p,q)$ for all possible models. Since the TORA is a fast algorithm for determining parameter estimates $\tilde{\theta}_n(p,q)$ for all the possible models and the TORA estimates converge to the corresponding LS estimates under some mild conditions, it is natural to consider replacing the LS estimates in the LS–APE by the TORA estimates. This idea results in the following *fast* method for simultaneous estimation of system order and parameters.

1. Obtain the TORA parameter estimate

$$\tilde{\theta}_n^T(p,q) = [-\tilde{A}_{\mathrm{n},1}^{\mathrm{p},\mathrm{q}} \;\cdots\; -\tilde{A}_{\mathrm{n},\mathrm{p}}^{\mathrm{p},\mathrm{q}} \;\; \tilde{C}_{\mathrm{n},0}^{\mathrm{p},\mathrm{q}} \;\; \tilde{C}_{\mathrm{n},1}^{\mathrm{p},\mathrm{q}} \;\cdots\; \tilde{C}_{\mathrm{n},\mathrm{q}}^{\mathrm{p},\mathrm{q}}]$$

   for each order $(p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}$.

2. Obtain the order estimate by minimizing the accumulated "honest" prediction error (APE) over the model complexity set $\mathcal{O}_{0,0}^{p^*,q^*}$

$$(\tilde{p}_n, \tilde{q}_n) = \arg \min_{(p,q)\in\mathcal{O}_{0,0}^{p^*,q^*}} \sum_{k=1}^{n} \|y_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2. \tag{5.36}$$

To distinguish the APE in (5.36) from the LS–APE in (5.35) we denote

$$\sum_{k=1}^{n} \|y_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2$$

by TORA-APE (p,q). Actually, the only difference between the proposed method and the Hemmerly and Davis approach is the parameter estimates used. In their approach, LS parameter estimates $\hat{\theta}_{k-1}(p,q)$ are used and the APE is defined as $APE(p,q,n) \triangleq \sum_{k=1}^{n} \|y_k - \hat{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2$. In the proposed method, TORA parameter estimates are used to reduce the computational burden. The strong consistency of the proposed method can be proved by following some previous results.

Theorem 5.4 [Hemmerly and Davis, [72]] Suppose that ARX system (5.31) satisfies assumptions **A-I**— **A-III**. If for each model order $(p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}$

$$\phi_n^T(p,q)V_n^{-1}(p,q)\phi_n(p,q) \to 0 \quad a.s. \ \text{ as } \ n \to \infty \tag{5.37}$$

$$\lambda_{min}(p,q,n) \to \infty \quad a.s. \ \text{ as } \ n \to \infty \tag{5.38}$$

and

$$\lambda_{max}(p,q,n) = \mathbf{O}(\lambda_{min}(p,q,n)(\log \lambda_{min}(p,q,n))^\gamma) \quad a.s., \gamma < 1 - \frac{2}{\alpha}, \tag{5.39}$$

where α is the constant appearing in (5.33), then for n big enough,

$$\text{LS–APE}(p,q,n) > \text{LS–APE}(p_t,q_t,n) \quad a.s. \ \ \forall (p,q) \neq (p_t,q_t) \text{ and } (p,q) \in \mathcal{O}_{0,0}^{p^*,q^*} \tag{5.40}$$

where $V_n(p,q) \triangleq \sum_{k=1}^n \phi_{k-1}(p,q)\phi_{k-1}^T(p,q)$ and $\lambda_{min}(p,q,n)$ and $\lambda_{max}(p,q,n)$ are the minimum and maximum eigenvalues of $V_n(p,q)$. As a result of (5.40), the order estimate $(\hat{p}_n, \hat{q}_n)$ is strongly consistent. ∎

Observation 5.1: Suppose that the driving noise ω_n satisfying (5.33) and the "honest" predictions $\hat{\mathbf{y}}_n(p,q), n \in Z_1$, are measurable with respect to $\mathcal{F}_{n-1}$. Then applying Lemma C.5 to the APE immediately yields:

$$\sum_{k=1}^n \|\mathbf{y}_k - \hat{\mathbf{y}}_k(p,q)\|^2 = \sum_{k=1}^n \|\omega_k\|^2 + C_n(1 + \mathbf{o}(1)) \quad a.s. \tag{5.41}$$

on the set $\{C_n \to \infty\}$ and

$$\sum_{k=1}^n \|\mathbf{y}_k - \hat{\mathbf{y}}_k(p,q)\|^2 = \sum_{k=1}^n \|\omega_k\|^2 + C_n(1 + \mathbf{O}(1)) \quad a.s. \tag{5.42}$$

on the set $\{\lim_{n\to\infty} C_n < \infty\}$, where $C_n \triangleq \sum_{k=1}^n \|\mathbf{y}_k - \hat{\mathbf{y}}_k(p,q) - \omega_k\|^2$ is called the accumulated pure prediction error (APPE). $C_n\mathbf{o}(1)$ and $C_n\mathbf{O}(1)$ denote $\mathbf{o}(C_n)$ and $\mathbf{O}(C_n)$. (5.41) and (5.42) reflect that the limit behavior of APE (p,q,n) is decided by the APPE $C_n(p,q)$ and the accumulated energy of noise.

Specifically, we define the LS–APPE as

$$C_n(p,q) \triangleq \sum_{k=1}^n \|\mathbf{y}_k - \hat{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q) - \omega_k\|^2. \tag{5.43}$$

Then,

$$\sum_{k=1}^{n} \|\mathbf{y}_k - \hat{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2 - \sum_{k=1}^{n} \|\mathbf{y}_k - \hat{\theta}_{k-1}^T(p_t,q_t)\phi_{k-1}(p_t,q_t)\|^2$$

$$= (C_n(p,q) - C_n(p_t,q_t))(1+\mathbf{o}(1)) \qquad (5.44)$$

on the set $\{C_n(p,q) \to \infty, C_n(p_t,q_t) \to \infty\}$. *Thus, the APPE becomes a key factor for consistent order estimation provided the LS-APE is used as an order estimation criterion.* ∎

Observation 5.2: It has been shown [Lemma 2.2, [72]] that under the assumptions of Theorem 5.4, LS–APPE $C_n(p,q)$ defined in (5.43) has the following limit behavior:

$$C_n(p,q) = (1+\mathbf{o}(1))\sigma^2 \log \det(V_n(p,q)) \quad a.s. \qquad (5.45)$$

when $p \geq p_t$ and $q \geq q_t$. It has also been proved [(67), [72]] that in the undermodeled case, for instance, $p < p_t$,

$$\text{LS–APE}(p,q,n) \geq \sum_{k=1}^{n} \|w_n\|^2 + (1+\mathbf{o}(1))\|A_{p_t}\|^2 \lambda_{min}(p,q,n) + \mathbf{O}(1). \quad a.s.$$

Thus, (5.38), (5.41), and (5.42) imply that $C_n(p,q) \to \infty$ $a.s.$ as $n \to \infty$ and

$$C_n(p,q) \geq (1+\mathbf{o}(1))\|A_{p_t}\|^2 \lambda_{min}(p,q,n) \quad a.s. \qquad (5.46)$$

in the undermodeled case. ∎

Lemma 5.3 Suppose that $\mathbf{y}$ and $\mathbf{u}$ are almost surely and uniformly bounded. Then, under the assumptions of Theorem 5.4,

$$\sum_{k=1}^{n} \|\mathbf{y}_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q) - \omega_k\|^2 \sim C_n(p,q) \quad a.s. \qquad (5.47)$$

∎

To prove Lemma 5.3, we need the following lemma.

Lemma 5.4 Let $\{a_t, t \in Z_+\}$ be a sequence of positive numbers. If the partial sums $\{b_t = \sum_{i=0}^{t} a_i, t \in Z_+\}$ are divergent, then

$$\sum_{t=0}^{n} a_t/b_t \leq 1 + \log b_n - \log b_0. \tag{5.48}$$

In addition, if $\{x_t, t \in Z_+\}$ is a sequence of positive numbers which converges to zero, then

$$\sum_{t=0}^{n} x_t a_t = \mathbf{o}(\sum_{t=0}^{n} a_t). \tag{5.49}$$

∎

Proof:

(i) $\sum\limits_{t=0}^{n} a_t/b_t = 1 + \sum\limits_{t=1}^{n}(b_t - b_{t-1})/b_t \leq 1 + \sum\limits_{t=1}^{n} \log(b_t/b_{t-1}) = 1 + \log b_n - \log b_0.$

(ii) For any given $\epsilon > 0$, $\exists \epsilon', N > 0$ such that $x_n < \epsilon' < \epsilon$, $n \geq N$. Hence, for any $n > N$,

$$0 \leq \frac{\sum_{t=0}^{n} x_t a_t}{\sum_{t=0}^{n} a_t} \leq \frac{\sum_{t=0}^{N} x_t a_t}{\sum_{t=0}^{n} a_t} + \frac{\epsilon' \sum_{t=N+1}^{n} a_t}{\sum_{t=0}^{n} a_t} \leq \frac{\sum_{t=0}^{N} x_t a_t}{\sum_{t=0}^{n} a_t} + \epsilon'.$$

Letting n be big enough, we have $0 \leq \frac{\sum_{t=0}^{N} x_t a_t}{\sum_{t=0}^{n} a_t} < \epsilon$ since $\sum_{t=0}^{n} a_t \to \infty$ as $n \to \infty$. ∎

Returning now to Lemma 5.3, from the definition of $C_n(p,q)$ in (5.43) we have that for any (p,q),

$$C_n(p,q) + \sum_{k=1}^{n} \|(\hat{\theta}_{k-1}^T(p,q) - \tilde{\theta}_{k-1}^T(p,q))\phi_{k-1}(p,q)\|^2$$

$$\geq \sum_{k=1}^{n} \|\mathbf{y}_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q) - \omega_k\|^2$$

$$\geq C_n(p,q) - \sum_{k=1}^{n} \|(\hat{\theta}_{k-1}^T(p,q) - \tilde{\theta}_{k-1}^T(p,q))\phi_{k-1}(p,q)\|^2.$$

This implies that

$$|C_n(p,q) - \sum_{k=1}^{n} \|\mathbf{y}_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q) - \omega_k\|^2|$$

$$\leq \sum_{k=1}^{n} \|\hat{\theta}_{k-1}^T(p,q) - \tilde{\theta}_{k-1}^T(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2.$$

At this point we care about the limit behavior of sum

$$\sum_{k=1}^{n} \|\hat{\theta}_{k-1}^{T}(p,q) - \tilde{\theta}_{k-1}^{T}(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2.$$

It follows from Lemma 5.2 and (5.39) that for almost every realization of $(\mathbf{y}, \mathbf{u})$, there exists an finite number N such that

1. $\|\hat{\theta}_n(p,q) - \tilde{\theta}_n(p,q)\| \le K_1 \lambda_{min}^{-1}(p,q,n)$ for some finite positive number K_1,

2. $\lambda_{max}(p,q,n) \le K_2 \lambda_{min}^{3/2}(p,q,n)$ for some finite positive number K_2.

Hence, for any $n > N$,

$$\sum_{k=1}^{n} \|\hat{\theta}_{k-1}(p,q) - \tilde{\theta}_{k-1}(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2$$

$$\le \sum_{k=1}^{N} \|\hat{\theta}_{k-1}(p,q) - \tilde{\theta}_{k-1}(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2 + \sum_{k=N+1}^{n} K_1^2 \lambda_{min}^{-2}(p,q,k) \|\phi_{k-1}(p,q)\|^2. \tag{5.50}$$

Note that the first term on the right side of (5.50) is finite. The second term satisfies

$$K_1^2 \sum_{k=N+1}^{n} \lambda_{min}^{-2}(p,q,k) \|\phi_{k-1}(p,q)\|^2$$

$$\le K_1^2 K_2 \sum_{k=1}^{n} \lambda_{min}^{-1/2}(p,q,k) \lambda_{max}^{-1}(p,q,k) \|\phi_{k-1}(p,q)\|^2. \tag{5.51}$$

Using assumption (5.38) and (5.49) in Lemma 5.4, we have that

$$\sum_{k=1}^{n} \|\hat{\theta}_{k-1}(p,q) - \tilde{\theta}_{k-1}(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2$$

$$= \mathbf{o}\Big(\sum_{k=1}^{n} \lambda_{max}^{-1}(p,q,k) \|\phi_{k-1}(p,q)\|^2\Big) + \mathbf{O}(1)$$

$$\le \mathbf{o}\Big(\sum_{k=1}^{n} K_3 \frac{\|\phi_{k-1}(p,q)\|^2}{\mathrm{trace}(\sum_{t=1}^{k} \phi_{t-1}(p,q)\phi_{t-1}^{T}(p,q))}\Big), \tag{5.52}$$

where K_3 is the dimension of matrix $\sum_{k=1}^{n} \phi_{k-1}(p,q)\phi_{k-1}^{T}(p,q)$. Applying (5.48) in Lemma 5.4 to (5.52) and then using (5.49) yield

$$\sum_{k=1}^{n} \|\hat{\theta}_{k-1}(p,q) - \tilde{\theta}_{k-1}(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2 \le \mathbf{o}(\log \mathrm{trace}(V_n(p,q))) + \mathbf{O}(1)$$

$$= \mathbf{o}(\log \lambda_{max}(p,q,n)) + \mathbf{O}(1) \leq \mathbf{o}((1+\mathbf{o}(1)) \log \lambda_{min}(p,q,n)) + \mathbf{O}(1). \quad (5.53)$$

Comparing (5.53) with either (5.45) or (5.46) yields

$$\sum_{k=1}^{n} \|\hat{\theta}_{k-1}(p,q) - \tilde{\theta}_{k-1}(p,q)\|^2 \|\phi_{k-1}(p,q)\|^2 = \mathbf{o}(C_n(p,q)) \quad a.s.$$

for $\forall (p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}$ ∎

Lemma 5.4 and Observation 5.1 tell us that the LS–APE(p,q,n) used in the Hemmerly and Davis approach has the same limit behavior as the TORA–APE

$$\sum_{k=1}^{n} \|y_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2.$$

Noting that the assumptions of Theorem 5.2 are implied by the assumptions of Theorem 5.4, thus, we have the following theorem.

Theorem 5.5 Consider stochastic control systems in (5.31). Suppose that the input and output processes are uniformly bounded for each realization. If the estimation of the system order and parameters is approached in the following manner:

$$(\tilde{p}_n, \tilde{q}_n) = \arg \min_{(p,q) \in \mathcal{O}_{0,0}^{p^*,q^*}} \{\sum_{k=1}^{n} \|y_k - \tilde{\theta}_{k-1}^T(p,q)\phi_{k-1}(p,q)\|^2\} \quad (5.54)$$

and

$$\tilde{\theta}_n^T(p,q) = \text{ TORA parameter estimates,} \quad (5.55)$$

then , under assumptions **A-I** — **AI-II** and (5.37) — (5.39),

$$(\tilde{p}_n, \tilde{q}_n) \to (p_t, q_t) \quad a.s. \quad \text{as} \quad n \to \infty, \quad (5.56)$$

and

$$\tilde{\theta}_n(\tilde{p}_n, \tilde{q}_n) \to \theta(p_t, q_t) \quad a.s. \quad \text{as} \quad n \to \infty, \quad (5.57)$$

with a convergence rate of $\mathbf{O}((\frac{\log(\lambda_{min}(p,q,n))}{\lambda_{min}(p,q,n)})^{1/2})$. ∎

Remark 5.4 (Applications to adaptive control) The LS–APE, combined with the adaptive control strategy devised in [25], has been used in self-tuning control of ARX systems with martingale difference noise[72]. It is shown there that, under the assumptions of Theorem 5.4, the parameters and order are estimated in a strongly consistent way while the optimal cost of the whole adaptive control system is achieved asymptotically. This implies that the TORA-APE can also be applied to self-tuning control of ARX systems.

5.4 Simulation Study

A large number of simulations has been done on both parameter estimation and order estimation for ARX systems by using Pro-Matlab. As we know, all the order estimates discussed in this paper are obtained through comparisons of accumulated prediction errors (APE's) of "true" output predictors of different order. Four kinds of output predictors have been considered in simulations. They are LS predictors, TORA predictors, OP predictors and modified TORA predictors. The parameters of the predictors are generated via the LS parameter estimation algorithm, the TORA, the OP algorithm,[2] and modified TORA, respectively. The corresponding order estimates are denoted by the LS, TORA, OP, and modified TORA order estimates.

The simulations are conducted to further investigate parameter estimation and order estimation. Specifically, we wish (1) to investigate the transient performance of parameter estimates, including how the TORA, OP, and modified TORA estimates deviate from the corresponding LS estimates, (2) to show how the minimum eigenvalues of normal matrices affect the convergence rate of parameter estimates, (3) to study the effect of some system characteristics, such as stability, stability margin, controllability/observability, and fast dynamics, on the minimum eigenvalues and then on the parameter estimates. For order estimation, the following are investigated or demonstrated: (4) the performance of

[2]The OP algorithm represents the algorithm suggested in [62], where the parameter estimates of ARX models having different numbers of autoregressive terms and moving-average terms are determined via over-parameterized models.

order estimates when the iteration number is small, (5) the transition of condition numbers of the normal matrices, (6) the effect of the system characteristics on the condition numbers and then APE's. In addition, we also hope to see if the simulation results support Rissanen's claim that the APE is a criterion for stochastic inquiry[122].

Here, we report a selection of results that illustrate five typical situations, where the model inputs are always pseudo white noises with variance of 1.5. The model noises are pseudo white noises with variance of 0.5. We describe a single-input single-output ARX model in (5.3), for each case, by

$$\{(p,q); [a_1, a_2, \cdots, a_p; -c_0, -c_1, \cdots, -c_q]; [poles; zeros]\},$$

the last two being the poles and zeros of the ARX models in question. The simulation results are summarized, for each situation, in two tables on parameter estimates and order estimates and one set of figures. The tables reflect the sample averages and standard deviations of parameter estimates and the distributions of order estimates over the possible model order, which are computed from 50 replications. In the table on parameter estimates, the numbers in the first column represent the model parameters. The other columns have 5 segments, the first segment showing the iteration number. The others provide the sample averages and corresponding standard deviations which are presented in parentheses. The table on order estimates has the same structure as the table of parameter estimates, where the numbers in the first column are possible model order and the ones in bold-face are the true model order. The other columns express the distributions of various order estimates for different iteration numbers. For simplicity, the order of the moving-average part of ARX systems, q, are assumed known. The order of the autoregressive part of model candidates varies from 1 to 6. The simulation results, as well as some internal information, of one of the 50 replications are also depicted in the figures, where the horizontal axes always represent the iteration number. The plots in the figures consist of curves of model input/output, parameter/order estimates, and APE's, as well as minimum eigenvalues and condition numbers of normal matrices.

Example 1: A stable but lightly damped ARX system is considered here. The model parameters are

$$\{(3,0); [2.4, 1.91, 0.504; -1.0]; [-0.9, -0.8, -0.7]; \}.$$

The parameter/order estimates of the system are expressed in Table 5.1 and Table 5.2 and Figures 5.1 and 5.2. The simulation results in this example show that the LS parameter estimation algorithm is superior in terms of convergence rate over the other three algorithms. The superiority is significant for the TORA and OP algorithms, but is slight compared with the modified TORA. As we pointed out before, the deviation of the TORA estimates from the corresponding LS estimates is proportional to the squared model output and inversely proportional to the minimum eigenvalues of the normal matrices associated with the LS parameter estimator. Subplot 1 in Figure 5.1 shows that the model output changes dramatically. The magnitude of the model output varies from more than 100 to 5 within 50 iterations because the ARX system is poorly damped. Meanwhile, as shown in Subplot 1 in Figure 5.2, the associated minimum eigenvalues increase almost linearly with a small slope of about 0.54 as the number of data points increases. Consequently, the small minimum eigenvalues cannot prevent the burst of the model output in magnitude from degenerating the TORA and OP estimates, which is shown in Subplots 2 and 3 in Figure 5.1. As we expect, Table 5.1 and Subplots 2 and 3 in Figure 5.1 indicate that the OP parameter estimates are worse than the TORA estimates. This is because of the overparametrization and the two extra parameters that have to be introduced to equalize the numbers of the feedback coefficients and feedforward parameters. The modified TORA shows its remarkable improvement on the TORA in Subplot 4 in Figure 5.1, especially when the iteration number is large. For instance, as shown in Subplots 1 and 4 in figure 5.1, only a very small deviation is caused by the jumping of the output magnitude to more than 60 when the iteration number is close to 500.

The LS order estimator can identify the true model order rapidly in this example. As shown in Table 5.2, there is a likelihood of 0.94 that the LS order estimates are equal to the true model order when only 50 data points are available. For the TORA order estimates, such a likelihood is very small (0.06) when the same data are processed, and gradually goes up to 0.84 when 350 more data points arrive. This is not surprising because the TORA parameter estimation errors are big during the initial time period. As a result of the improvement on the TORA parameter estimates, the modified TORA order estimator has much larger chance of generating the true model order than the TORA order estimator during the initial time period. Table 5.2 also indicates that the OP order estimator cannot consistently identify the true model order for the system in this example.

For an ARX system with order (p_t, q_t), an under-modeled model represents an ARX model of order (p, q) satisfying $p \leq p_t, q \leq q_t$, and $p + q < p_t + q_t$. Such an order is also called the under-modeled order. It comes from Subplots 4–6 in Figure 5.2, the under-modeled order is easily rejected by the LS, TORA, or modified TORA estimators.

Example 2: An unstable ARX system is considered in this example with the following model parameters:

$$\{(3, 1); [0.3, -0.56, -1.078; -1.0, 0.5]; [1.1, -0.7 + j0.7, -0.7 - j0.7; 0.5]\}.$$

The simulation results are presented in Tables 5.3 and 5.4 and Figure 5.3. The OP, TORA, and modified TORA parameter estimates diverge as the model output blows up. This is because the ratio of the squared model output to the minimum eigenvalues of the normal matrices associated with the parameter estimators goes to infinity rather than zero when more and more data arrive. Consequently, none of the OP, TORA, and modified TORA order estimators can perform properly. In this tough situation, the LS parameter/order estimator works well. The very big condition numbers do not cause noticeable troubles to the LS parameter/order estimation numerically in this example. In addition,

iteration number	50	100	200	300	400	500
model parameters	**LS parameter estimates**					
2.400	2.3534	2.3786	2.3895	2.3941	2.3960	2.3973
	(0.0448)	(0.0277)	(0.0196)	(0.0143)	(0.0130)	(0.0121)
1.910	1.8188	1.8687	1.8890	1.8976	1.9020	1.9048
	(0.0885)	(0.0553)	(0.0387)	(0.0277)	(0.0252)	(0.0236)
0.504	0.4585	0.4840	0.4935	0.4975	0.5000	0.5015
	(0.0460)	(0.0286)	(0.0201)	(0.0141)	(0.0129)	(0.0121)
-1.000	-1.0034	-0.9996	-1.0009	-0.9999	-0.9978	-0.9973
	(0.0478)	(0.0291)	(0.0241)	(0.0199)	(0.0186)	(0.0163)
model parameters	**TORA parameter estimates**					
2.400	1.4393	1.6199	1.7695	1.9063	1.7897	1.8207
	(0.3488)	(0.3931)	(0.3507)	(0.3208)	(0.2589)	(0.3525)
1.910	0.3701	0.5760	0.7696	0.9907	0.7710	0.8517
	(0.4724)	(0.6164)	(0.6005)	(0.5759)	(0.4579)	(0.6136)
0.504	-0.1014	-0.0647	-0.0150	0.0718	-0.0329	0.0172
	(0.1687)	(0.2499)	(0.2596)	(0.2611)	(0.2044)	(0.2697)
-1.000	-1.0474	-0.9950	-1.0044	-0.9928	-0.9930	-0.9917
	(0.2684)	(0.1358)	(0.0806)	(0.0502)	(0.0365)	(0.0438)
model parameters	**OP parameter estimates**					
2.400	1.3005	1.4197	1.5439	1.6355	1.5025	1.5692
	(0.2805)	(0.2900)	(0.3084)	(0.3144)	(0.2027)	(0.3233)
1.910	0.1714	0.2496	0.3841	0.5209	0.2785	0.4222
	(0.3355)	(0.4167)	(0.5157)	(0.5458)	(0.3393)	(0.5496)
0.504	-0.1652	-0.1939	-0.1779	-0.1308	-0.2419	-0.1646
	(0.1316)	(0.1647)	(0.2203)	(0.2388)	(0.1435)	(0.2376)
-1.000	-1.1488	-0.9500	-0.9823	-0.9953	-0.9881	-0.9795
	(0.7674)	(0.2506)	(0.1187)	(0.0721)	(0.1070)	(0.0892)
model parameters	**modified TORA parameter estimates**					
2.400	2.3299	2.3606	2.3860	2.3882	2.3945	2.3925
	(0.1308)	(0.0800)	(0.0485)	(0.0363)	(0.0247)	(0.0264)
1.910	1.7787	1.8373	1.8830	1.8861	1.8994	1.8954
	(0.2697)	(0.1608)	(0.0978)	(0.0722)	(0.0499)	(0.0523)
0.504	0.4413	0.4704	0.4910	0.4919	0.4990	0.4968
	(0.1422)	(0.0825)	(0.0502)	(0.0365)	(0.0259)	(0.0265)
-1.000	-1.0794	-0.9931	-1.0049	-0.9938	-0.9956	-0.9929
	(0.2825)	(0.1389)	(0.0798)	(0.0488)	(0.0343)	(0.0392)

Table 5.1: Parameter estimates from 50 replications (Example 1)

the under-modeled order is also easily excluded by the APE criterion from being order estimates, shown in Subplots 6 and 7.

Example 3: In this example, our attention is devoted to an uncontrollable ARX system with the model parameters:

$$\{(3,1); [1.7, 1.4, 0.294; -1.0, -0.3]; [-0.7 + \jmath 0.7, -0.7 - \jmath 0.7, -0.3; -0.3]\}$$

The simulation results are summerized in Tables 5.5 and 5.6 and Figures 5.4–5.7. The model output, shown in Subplot 1 in Figure 5.4, has its largest magnitude of about 20, which is less than one fifth of the largest model output in Example 1. It is shown in Subplots 2 and 3 that the minimum eigenvalues increase almost linearly as the iteration number becomes large. When the order of parameter estimators rises from (2,1) to (3,1), the slope of the minimum eigenvalues decreases dramatically from 2 to 0.06. Consequently, the convergence rate of the LS parameter estimates, generated by the LS estimator of order (3,1), gets considerably slow. More severely, this causes a very big ratio of the squared model output to the minimum eigenvalues to the TORA estimator of order (3,1), and then the performance of the estimator degenerates. It is also because of this that the modified TORA does not bring significant improvement on the TORA estimates.

At the first glance, Table 5.6 looks surprising because the LS order estimates are strongly consistent. Note that the model output comes from the summation of two parts. The first is stimulated by the model input and is not affected by the cancelable poles and zeros. The second part is equal to the output of an AR system excited by the model noise, where the AR part is exactly the same as the AR part of the original ARX system, which contain the cancelable poles. Consequently, the true order of the uncontrollable system could be identified. However, compared with Example 1, the uncontrollability does pretty much reduce the chance of the LS order estimates being equal to the true order during the initial period of 200 iterations.

Example 4: In this example, the controllable subsystem of the ARX system in Example 3 is considered. Consequently, the model parameters of the subsystem are

$$\{(2,0);[1.4,0.98;-1.0];[-0.7+j0.7,-0.7-j0.7]\}$$

The simulation results are presented in Tables 5.7 and 5.8 and Figures 5.8 and 5.9. In this case, all four parameter estimation algorithms work well. During the period of the first 500 iterations, all the order estimators, except the TORA estimator, generate order estimates consistent with the true model order with a very high chance. Subplot 4 in Figure 5.9 and Table 5.8 indicate that the TORA estimator has an increasingly high chance to figure out the true model order as more data arrive.

Example 5: In this example, an ARX system with fast dynamics is considered. The fast dynamics are characterized by a pole very close to the origin. The model parameters are

$$\{(3,0);[1.35,0.91,-0.049;-1.0];[-0.7+j0.7,-0.7-j0.7;0.05]\}$$

The simulation results are illustrated in Tables 5.9 and 5.10 and Figures 5.10 and 5.11. The minimum eigenvalues of the different order look like those in Example 4. The difference of the parameter estimation in this example from that in Example 4 is the order of the parameter estimators. Here the order is (3,0), instead of (2,0). Note that the slope of the minimum eigenvalue function of order (2,0) is about 4 times bigger than that of order (3,0). Thus, the parameter estimators of order (3,0) have about 4 times smaller minimum eigenvalues than those of order (2,0). As a result, the OP and TORA estimates have a large deviation from the corresponding LS parameter estimates when the model output is big in magnitude. Subplot 4 in Figure 5.10 reflects that the modified TORA behaves like the LS estimation algorithm. As evidence, the curves of the APE's of the modified TORA predictors are very close to those of the LS predictors.

A few remarks can be made from the above simulation results:

Remark 5.5 As we see from (5.12), the minimum eigenvalues of normal matrices associated with LS parameter estimators affect the convergence rate. They also have big impact on the performance of the corresponding TORA parameter estimates. This is because the deviation of the TORA estimates from the corresponding LS estimates is proportional to the squared model output and inversely proportional to the minimum eigenvalues of the normal matrices associated with the LS parameter estimator. Therefore, large minimum eigenvalues are preferred.

As shown in (5.24), the minimum eigenvalues are bounded above by the total energy of the model input signals if the effect of the initial gain $P_0(0,0)$ is neglected. This is also true for unstable systems. The order of parameter estimators is a crucial factor for minimum eigenvalues being big or small. Denote by p_t and q_t the number of dominant and uncancelable poles and the number of uncancelable zeros. The parameter estimators of order $(p,q), p \leq p_t$ and $q \leq q_t$, usually have large minimum eigenvalues relative to the total energy of the model input. A significant drop of minimum eigenvalues will be observed if the order of parameter estimators are beyond the bound (p_t, q_t). Therefore, for an ARX system with poles and zeros clustering together, the convergence rate of the parameter estimation for the system is slow. In other words, an identified model in this case has a big uncertainty for a long time period. ∎

Remark 5.6 As observed from Figures 5.1, 5.4, 5.5, 5.8, and 5.10, a big jump of the model output could cause big deviation of TORA parameter estimates from the corresponding LS estimates. This problem can be relieved considerably by using the modified TORA, at a cost of $\mathbf{O}(p^2+q^2)$ extra arithmetic operations. In fact, this problem could be avoided by using pre-filters to reduce the magnitude of model output. ∎

Remark 5.7 The condition number of normal matrices associated with a parameter estimator could reach thousands easily. Therefore, for on-line implementation, numerically robust parameter estimation algorithms, e.g., QR-based algorithms, may be needed. The impact of parameter estimator order on the condition number is not as great as on the minimum eigenvalues. ∎

Remark 5.8 The LS–APE criterion and TORA–APE criterion for order estimation can reject the under-modeled order as order estimates easier than the over-modeled order. In other words, after a short time period, the APE's of the predictors of under-modeled order get larger and larger than the minimum APE. Meanwhile, the APE's of the predictors of over-modeled order remain close to the minimum APE. A very nice property of the LS, OP, TORA, and modified TORA predictors is that after tens of iterations, the order estimates are very close to the true model order if the ARX system in question is stable. Of course, the LS order estimator is the best in performance. The modified TORA order estimator is competitive with the LS estimator. ∎

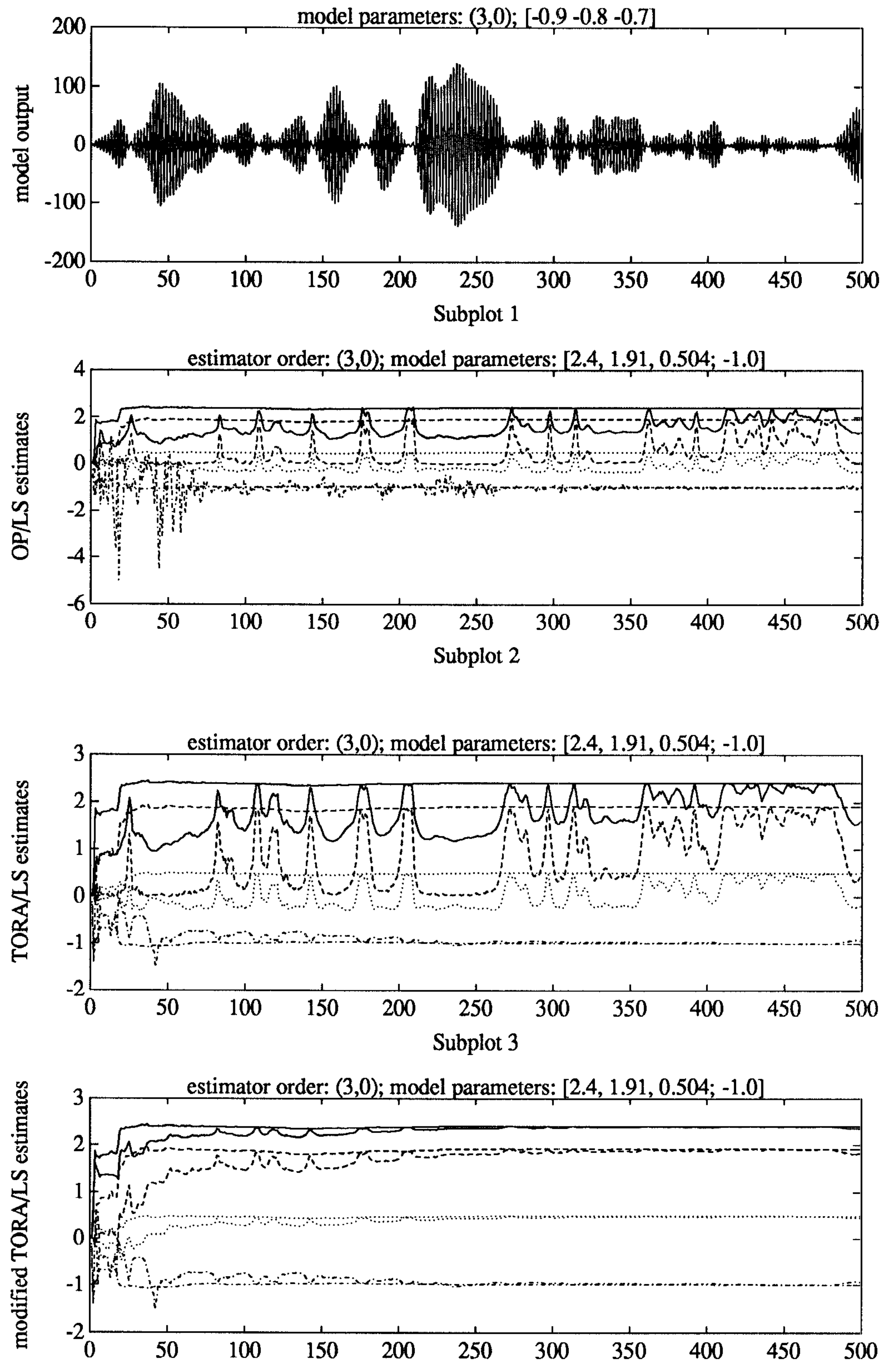


Figure 5.1: Simulation results of Example 1

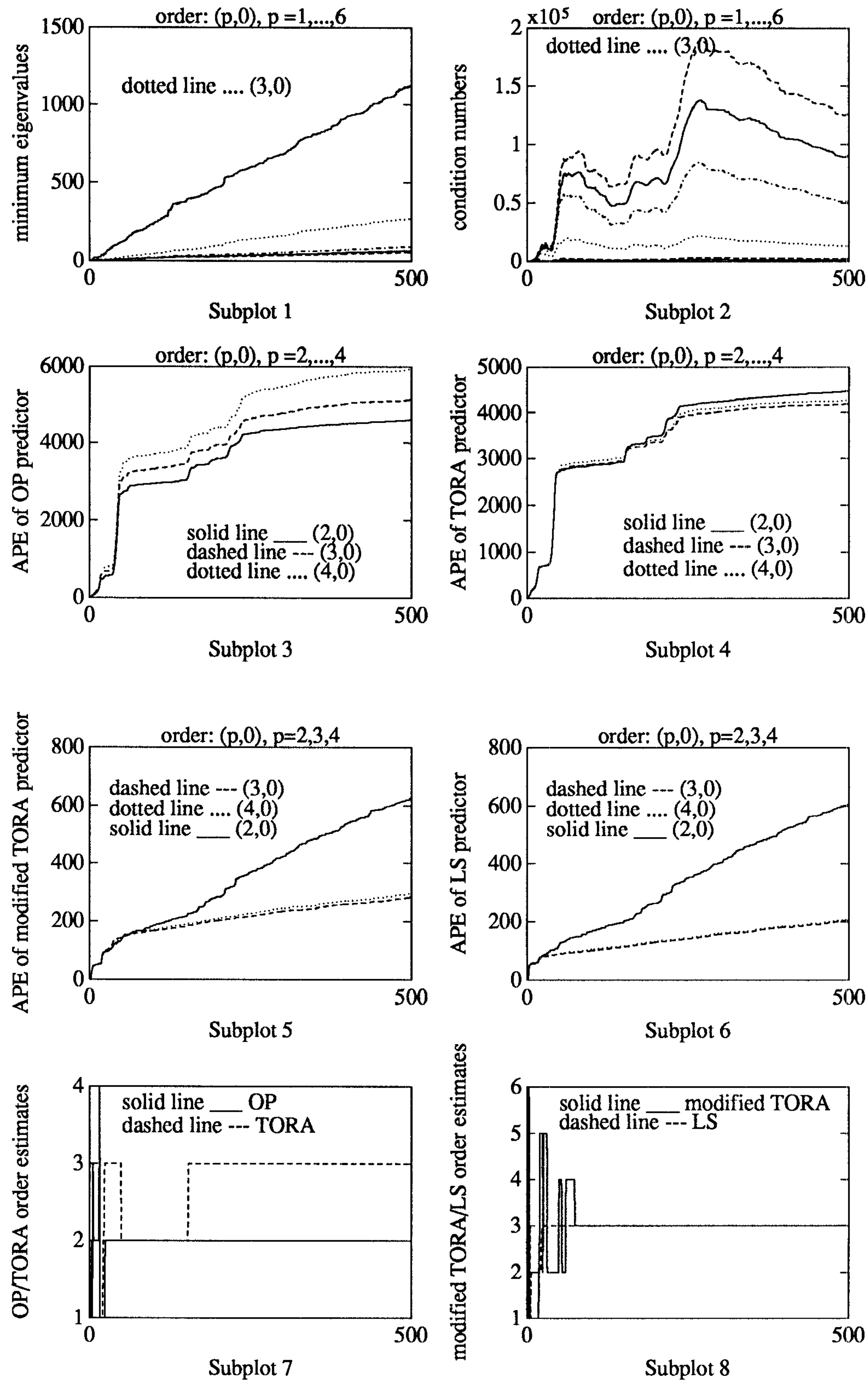


Figure 5.2: Simulation results of Example 1 (Continued)

iteration number	50	100	200	300	400	500
model order	**distribution of LS order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0	0	0	0	0	0
(3,0)	0.94	0.92	0.92	0.94	0.94	0.94
(4,0)	0.06	0.08	0.08	0.06	0.06	0.06
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of TORA order estimates**					
(1,0)	0.06	0	0	0	0	0
(2,0)	0.88	0.68	0.38	0.24	0.16	0.14
(3,0)	0.06	0.32	0.62	0.76	0.84	0.84
(4,0)	0	0	0	0	0	0.02
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of OP order estimates**					
(1,0)	0.54	0.18	0.02	0.02	0	0
(2,0)	0.42	0.80	0.96	0.96	0.98	1.0
(3,0)	0	0	0	0	0	0
(4,0)	0.02	0	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0.02	0.02	0.02	0.02	0.02	0
model order	**distribution of modified TORA order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.30	0.06	0	0	0	0
(3,0)	0.60	0.86	0.92	0.92	0.92	0.92
(4,0)	0.10	0.08	0.08	0.08	0.08	0.08
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0

Table 5.2: The simulation results of order estimates (Example 1)

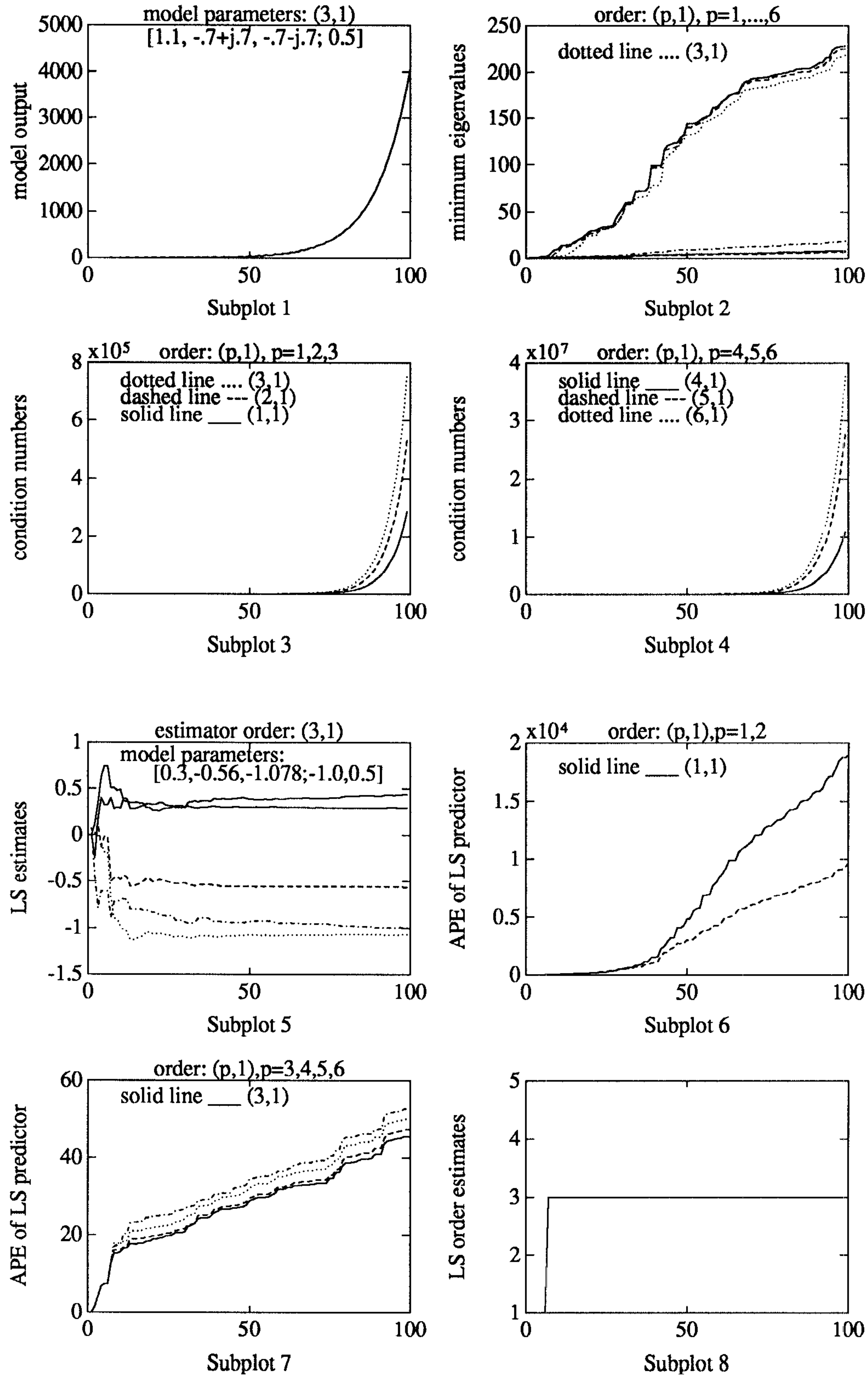


Figure 5.3: Simulation results of Example 2

iteration number	50	100	200
model parameters	**LS parameter estimates**		
0.300	0.2990 (0.0186)	0.2991 (0.0070)	0.2995 (0.0053)
-0.560	-0.5601 (0.0130)	-0.5606 (0.0059)	-0.5600 (0.0037)
-1.078	-1.0721 (0.0177)	-1.0762 (0.0079)	-1.0774 (0.0052)
-1.000	-0.9878 (0.0479)	-0.9969 (0.0386)	-1.0017 (0.0269)
0.500	0.4939 (0.0590)	0.4904 (0.0404)	0.4922 (0.0283)
model parameters	**TORA parameter estimates**		
0.300	-0.3712154762 (0.406662618)	-0.9000696574 (0.020418395)	-0.9083669038 (0.003185312)
-0.560	-0.2319899347 (0.119251557)	-0.0065345252 (0.019908248)	0.0000001358 (0.000001767)
-1.078	-0.2518080044 (0.256476240)	0.0003026850 (0.004668810)	-0.0000001461 (0.000000855)
-1.000	-0.7789792101 (0.659824237)	-4.1848846744 (21.033032455)	78727.4793546076 (144065.538078134)
0.500	0.8943099163 (0.964790229)	13.4759970628 (47.919303239)	-63459.7022198779 (451272.430787421)
model parameters	**OP parameter estimates**		
0.300	-0.3910163212 (0.430930697)	-0.9044064402 (0.022241440)	-0.9063474523 (0.006035854)
-0.560	-0.1871566446 (0.154158916)	-0.0023923340 (0.022138957)	-0.0035371019 (0.006401347)
-1.078	-0.2627135951 (0.253884520)	-0.0023835914 (0.009965174)	0.0009260508 (0.005117898)
-1.000	-1.1029889526 (0.870026613)	9.7369349568 (41.657204188)	-53998.2458504089 (400916.787435837)
0.500	1.0009474696 (1.084341006)	-0.5345889069 (36.108459765)	-18096.3174297517 (337849.021139464)
model parameters	**modified TORA parameter estimates**		
0.300	0.3131445391 (0.071832654)	0.3020065000 (0.151914434)	-306.8589103307 (1314.194793270)
-0.560	-0.5349927379 (0.084229397)	-0.6084868085 (0.108820727)	-167.8969359000 (900.949953930)
-1.078	-1.0481421104 (0.072038314)	-1.0225369070 (0.222299667)	554.6584854369 (2050.827372189)
-1.000	-0.7601714620 (0.607390115)	-4.9328225979 (22.979217262)	77259.4593163985 (145758.700379236)
0.500	1.1129175056 (0.673919600)	-0.4604518222 (25.216176769)	58244.9935702807 (151054.525066415)

Table 5.3: The simulation results of parameter estimates (Example 2)

iteration number	50	100	200
model order	**distribution of LS order estimates**		
(1,1)	0	0	0
(2,1)	0	0	0
(3,1)	0.98	1.00	1.0000
(4,1)	0	0	0
(5,1)	0	0	0
(6,1)	0.02	0	0
model order	**distribution of OP order estimates**		
(1,1)	0	0.40	0.92
(2,1)	0	0.30	0.04
(3,1)	0	0.04	0.04
(4,1)	0.16	0.08	0
(5,1)	0.36	0.12	0
(6,1)	0.48	0.06	0
model order	**distribution of TORA order estimates**		
(1,1)	0	0.08	0.16
(2,1)	0	0.16	0.22
(3,1)	0.06	0.02	0.02
(4,1)	0.24	0.10	0.04
(5,1)	0.14	0.08	0
(6,1)	0.56	0.56	0.56
model order	**distribution of modified TORA order estimates**		
(1,1)	0.04	0.14	0.60
(2,1)	0.04	0	0.10
(3,1)	0.90	0.74	0.16
(4,1)	0.02	0.04	0.08
(5,1)	0	0	0.02
(6,1)	0	0.08	0.04

Table 5.4: The simulation results of order estimates (Example 2)

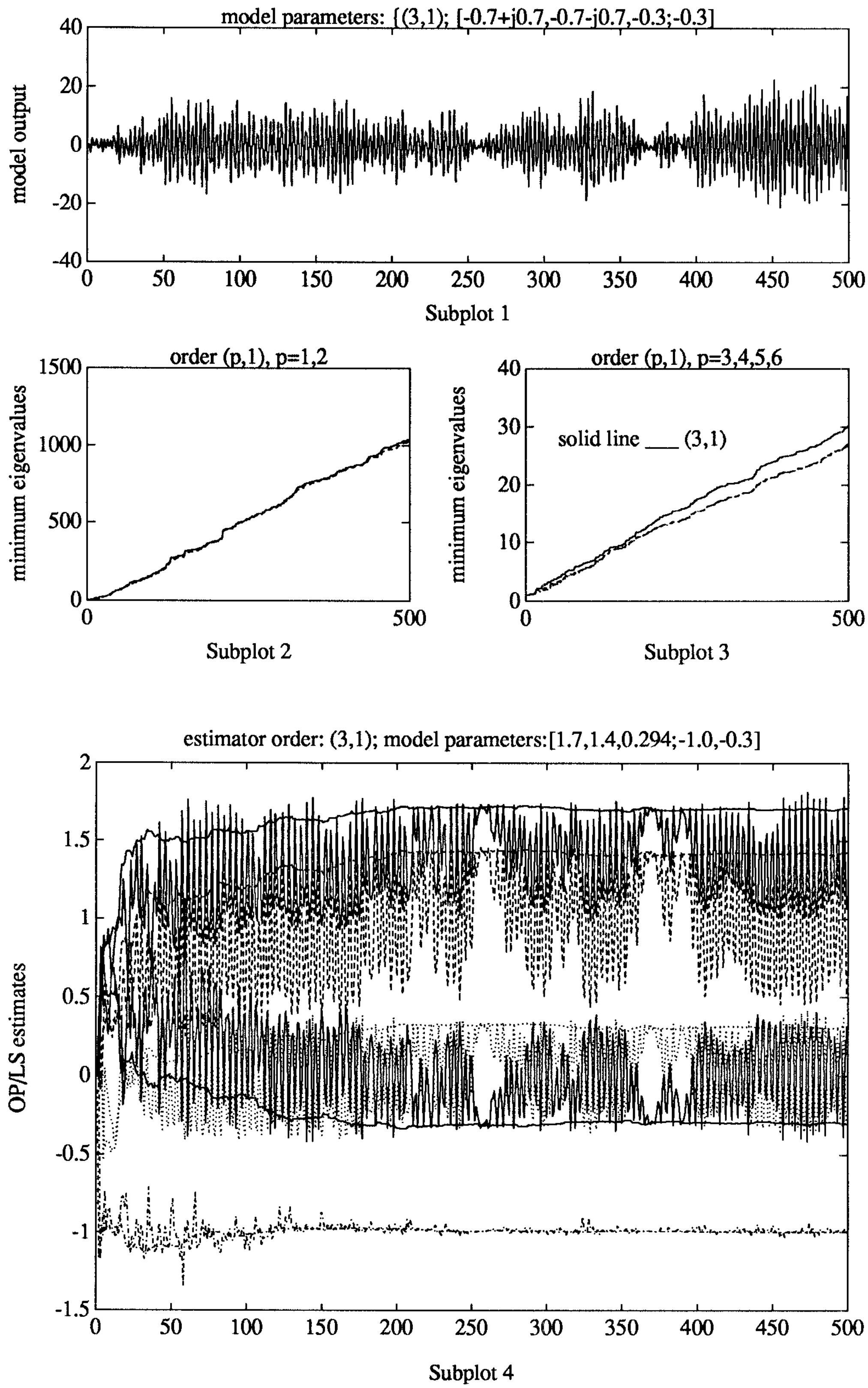
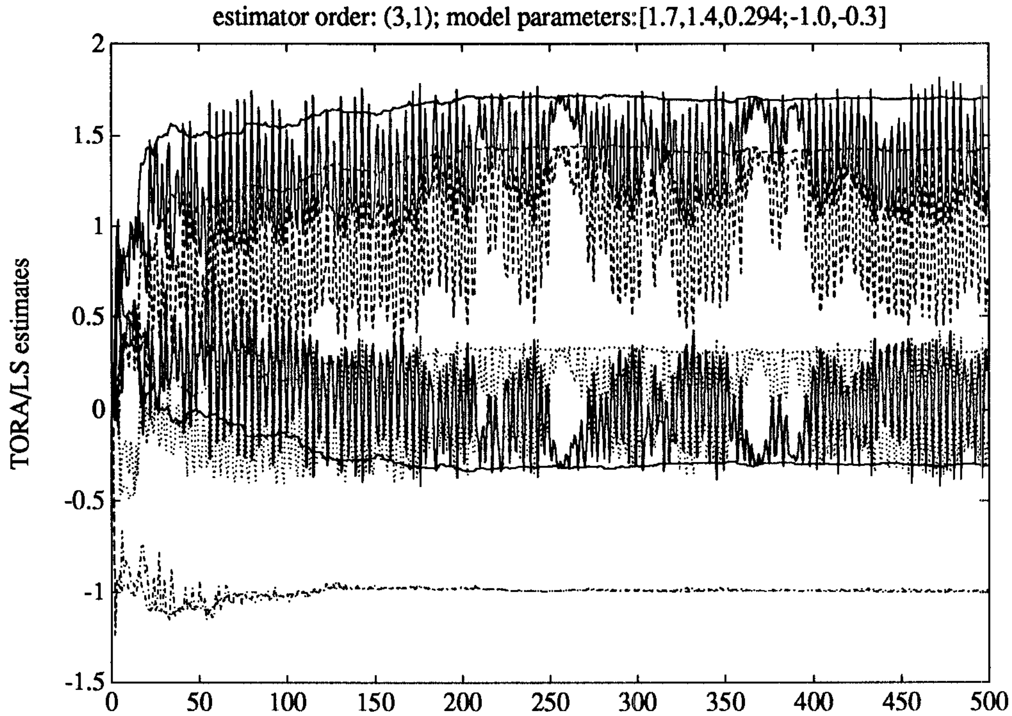
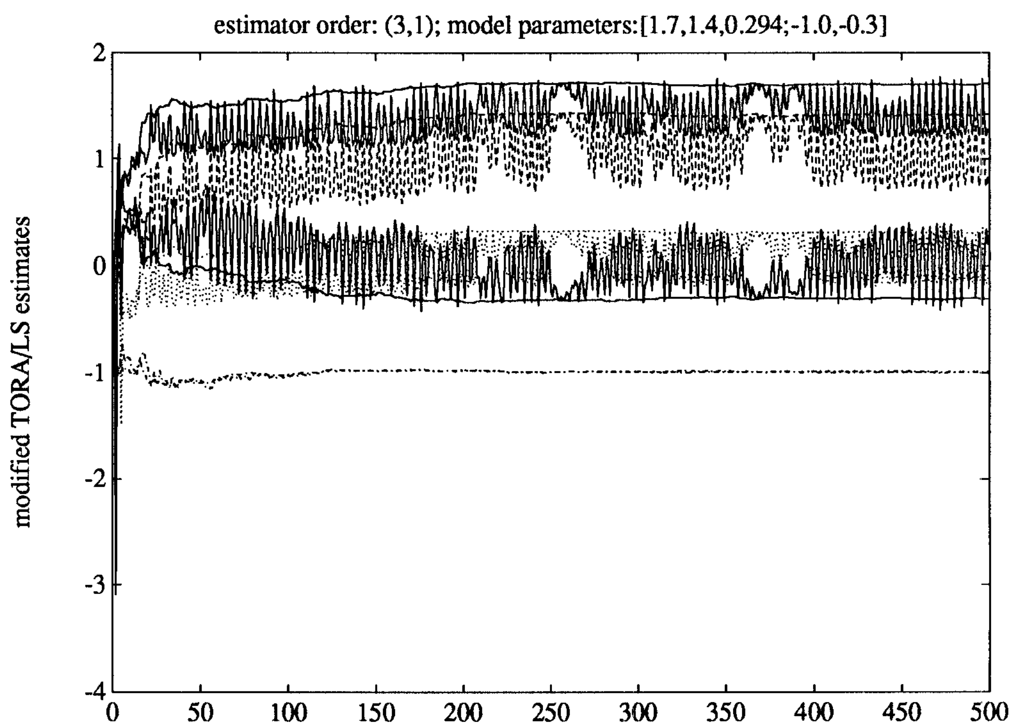


Figure 5.4: Simulation results of Example 3



Subplot 1



Subplot 2

Figure 5.5: Simulation results of Example 3 (Continued)

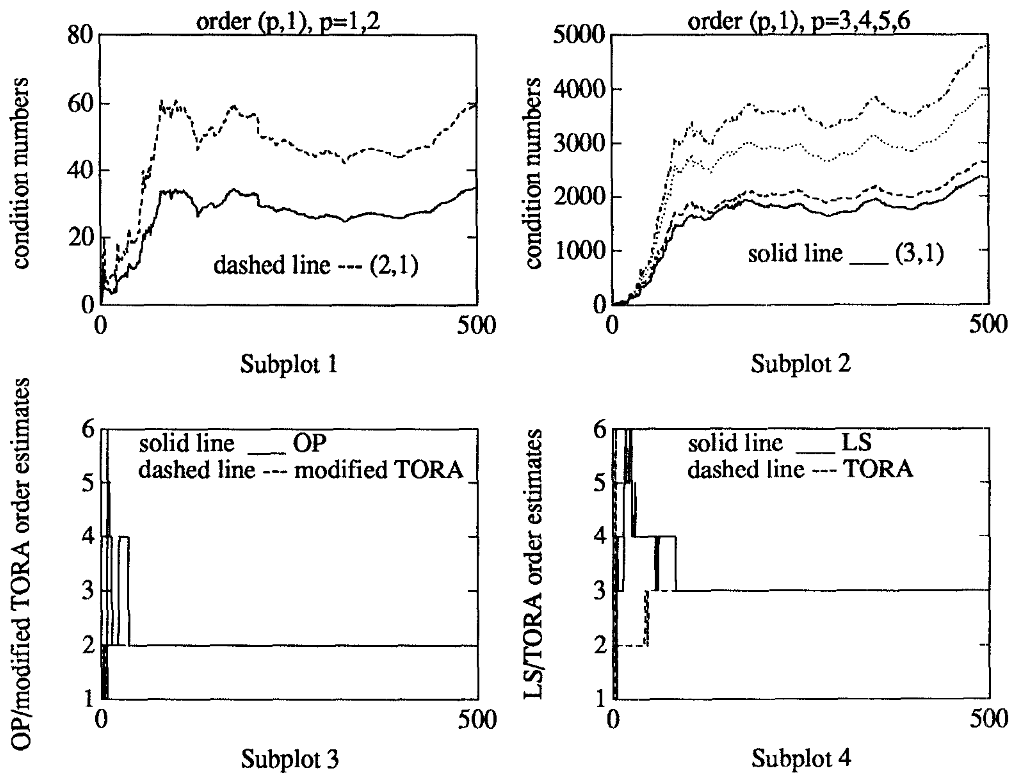


Figure 5.6: Simulation results of Example 3 (Continued)

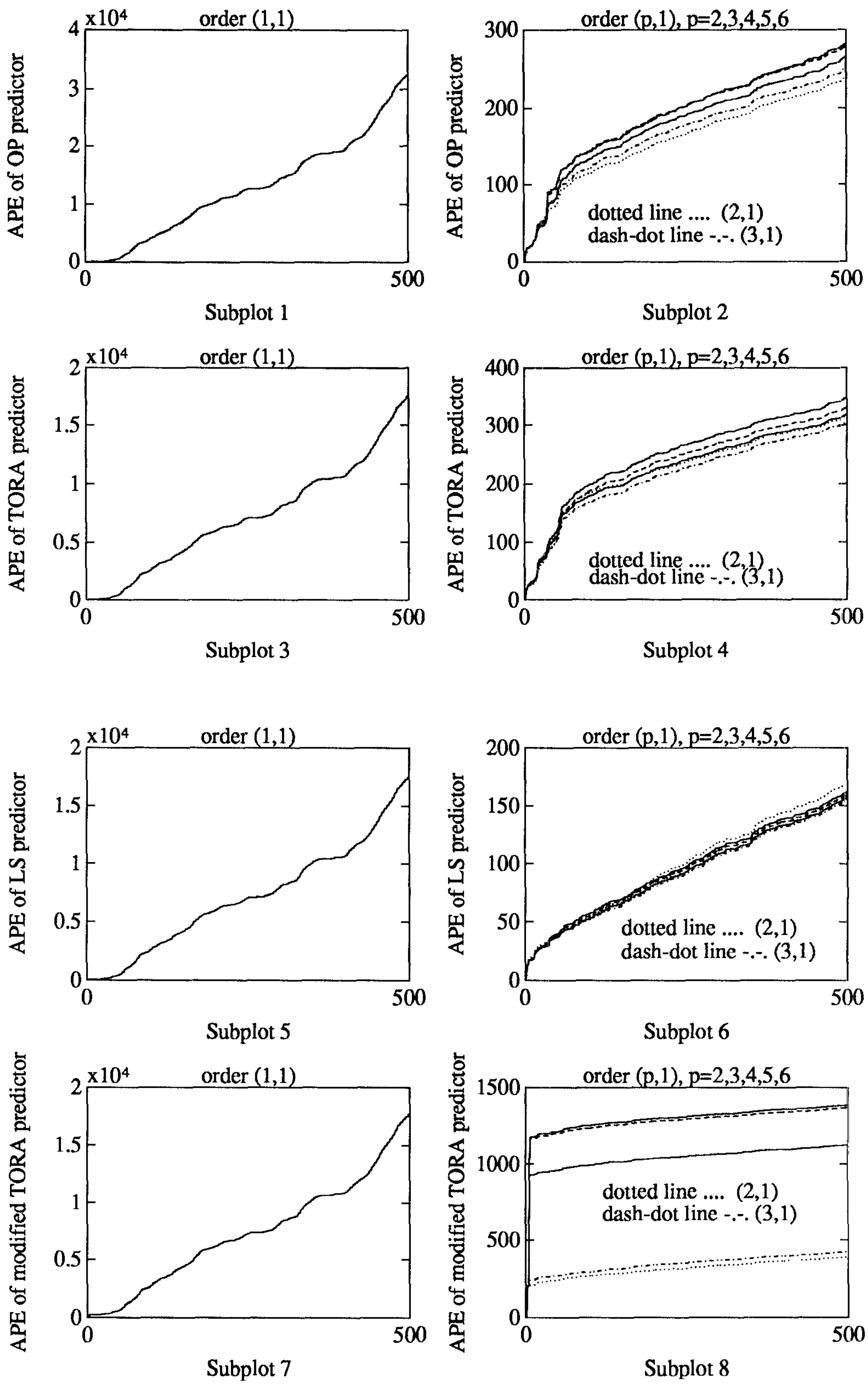


Figure 5.7: Simulation results of Example 3 (Continued)

iteration number	50	100	200	300	400	500
model parameters	**LS parameter estimates**					
1.7	1.4181 (0.1054)	1.5409 (0.0890)	1.6290 (0.0671)	1.6536 (0.0532)	1.6689 (0.0471)	1.6741 (0.0394)
1.4	1.0040 (0.1498)	1.1758 (0.1245)	1.3002 (0.0949)	1.3351 (0.0742)	1.3573 (0.0653)	1.3645 (0.0552)
0.294	0.0198 (0.1052)	0.1390 (0.0876)	0.2261 (0.0662)	0.2494 (0.0523)	0.2648 (0.0456)	0.2695 (0.0385)
-1.0	-1.0069 (0.0506)	-1.0006 (0.0337)	-1.0030 (0.0239)	-1.0007 (0.0203)	-0.9985 (0.0180)	-0.9979 (0.0154)
-0.3	-0.0048 (0.1076)	-0.1322 (0.0967)	-0.2231 (0.0743)	-0.2504 (0.0599)	-0.2664 (0.0502)	-0.2707 (0.0402)
model parameters	**TORA parameter estimates**					
1.7	1.1663 (0.2541)	1.2248 (0.2809)	1.3280 (0.2847)	1.3531 (0.2331)	1.3157 (0.2363)	1.3615 (0.2262)
1.4	0.6835 (0.3193)	0.7498 (0.3702)	0.8838 (0.3920)	0.9165 (0.3232)	0.8652 (0.3276)	0.9280 (0.3148)
0.294	-0.1505 (0.2032)	-0.1310 (0.2486)	-0.0546 (0.2700)	-0.0369 (0.2230)	-0.0745 (0.2270)	-0.0328 (0.2186)
-1.0	-1.0329 (0.1353)	-1.0127 (0.0492)	-1.0060 (0.0317)	-1.0029 (0.0256)	-0.9985 (0.0180)	-0.9982 (0.0167)
-0.3	0.2185 (0.2973)	0.1687 (0.2957)	0.0752 (0.2906)	0.0489 (0.2388)	0.0790 (0.2272)	0.0415 (0.2256)
model parameters	**OP parameter estimates**					
1.7	1.2126 (0.2845)	1.2203 (0.2682)	1.2880 (0.2591)	1.3448 (0.2456)	1.3453 (0.2585)	1.3849 (0.2359)
1.4	0.7376 (0.3751)	0.7439 (0.3536)	0.8293 (0.3540)	0.9043 (0.3420)	0.9071 (0.3583)	0.9603 (0.3288)
0.294	-0.1155 (0.2604)	-0.1354 (0.2409)	-0.0925 (0.2439)	-0.0454 (0.2383)	-0.0447 (0.2468)	-0.0102 (0.2286)
-1.0	-1.0069 (0.1336)	-0.9928 (0.0977)	-0.9970 (0.0407)	-1.0021 (0.0343)	-0.9922 (0.0331)	-0.9976 (0.0236)
-0.3	0.1689 (0.2988)	0.1747 (0.2995)	0.1171 (0.2754)	0.0503 (0.2422)	0.0577 (0.2648)	0.0122 (0.2314)
model parameters	**modified TORA parameter estimates**					
1.7	1.3117 (0.1303)	1.3725 (0.1492)	1.4525 (0.1882)	1.4478 (0.1733)	1.4006 (0.1393)	1.4261 (0.1760)
1.4	0.8630 (0.1781)	0.9470 (0.2001)	1.0555 (0.2607)	1.0479 (0.2413)	0.9827 (0.1942)	1.0176 (0.2450)
0.294	-0.0740 (0.1179)	-0.0220 (0.1388)	0.0560 (0.1810)	0.0482 (0.1682)	0.0024 (0.1343)	0.0266 (0.1707)
-1.0	-1.0298 (0.0990)	-1.0159 (0.0432)	-1.0049 (0.0303)	-1.0031 (0.0245)	-0.9991 (0.0175)	-0.9977 (0.0156)
-0.3	0.3083 (0.3293)	0.2056 (0.2964)	0.0560 (0.3214)	0.0231 (0.2508)	0.0775 (0.2162)	0.0346 (0.2339)

Table 5.5: The simulation results of parameter estimates (Example 3)

iteration number	50	100	200	300	400	500
model order	**distribution of LS order estimates**					
(1,1)	0	0	0	0	0	0
(2,1)	0.58	0.44	0.14	0.02	0	0
(3,1)	0.38	0.54	0.80	0.92	0.98	1.0
(4,1)	0.04	0.02	0.04	0.04	0.02	0
(5,1)	0	0	0.02	0.02	0	0
(6,1)	0	0	0	0	0	0
model order	**distribution of TORA order estimates**					
(1,1)	0	0	0	0	0	0
(2,1)	0.04	0.02	0	0	0	0.04
(3,1)	0.74	0.80	0.86	0.90	0.94	0.90
(4,1)	0.20	0.18	0.14	0.10	0.06	0.06
(5,1)	0.02	0	0	0	0	0
(6,1)	0	0	0	0	0	0
model order	**distribution of OP order estimates**					
(1,1)	0	0	0	0	0	0
(2,1)	0.78	0.78	0.82	0.88	0.90	0.90
(3,1)	0.08	0.14	0.12	0.08	0.08	0.08
(4,1)	0.14	0.08	0.06	0.04	0.02	0.02
(5,1)	0	0	0	0	0	0
(6,1)	0	0	0	0	0	0
model order	**distribution of modified TORA order estimates**					
(1,1)	0.22	0.14	0.02	0.02	0	0
(2,1)	0.66	0.78	0.94	0.98	1.00	1.00
(3,1)	0.08	0.08	0.04	0	0	0
(4,1)	0.04	0	0	0	0	0
(5,1)	0	0	0	0	0	0
(6,1)	0	0	0	0	0	0

Table 5.6: The simulation results of order estimates (Example 3)

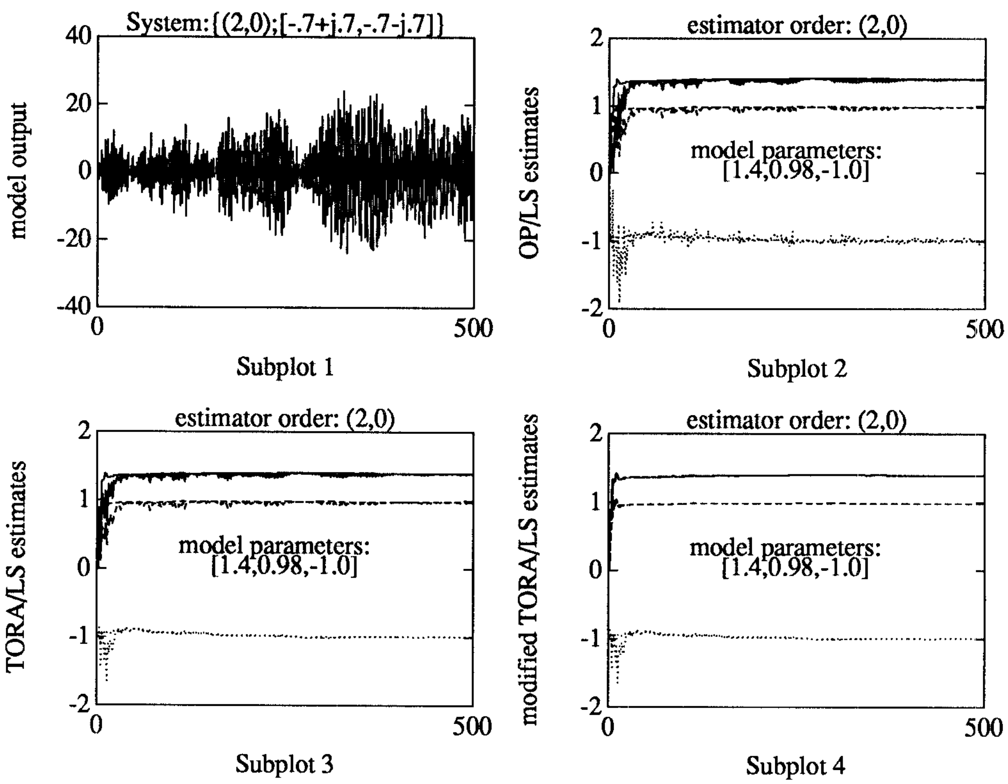


Figure 5.8: Simulation results of Example 4

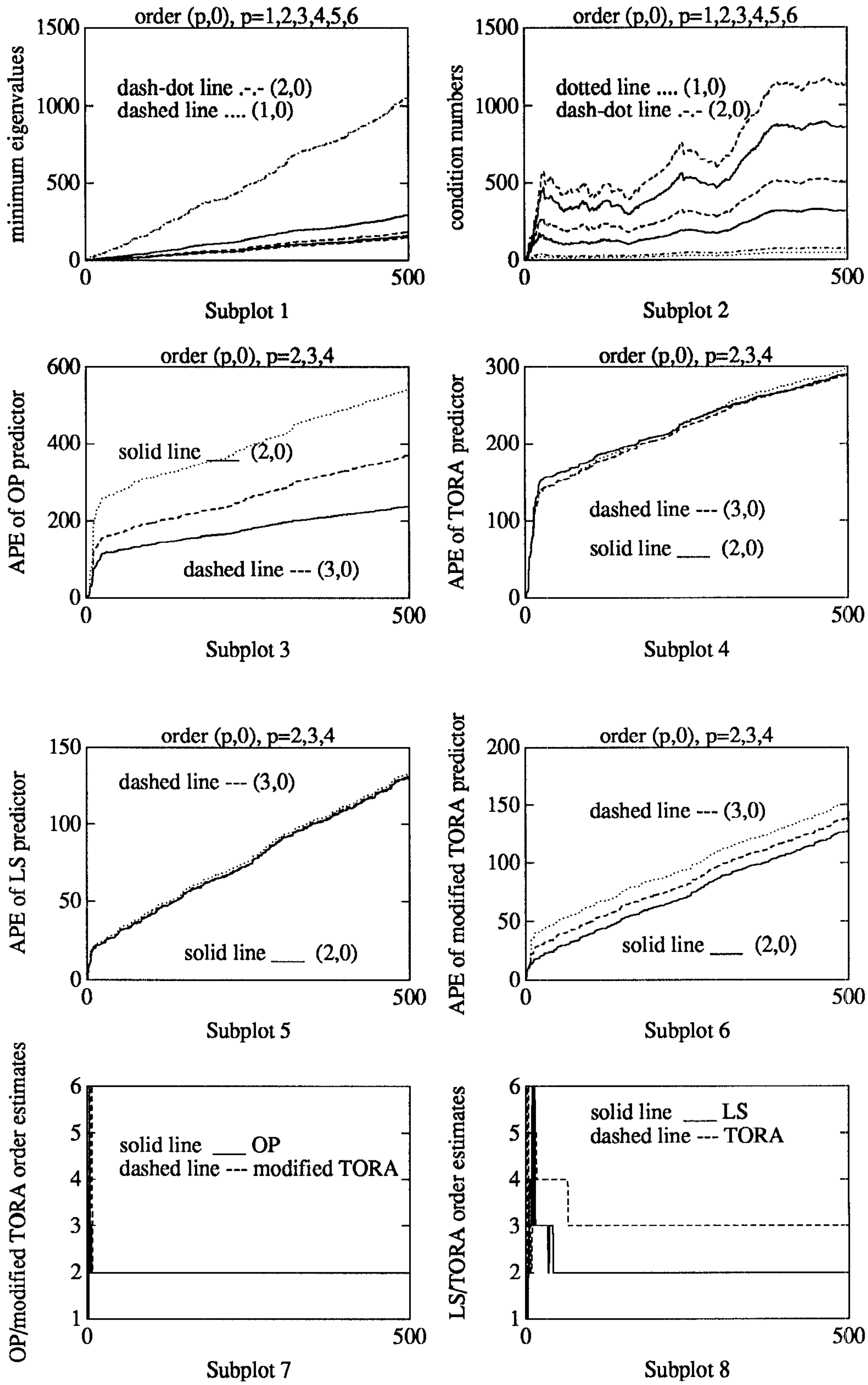


Figure 5.9: Simulation results of Example 4 (Continued)

iteration number	50	100	200	300	400	500
model parameters	**LS parameter estimates**					
1.4	1.3945	1.3976	1.3978	1.3990	1.3987	1.3993
	(0.0223)	(0.0112)	(0.0079)	(0.0051)	(0.0041)	(0.0036)
0.98	0.9740	0.9777	0.9772	0.9794	0.9796	0.9798
	(0.0249)	(0.0134)	(0.0078)	(0.0043)	(0.0035)	(0.0032)
-1.0	-1.0031	-1.0022	-1.0012	-1.0014	-0.9997	-0.9978
	(0.0443)	(0.0313)	(0.0229)	(0.0185)	(0.0171)	(0.0148)
model parameters	**TORA parameter estimates**					
1.4	1.3052	1.3425	1.3800	1.3877	1.3881	1.3931
	(0.0897)	(0.0596)	(0.0184)	(0.0125)	(0.0149)	(0.0076)
0.98	0.8829	0.9201	0.9583	0.9667	0.9687	0.9732
	(0.0818)	(0.0535)	(0.0179)	(0.0120)	(0.0146)	(0.0069)
-1.0	-1.0302	-1.0096	-1.0032	-1.0012	-1.0000	-0.9977
	(0.1574)	(0.0471)	(0.0288)	(0.0201)	(0.0165)	(0.0148)
model parameters	**OP parameter estimates**					
1.4	1.3103	1.3440	1.3752	1.3859	1.3862	1.3917
	(0.0673)	(0.0555)	(0.0233)	(0.0124)	(0.0169)	(0.0088)
0.98	0.8853	0.9215	0.9549	0.9655	0.9675	0.9722
	(0.0633)	(0.0493)	(0.0217)	(0.0124)	(0.0160)	(0.0081)
-1.0	-1.0083	-0.9934	-0.9984	-1.0009	-0.9934	-0.9968
	(0.1162)	(0.0899)	(0.0396)	(0.0308)	(0.0299)	(0.0206)
model parameters	**modified TORA parameter estimates**					
1.4	1.3911	1.3969	1.3975	1.3989	1.3987	1.3993
	(0.0227)	(0.0115)	(0.0078)	(0.0051)	(0.0041)	(0.0036)
0.98	0.9701	0.9768	0.9768	0.9792	0.9795	0.9798
	(0.0247)	(0.0135)	(0.0079)	(0.0042)	(0.0035)	(0.0032)
-1.0	-1.0302	-1.0096	-1.0032	-1.0012	-1.0000	-0.9977
	(0.1574)	(0.0471)	(0.0288)	(0.0201)	(0.0165)	(0.0148)

Table 5.7: The simulation results of parameter estimates (Example 4)

iteration number	50	100	200	300	400	500
model order	**distribution of LS order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.86	0.94	0.98	1.00	.98	0.98
(3,0)	0.14	0.06	0.02	0	0.02	0.02
(4,0)	0	0	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of TORA order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.18	0.24	0.22	0.30	0.24	0.32
(3,0)	0.68	0.72	0.74	0.66	0.76	0.68
(4,0)	0.14	0.04	0.04	0.04	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of OP order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.92	0.98	0.98	0.98	0.98	0.98
(3,0)	0.04	0.02	0.02	0.02	0.02	0.02
(4,0)	0.04	0	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of modified TORA order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.94	0.92	0.94	0.94	0.94	0.94
(3,0)	0.06	0.08	0.06	0.06	0.06	0.06
(4,0)	0	0	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0

Table 5.8: The simulation results of order estimates (Example 4)

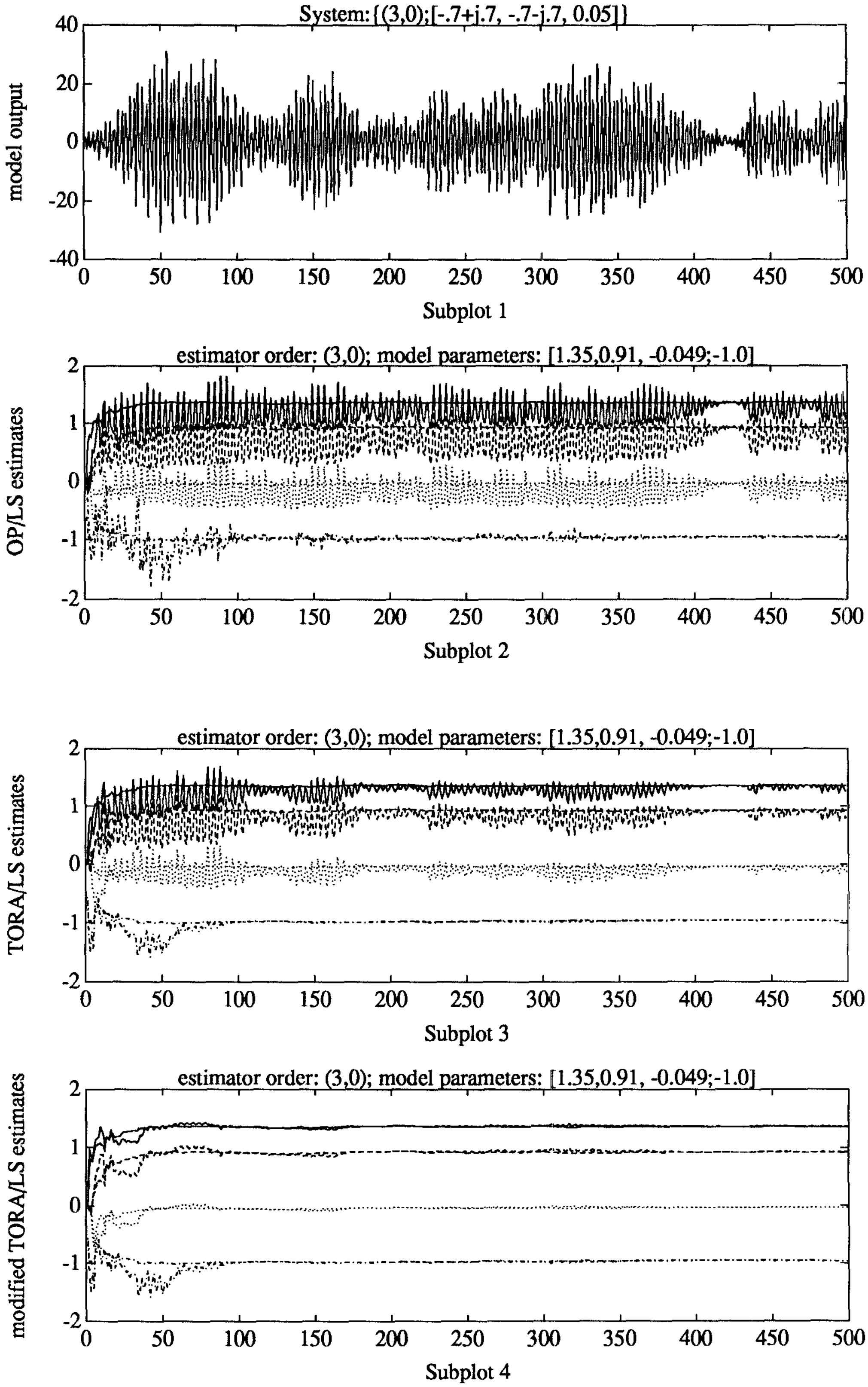


Figure 5.10: Simulation results of Example 5

5.5 Summary

In this chapter a fast order and parameter estimation algorithm is proposed for MIMO ARX systems. This algorithm proceeds by minimizing the accumulated prediction error over a family of ARX models of different order. The family of ARX models is obtained by using the time and order recursive algorithm. The simultaneous estimation of ARX system order and parameters generated by the proposed algorithm is strongly consistent if the noise of an ARX system is a martingale difference process with respect to an increasing family of σ-fields. The consistency is proved after carefully reviewing previous results on martingale analysis of linear least-squares modeling. The proposed fast algorithm could become an effective and practical tool for linear modeling.

Also, a systematic simulation study of parameter and order estimation for ARX systems has benn done. This can be regarded as a complement of Hemmerly and Davis' paper[72]. The analysis of the internal variables involved in parameter and order estimation like the minimum eigenvalues and condition number of normal matrices could help understand the interaction between the controller and identifier in an adaptive control system.

The proposed algorithm needs further study. Lattice forms of the algorithm may fit order estimation better than the current version because we just need prediction errors during computing the accumulated prediction error. In addition, the QR-decomposition technique needs to be introduced to make the proposed algorithm better conditioned.

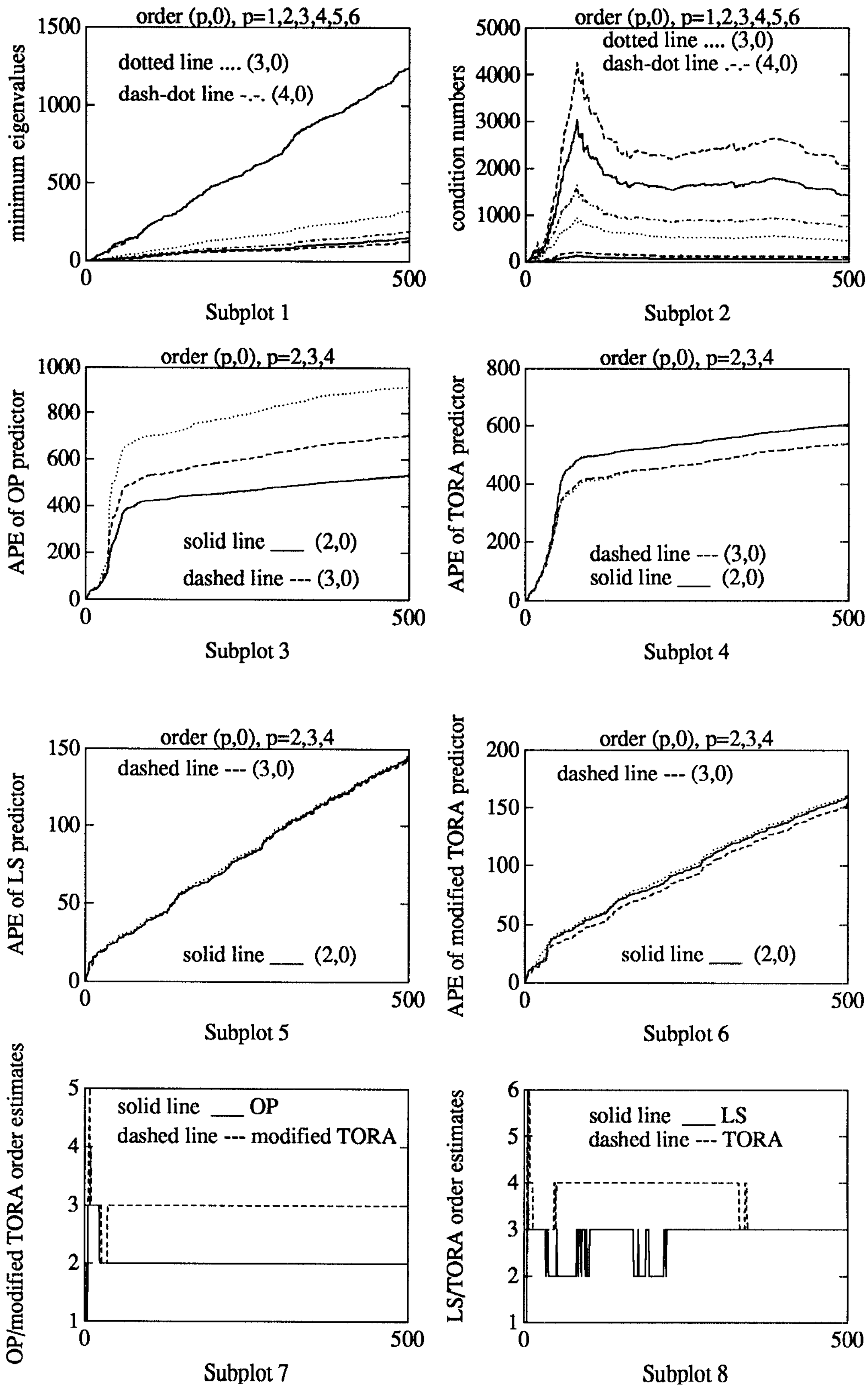


Figure 5.11: Simulation results of Example 5 (Continued)

iteration number	50	100	200	300	400	500
model parameters	**LS parameter estimates**					
1.35	1.3316	1.3448	1.3494	1.3494	1.3499	1.3511
	(0.0483)	(0.0308)	(0.0217)	(0.0162)	(0.0153)	(0.0154)
0.91	0.8862	0.9024	0.9093	0.9095	0.9110	0.9124
	(0.0641)	(0.0426)	(0.0299)	(0.0226)	(0.0211)	(0.0217)
-0.049	-0.0625	-0.0529	-0.0477	-0.0486	-0.0477	-0.0470
	(0.0423)	(0.0280)	(0.0209)	(0.0163)	(0.0154)	(0.0153)
-1.0	-1.0029	-0.9993	-1.0026	-1.0004	-0.9984	-0.9978
	(0.0460)	(0.0321)	(0.0233)	(0.0198)	(0.0176)	(0.0153)
model parameters	**TORA parameter estimates**					
1.35	1.2081	1.2346	1.2746	1.2977	1.2866	1.3107
	(0.1738)	(0.1487)	(0.1177)	(0.0752)	(0.0831)	(0.0585)
0.91	0.7386	0.7603	0.8086	0.8392	0.8242	0.8569
	(0.2115)	(0.1899)	(0.1610)	(0.1035)	(0.1146)	(0.0808)
-0.049	-0.1160	-0.1232	-0.1079	-0.0913	-0.1036	-0.0826
	(0.1273)	(0.1219)	(0.1116)	(0.0700)	(0.0783)	(0.0552)
-1.0	-1.0186	-1.0075	-1.0054	-1.0005	-0.9979	-0.9978
	(0.1163)	(0.0442)	(0.0302)	(0.0213)	(0.0175)	(0.0154)
model parameters	**OP parameter estimates**					
1.35	1.1413	1.1418	1.1803	1.2152	1.2073	1.213
	(0.2518)	(0.2240)	(0.2091)	(0.2014)	(0.1943)	(0.1603)
0.91	0.6437	0.6385	0.6806	0.7247	0.7151	0.7221
	(0.3285)	(0.2901)	(0.2840)	(0.2791)	(0.2678)	(0.2222)
-0.049	-0.1810	-0.2082	-0.1964	-0.1712	-0.1793	-0.1768
	(0.2312)	(0.1992)	(0.1953)	(0.1953)	(0.1843)	(0.1541)
-1.0	-0.9991	-0.9892	-0.9964	-1.0017	-0.9923	-0.9978
	(0.1238)	(0.0930)	(0.0385)	(0.0297)	(0.0309)	(0.0222)
model parameters	**modified TORA parameter estimates**					
1.35	1.3246	1.3455	1.3505	1.3485	1.3487	1.3509
	(0.0677)	(0.0505)	(0.0336)	(0.0180)	(0.0164)	(0.0152)
0.91	0.8763	0.9034	0.9110	0.9082	0.9093	0.9121
	(0.0974)	(0.0724)	(0.0468)	(0.0257)	(0.0230)	(0.0214)
-0.049	-0.0682	-0.0518	-0.0463	-0.0494	-0.0489	-0.0472
	(0.0675)	(0.0493)	(0.0326)	(0.0184)	(0.0166)	(0.0151)
-1.0	-1.0186	-1.0075	-1.0054	-1.0005	-0.9979	-0.9978
	(0.1152)	(0.0438)	(0.0299)	(0.0211)	(0.0173)	(0.0153)

Table 5.9: The simulation results of parameter estimates (Example 5)

iteration number	50	100	200	300	400	500
model order	**distribution of LS order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.78	0.74	0.64	0.64	0.56	0.46
(3,0)	0.22	0.26	0.36	0.36	0.44	0.54
(4,0)	0	0	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of TORA order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.24	0.20	0.18	0.14	0.22	0.18
(3,0)	0.64	0.72	0.76	0.82	0.76	0.80
(4,0)	0.12	0.08	0.06	0.04	0.02	0.02
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of OP order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.90	0.94	0.98	0.98	0.98	0.98
(3,0)	0.04	0.04	0.02	0.02	0.02	0.02
(4,0)	06	0.02	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0
model order	**distribution of modified TORA order estimates**					
(1,0)	0	0	0	0	0	0
(2,0)	0.94	0.92	0.92	0.88	0.90	0.84
(3,0)	0.06	0.08	0.08	0.12	0.10	0.16
(4,0)	0	0	0	0	0	0
(5,0)	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0

Table 5.10: The simulation results of order estimates (Example 5)

Chapter 6

Robust Stability of LS ARX Models/IIR Filters

In this chapter, the stability issue for least-squares (LS) ARX models and adaptive IIR filters is studied. Firstly, a sufficient condition for asymptotic stability is developed for general LMMSE ARX models. Secondly, it is shown that an adaptive all-pole filter with parameters updated by using the Levinson-Durbin algorithm [66] has all its poles within the unit circle uniformly if the system generating the data satisfies a persistent excitation condition. We will call this the instantaneous stability to emphasize that time-varying systems having this property need not be stable in the sense of BIBO stability. This distinction is often missed in the signal processing literature [134]. Then an algorithm is proposed for general adaptive IIR filters to maintain their instantaneous stability. The complexity of the proposed algorithm is acceptable in practice. In addition, all the stability results are *robust* in the sense that the true model structure of the plant which generates the data needs not be known.

6.1 Preliminaries

Consider two bounded data sequences $\{\mathbf{y}_t\}_{t=0}^{n}$ and $\{\mathbf{u}_t\}_{t=0}^{n}$, which are realizations of zero-mean processes $\mathbf{y}$ and $\mathbf{u}$. The processes are related by an unknown discrete-time system with disturbances. A basic step in the analysis and use of the data is the determination of a model for the system. The most popular approach for determination of a system model is linear modeling because many theories and tools are available. However, practical systems usually have some nonlinearity and/or some high frequency noise. Therefore, for real situations, only an approximate model can or should be obtained. The approximate modeling framework is receiving increasingly wide attention. A model-searching approach has been suggested in Chapter 5. This approach is composed of two steps: (1) determining a family of least squares (LS) ARX models (for autoregressive models with exogenous input) of different order (p, q). The LS ARX models are described as

$$\hat{A}_n(z^{-1})\mathbf{y}_n = \hat{C}_n(z^{-1})\mathbf{u}_n + \mathbf{e}_n \tag{6.1}$$

where $\mathbf{e}_n$ is the output prediction error.

$$\hat{A}_n(z^{-1}) = I + \hat{A}_{n,1}z^{-1} + \cdots + \hat{A}_{n,p}z^{-p}$$

and

$$\hat{C}_n(z^{-1}) = \hat{C}_{n,0} + \hat{C}_{n,1}z^{-1} + \cdots + \hat{C}_{n,q}z^{-q}$$

are matrix polynomials in the backward shift operator z^{-1}. They are determined by solving the following minimization problem:

$$\min_{\hat{A}_{n,i},\hat{C}_{n,j}} \{\sum_{t=0}^{n} \|\hat{A}_n(z^{-1})\mathbf{y}_t - \hat{C}_n(z^{-1})\mathbf{u}_t\|^2\} \tag{6.2}$$

(2) selecting the best model according to some criterion for evaluating the resulting models. The parameters of the LS ARX models within the family can be well approximated by the TORA estimates which can be generated efficiently via the time and order recursive algorithm.

LS ARX models represent a class of linear models of control systems which have been extensively used for adaptive control and other purposes. In the areas of communications and signal processing, LS ARX models are also known as a large class of (adaptive) IIR filters, which have increasingly wide use. In the use of LS ARX models, an important issue is their stability. For instance, in geophysical signal processing[127], the LS ARX models are often used as a non-adaptive or time-invariant model obtained by processing *long* sequences $\{\mathbf{y}_t\}_{t=0}^{N}$ and $\{\mathbf{u}_t\}_{t=0}^{N}$. For this kind of applications, we need to consider asymptotic properties of LS ARX models. When the measured data $\{\mathbf{y}_n\}_{n=0}^{\infty}$ and $\{\mathbf{u}_n\}_{n=0}^{\infty}$ are a realization of a jointly stationary and mean ergodic process $(\mathbf{y}, \mathbf{u})$ with zero-mean, the corresponding LS ARX model converges to an LMMSE ARX model in the mean square sense as the number of data points approaches infinity. Therefore, we need investigate if all poles of the LMMSE model are within the unit circle. In model predictive adaptive control[34][35][36] and in speech synthesis[78], the LS ARX models are used to predict the future behavior of an unknown stable system. The prediction is not very useful unless the poles of $\hat{A}_n(z^{-1})$ at each time instant are inside the unit circle. We call this property instantaneous stability. LS ARX models are also often used as a part of an adaptive system, e.g., in adaptive control[11] and adaptive IIR filtering[37]. In this case, the LS ARX models must be stable in the Lyapunov sense, or in the bounded-input bounded-output (BIBO) sense; otherwise, the adaptive system would blow up.

In this chapter, we will develop some stability results for LS ARX models which are <u>robust</u> in the sense that *the true structure of the system which generates the data does not necessarily match that of the ARX model in question.* The results can be divided into two categories: asymptotic stability for LMMSE models and instantaneous stability for adaptive IIR filters.

Note that instantaneous stability is also accepted in signal processing as BIBO (for bounded-input and bounded-output) stability of adaptive filters. In fact, an adaptive filters is a time-varying system. Therefore, it is not sufficient to keep the poles inside the unit circle at each time instant. Even if the poles always lie inside the unit circle it is still possible that the system output blows up for certain "pathological" input signals[145]. This potential problem is often ignored in the practice of adaptive signal processing and is usually not observed in computer simulations [134].

This chapter is organized as follows. It is proven in Section 6.2 that an LS ARX model is asymptotically stable if the output and input data are a realization of a jointly stationary and ergodic process and the input is a white process. In Section 6.3, a simple sufficient condition for instantaneous stability of an LS adaptive all-pole filter is established. Suppose the parameters of an LS adaptive all-pole filter at each time instant are calculated by using the well-known Levinson-Durbin algorithm[138]. If the true plant generating the measurements is persistently excited such that the eigenvalues of the (time varying) Yule-Walker matrix associated with the adaptive all-pole filter are uniformly bounded below by a nonzero number, then the filter is instantaneously stable. Moreover, a stabilizing algorithm is proposed for general adaptive IIR filters. The proposed algorithm modifies an arbitrary adaptive IIR filter into a new adaptive IIR filter which is instantaneously stable. Some simulations have been done, which shows that the overall performance of the new filter is as good as that of the original filter if the original filter is stable and performs well. Some simulation results are reported in Section 6.4. We conclude in Section 6.5.

6.2 Asymptotic Stability of LMMSE ARX Models

In this section, we consider the case that the data sequences are very large. When the measured data $\{\mathbf{y}_n\}_{n=0}^{\infty}$ and $\{\mathbf{u}_n\}_{n=0}^{\infty}$ are a realization of a jointly stationary and mean ergodic process $(\mathbf{y},\ \mathbf{u})$ with zero-mean, the polynomials of an LS ARX model can also be determined such that

$$\mathrm{E}[\hat{A}(z^{-1})\mathbf{y}_n - \hat{C}(z^{-1})\mathbf{u}_n]^2 = \min. \tag{6.3}$$

In fact, the coefficients of the polynomials $\hat{A}(z^{-1})$ and $\hat{C}(z^{-1})$ are the solution to the following Yule-Walker equation:

$$(I\ \hat{A}_1\ \cdots\ \hat{A}_p\ -\hat{C}_0\ \cdots\ \hat{C}_q)\Gamma(p+1,q) = (R^f\ 0\ \cdots\ 0\ 0\ \cdots\ 0) \tag{6.4}$$

where

$$\Gamma(p+1,q) \triangleq \mathrm{E}[(\mathbf{y}_n^T \ \cdots \mathbf{y}_{n-p}^T \ \mathbf{u}_n^T \cdots \mathbf{u}_{n-q}^T)^T(\mathbf{y}_n^T \ \cdots \mathbf{y}_{n-p}^T \ \mathbf{u}_n^T \cdots \mathbf{u}_{n-q}^T)].$$

And R^f is the minimum prediction error variance. In the case, we have an LMMSE ARX model

$$\hat{A}(z^{-1})\mathbf{y}_n = \hat{C}(z^{-1})\mathbf{u}_n + w_n,$$

which is actually a time-invariant model. Asymptotic stability has been considered for LMMSE ARX models obtained by solving Eq.(6.2) [137] or obtained by using the Steiglitz-McBride Method [46]. There, the stable true system which generates the data is described via an output error model

$$y_n = \frac{c(z^{-1})}{a(z^{-1})}u_n + v_n.$$

y_n, u_n, and v_n are the output, input and noise at time n, respectively. $a(z^{-1})$ and $c(z^{-1})$ are polynomials in z^{-1}, $a(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{na} z^{-na}$ and $c(z^{-1}) = c_1 z^{-1} + \cdots + c_{nc} z^{-nc}$. When $na < p$ or $nc < q$, some dynamics of the true system are unmodeled. Their stability results *concerning unmodeled dynamics* require that the input u_n is white noise and the numerator order $nc = 1$[137], or the denominator order $na = 1$[46]. For a general true system, they require the signal-to-noise ratio (SNR) to be sufficiently small. However, this assumption is not reasonable for most situations in communication and signal processing. A more attractive result for general ARX systems is stated below:

Theorem 6.1 Assume that the unidirectional cross-correlation from $\{\mathbf{u}_n\}$ to $\{\mathbf{y}_n\}$ is causal[1] and $\{\mathbf{u}_n\}$ is an uncorrelated process. If the Yule-Walker matrix $\Gamma(p,q)$ defined in Eq.(6.4) is positive definite, then the corresponding LMMSE ARX model is asymptotically stable. ∎

Corollary 6.1: Consider a stable SISO system with the coprime transfer function

$$y_n/u_n = c(z^{-1})/a(z^{-1}),$$

[1]This implies $E[\mathbf{y}_n \mathbf{u}_{n+k}^T] = 0$ and $E[\mathbf{u}_n \mathbf{y}_{n-k}^T] = 0$ for any n whenever $k > 0$.

where

$$a(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{p^*} z^{-p^*}$$

and

$$c(z^{-1}) = c_0 + c_1 z^{-1} + \cdots + c_{q^*} z^{-q^*}.$$

If the input u is a white process, then all the reduced-order LMMSE ARX models of order $(p, q), 1 \leq p \leq p^*, 0 \leq q \leq q^*$, are asymptotically stable. Moreover, for the case of noisy output:

$$a(z^{-1}) y_n = c(z^{-1}) u_n + d(z^{-1}) w_n,$$

the above LMMSE ARX models are also asymptotically stable if $a(z^{-1})$ and $d(z^{-1})$ are coprime and w_n is white noise independent of the white signal u_n. ∎

The proof Theorem 6.1 and Corollary 6.1 is given in Appendix D. The core idea in the proof is the exploitation of the Toeplitz structure of submatrices of the Yule-Walker matrix. Notice that no explicit assumption on the dynamics of the system generating the measured data is required in Theorem 6.1. In Corollary 6.1, the reduced order case is considered. So, Theorem 6.1 and its corollary are actually a generalization of the well-known result on robust stability of LMMSE all-pole models[138]. The assumption of stationarity and ergodicity here could be too restrictive for some applications. Compared with the results in [46][137], the assumptions required in Theorem 6.1 are more reasonable. This is because Theorem 6.1 concerns general stable ARX systems and no requirement on the SNR is needed there. The causality assumption is no problem for most open-loop systems. Correlated input signals can be prewhitened by an inverse filter so that the requirement of uncorrelated input is not restrictive to real applications. The last assumption is involved with the persistent excitation condition. For system identification this is a minimum requirement.

6.3 Instantaneous Stability

In this section, we consider the instantaneous stability for general adaptive IIR Filters, which are not necessarily LS IIR filters. The instantaneous stability of an adaptive IIR filter depends on the model structure of the true system and the IIR filter, as well as the data sequence[46][137]. Due to the randomness of the measured data, an adaptive IIR filter could be instantaneously unstable at some time instant even when the structure of the IIR filter exactly matches the structure of the true system. Instantaneous stability has been extensively studied in adaptive signal processing, where it is often accepted as BIBO stability of adaptive filters, at least in practice [134]. However, instantaneous stability of adaptive filters is still "an ongoing area of research"[134]. In principle, seeking an instantaneously stable IIR filter by fitting observed data can be formulated as a minimization problem with inequality constraints. However, this approach is not practical for on-line system identification or signal processing because of the computational complexity and the existence of local minimizers. Instead of directly solving the constrained optimization problem, the most popular methods are composed of two dependent operations: stability monitoring and stabilization (projection). Thus, the core problem becomes how to trade off the computational complexity and the performance of algorithms for stability monitoring and stabilization. The performance of stability monitoring algorithms can be described in terms of the restriction of the stability region of an ARX model. The stability region (for a given ARX model structure) is defined as a coefficient space consisting of coefficients of all stable models with the same model structure. The modified Shur-Cohn test does not restrict the stability region. However, its computational complexity seems very high in practice because it needs to compute n determinants for a polynomial of degree n. The stabilizing algorithms are much more complicated in performance analysis than stability monitoring algorithms. This is because the overall performance of a stabilized ARX model involves the location (within the unit circle) to which the unstable poles will be projected, directly or indirectly.

In this section, we consider instantaneous stability of SISO adaptive IIR or zero-pole filters. The available data $\{y_t\}_{t=1}^n$ and $\{u_t\}_{t=1}^n$ are assumed to be generated by a stable linear stochastic time-varying system. *The output and input processes need not be assumed to be jointly stationary and ergodic.* We first develop a very simple sufficient condition for instantaneous stability of adaptive all-pole filters. Then, we apply the sufficient condition to general adaptive IIR filters so that a stabilizing algorithm is established.

6.3.1 Instantaneous Stability for All-Pole Filters

Before discussing instantaneous stability of all-pole adaptive filters, we need to introduce some notions and to make some observations.

Definition 6.1 [The TORA[2] all-pole filters]*:* A TORA all-pole filter is defined as

$$x_n + \tilde{a}_{n,1}^p x_{n-1} + \cdots + \tilde{a}_{n,p}^p x_{n-p} = s_n, \tag{6.5}$$

where x_n and s_n are the filter output and input. The filter parameters $(\tilde{a}_{n,1}^p \cdots \tilde{a}_{n,p}^p)$ are the TORA parameter estimates of an ARX model given data $\{y_t\}_{t=1}^n$. Specifically, the parameters of the TORA filter in (6.5) are identical to the solution to the following Yule-Walker equation:

$$(1\ \tilde{a}_{n,1}^p \cdots \tilde{a}_{n,p}^p)\tilde{\Gamma}_n(p+1) = (\tilde{R}_n^f(p)\ 0 \cdots 0), \tag{6.6}$$

where matrix $\tilde{\Gamma}_n(p+1)$ is a symmetric Toeplitz matrix of dimension $p+1$ with the first row being $(r_n(0)\ r_n(1) \cdots r_n(p))$, where $r_n(m) \triangleq \frac{1}{n}\sum_{k=1}^{n} y_k y_{k-m} + \delta(m)/nr_0$, $m = 0, 1, \cdots, p$. $\delta(m)$ is a delta function and r_0 is a positive number. ∎

As shown in Section 4.4, the parameters of a TORA all-pole filter converge to those of the corresponding LS all-pole filter if the sequence $\{y_k\}_{k=1}^\infty$ is uniformly bounded and the minimum eigenvalue of matrix $n\tilde{\Gamma}_n(p+1)$ approaches

[2] "TORA" is an abbreviation for the Time and Order Recursive Algorithm.

infinity as n goes to infinity. For the purpose of analysis, we assume that the solution to Eq.(6.6) is obtained by using the order-recursive Levinson-Durbin algorithm[138]. Computationally, in fact, the equation can be solved via the QR-based order-recursive Levinson algorithm[100], which is better conditioned. The quantity $r_n(m), m = 0, 1, \cdots, p$, can be updated time-recursively. The estimates of a TORA filter, which we call the TORA parameter estimates, are interpreted below (See [112] or Chapter 4 for details).

Observation 6.1 [An asymptotic property of TORA estimates]: Let $\{y_t\}_{t=1}^{\infty}$ be a uniformly bounded output sequence. Denote by the gain matrix

$$\hat{P}_{f,n}^{y} = [\sum_{k=1}^{n}(y_{k-1} \cdots y_{k-p})(y_{k-1} \cdots y_{k-p})^T]^{-1},$$

where $y_k, k \leq 0$, are assigned values of zero. The least-squares (LS) parameter estimate, minimizing $\sum_{k=1}^{n} \|y_k + a_{n,1}^p y_{k-1} + \cdots + a_{n,p}^p y_{k-p}\|^2$, of an all-pole model of order p can be well approximated by using the corresponding TORA estimate when the number of data is large. The approximation error approaches zero as n goes to infinity with convergence rate of $\mathbf{O}(\|\hat{P}_{f,n}^{y}\|_2)$ if

(1) the gain matrix satisfies: $\|\hat{P}_{f,n}^{y}\|_2 \to 0$ as $n \to \infty$.

(2) the LS parameter estimates of the all-pole model are uniformly bounded. ∎

Thus, it is reasonable to expect that the output of a TORA filter could converge to the output of the corresponding LS filter if the convergence rate of the TORA estimates to the LS estimates is fast enough. Note that the first assumption in Observation 6.1 is a minimum assumption for uniqueness or consistency of LS parameter estimates. As a result, Observation 6.1 indicates that, compared with the LS parameter estimate, the TORA parameter estimate does not give up much performance for LS adaptive all-pole filters which run for a long time.

Note that at each time instant, the parameters of a TORA adaptive all-pole filter are the solution to a Toeplitz Yule-Walker equation in (6.6). The Yule-Walker equation is a special case of the Yule-Walker equation in (6.4). As a result of Theorem 6.1, we have the following theorem.

Theorem 6.2 If the all-pole model in (6.5) is persistently excited in the sense that $\exists\ \alpha$ with $0 < \alpha$ such that for any $n \geq 1$

$$\tilde{\Gamma}_n(p+1) \geq \alpha I, \tag{6.7}$$

then the TORA all-pole filter described in (6.5) is instantaneously stable. That is, at each time instant $n > 0$, the poles of the polynomial $1 + \sum_{k=1}^{p} \tilde{a}_{n,k}^{p} z^{-1}$ are within the unit circle. ∎

Remark 6.1 The assumptions of Theorem 6.2 are weak and reasonable in practice. The first inequality in (6.7) is always true for uniformly bounded data sequences. ∎

6.3.2 Instantaneous Stability for Adaptive IIR Filters

In this subsection, we consider general adaptive IIR filters described as

$$\hat{A}_n(z^{-1})y_n = \hat{C}_n(z^{-1})u_n + v_n \tag{6.8}$$

where $\hat{A}_n(z^{-1})$ and $\hat{C}_n(z^{-1})$ are two polynomials in the backward shift operator z^{-1} with bounded coefficients described as

$$\hat{A}_n(z^{-1}) = 1 + \hat{a}_1(n)z^{-1} + \cdots + \hat{a}_p(n)z^{-p}$$

$$\hat{C}_n(z^{-1}) = \hat{c}_0(n) + \hat{c}_1(n)z^{-1} + \cdots + \hat{c}_q(n)z^{-q}.$$

The filter in (6.8) can be *any optimal filter*, say an LS filter. Lyapunov or BIBO stability is a minimum requirement for an adaptive IIR filter; otherwise, the filter output would blow up. As discussed in [134], BIBO stability is often treated as instantaneous stability in signal processing. The filter performance, which may be specified in terms of output prediction error, is another very important factor to be considered. However, obtaining a stable adaptive IIR filter which is optimal in some sense would be extremely difficult because (1) the time varying coefficients of an adaptive filter are unknown before adaptation, (2) on-line data processing cannot afford heavy computation. Instead, we may have to design a stable adaptive IIR filter, whose performance is close to that of the desired optimal stable filter.

Note that instantaneous stability of an adaptive filter in (6.8) is only dependent on $\hat{A}_n(z^{-1})$. It follows from Theorem 6.2 that, if we use a <u>*proper*</u> TORA all-pole filter $1/\tilde{A}_n(z^{-1})$ to replace $1/\hat{A}_n(z^{-1})$, then the new filter $\hat{C}_n(z^{-1})/\tilde{A}_n(z^{-1})$ is stable when some persistent excitation condition is satisfied. Now the problem is how to get the all-pole filter $1/\tilde{A}_n(z^{-1})$. In this subsection, we mainly devote our efforts to developing an algorithm for obtaining such an all-pole filter. The idea behind the proposed algorithm comes from the following observation.

Observation 6.2 Suppose that the available output and input data sequences are generated by a *stable* ARX system:

$$A(z^{-1})y_n = C(z^{-1})u_n + v_n \tag{6.9}$$

where $A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p}$ and $C(z^{-1}) = c_0 + c_1 z^{-1} + \cdots + c_q z^{-q}$. v_n is white noise. The system has an equivalent description

$$A(z^{-1})\ (y_n - \tfrac{C(z^{-1})}{A(z^{-1})}u_n) = v_n.$$

Denote $y_n - \frac{C(z^{-1})}{A(z^{-1})}u_n$ by $\tilde{y}_n$. Thus, $\tilde{y}_n$ is the output prediction error of an output error model[102]. And $\tilde{y}_n$ is *the output of an all-pole model with denominator polynomial* $A(z^{-1})$. Furthermore, $\frac{C(z^{-1})}{A(z^{-1})}$ can be approximated by a polynomial in z^{-1} of finite order because all poles of $A(z^{-1})$ are assumedly inside the unit circle. So, we may use the output of a long moving-average (MA) or all-zero model to approximate the $\frac{C(z^{-1})}{A(z^{-1})}u_n$. ∎

In the proposed algorithm, a long MA filter, $M_n(z^{-1})$, is constructed by fitting the output and input data sequences $\{y_t\}_{t=1}^n$ and $\{u_t\}_{t=1}^n$. Then, the MA filter is used to generate $w_n = y_n - M_n(z^{-1})u_n$. Finally, a TORA all-pole filter $1/\tilde{A}_n(z^{-1})$ is obtained based upon $\{w_t\}_{t=1}^n$. A formal statement of the algorithm as follows.

Stabilizing algorithm.

Step 1: Determine the coefficients of $M_n(z^{-1}) = \sum_{i=0}^s \alpha_i z^{-i}$ which minimize

$$\sum_{t=1}^n (y_t - \sum_{i=0}^s \alpha_i u_{t-i})^2.$$

Step 2: Determine the output prediction error $\{w_t\}_{t=1}^n$

$$w_t = y_t - M_t(z^{-1})u_t.$$

Step 3: Determine the TORA all-pole filter, using $\{w_t\}_{t=1}^n$,

$$\tilde{A}_n(z^{-1}) = 1 + \tilde{a}_{n,1}z^{-1} + \cdots + \tilde{a}_{n,p}z^{-p}.$$

∎

The two least-squares problems involved in the algorithm can be solved recursively in time. As a result, the computational complexity of the proposed algorithm is $\mathbf{O}(s^2 + p^2)$, which is acceptable in practice. Applying Theorem 6.2 to the stabilizing algorithm, we have the following conclusion.

Theorem 6.3 If the sequences of Yule-Walker matrices $\tilde{\Gamma}_n(p+1)$ associated with the TORA estimate $(\tilde{a}_{n,1} \ \cdots \ \tilde{a}_{n,p})$ satisfy Eq.(6.7) for some $\alpha > 0$. then the adaptive IIR filter $\hat{C}_n(z^{-1})/\tilde{A}_n(z^{-1})$ is instantaneously stable. ∎

6.4 Simulation

The following simulations were made to test the proposed algorithm. The true plant generating the measurements $\{y_t\}_{t=1}^n$ and $\{u_t\}_{t=1}^n$ is:

$$y_n - 1.5y_{n-1} + 0.62y_{n-2} - 0.048y_{n-3} = u_n - 0.5u_{n-1} + v_n$$

where $u_n = sign(w_n)$ is a pseudo random binary signal and w_n is a pseudo white signal. The plant is an ARX system of order (3, 1) with poles -0.8, -0.6, and -0.1.

1. The exact modeling case: (p, q) = (3, 1)

   $s = 15$ and v_n is a white signal.

2. The approximate modeling case: (p,q) = (2, 1)

   $s = 10$ and v_n is a correlated noise process generated by

   $$v_n + 0.7v_{n-1} = e_n + 0.3e_{n-1},$$

   where e_n is a white signal.

The simulation results for case 1 and 2 are shown in Figures 6.1 and 6.2, respectively. In case 1, the LS parameter estimates, i.e., the coefficients of $\hat{A}_n(z^{-1})$ and $\hat{C}_n(z^{-1})$, are consistent. The simulation result shows that the parameters of the stabilized adaptive IIR filter also converge to the parameters of the true plant. In case 2, the parameters of the LS adaptive IIR filter are not updated in a consistent way because the parameter estimator is undermodeled and the noise v_n is correlated. We use the weighted accumulated *posterior* prediction error (APE) defined as

$$\frac{1}{N}\sum_{k=1}^{N}\|y_k + a_{k,1}^{2,1}y_{k-1} + a_{k,2}^{2,1}y_{k-2} - c_{k,0}^{2,1}u_k - c_{k,1}^{2,1}u_{k-1}\|^2$$

to measure the performance of the LS adaptive IIR filter and the stabilized filter, where $(a_{k,1}^{2,1}\ a_{k,2}^{2,1}\ \ -c_{k,0}^{2,1}\ \ -c_{k,1}^{2,1})$ is the parameter vector of the LS adaptive IIR filter or the stabilized filter.

The simulation shows that the performance of the stabilized filter asymptotically approaches that of the LS IIR filter.

Remark 6.2 This algorithm may not need stability-monitoring algorithms to detect instantaneous instability. This is because, as the simulation results suggest, the stabilized filter just degrades the original filter slightly if the original filter is stable and its overall performance is good. When the original filter is instantaneously unstable for some time instant, the proposed algorithm shifts all the filter poles to the interior of the unit circle. ∎

6.5 Summary

In this chapter, asymptotic stability and instantaneous stability for least-squares (LS) ARX models and adaptive IIR filters are studied. All the stability results obtained are *robust* to the model structure of the true system generating the data measurements. The core result in this paper is the sufficient condition on instantaneous stability of TORA adaptive all-pole filters. Also it provides a

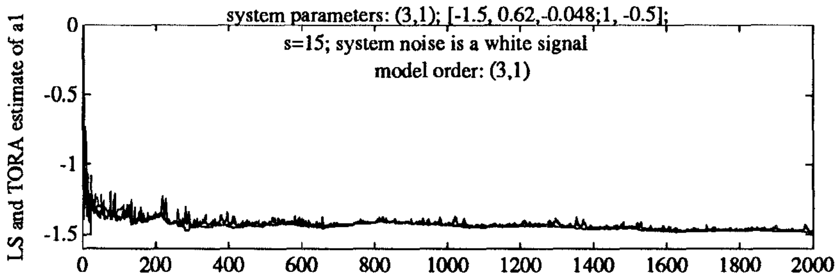


Fig. 6.1.1

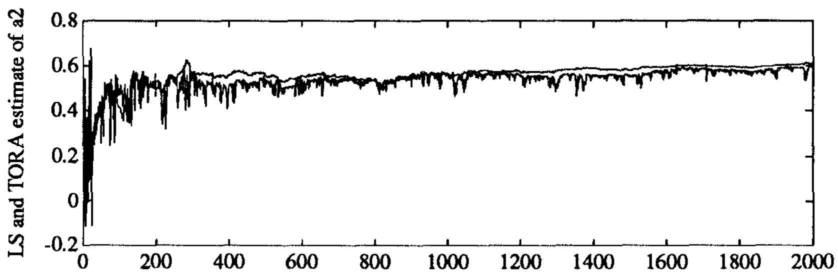


Fig. 6.1.2

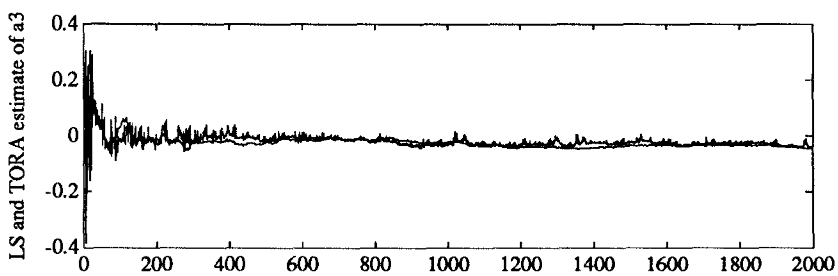


Fig. 6.1.3

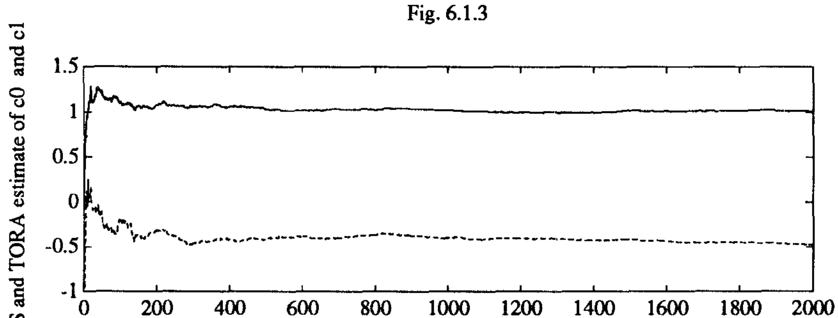


Fig. 6.1.4

Figure 6.1: An LS adaptive IIR filter $\hat{C}_n(z^{-1})/\hat{A}_n(z^{-1})$ and its modified version $\hat{C}_n(z^{-1})/\tilde{A}_n(z^{-1})$, which is obtained by using the proposed stabilizing algorithm, are compared. The coefficients of the denominator polynomials $\hat{A}_n(z^{-1}) = 1 + \hat{a}_{n,1}z^{-1} + \hat{a}_{n,2}z^{-2} + \hat{a}_{n,3}z^{-3}$ and $\tilde{A}_n(z^{-1}) = 1 + \tilde{a}_{n,1}z^{-1} + \tilde{a}_{n,2}z^{-2} + \tilde{a}_{n,3}z^{-3}$ are shown, respectively, in Figures 6.1.1 - 6.1.3, where the "jumpier" curves represent $\tilde{a}_{n,1}, \tilde{a}_{n,2}$, and $\tilde{a}_{n,3}$. The parameters of the common numerator are shown in Figure 6.1.4.

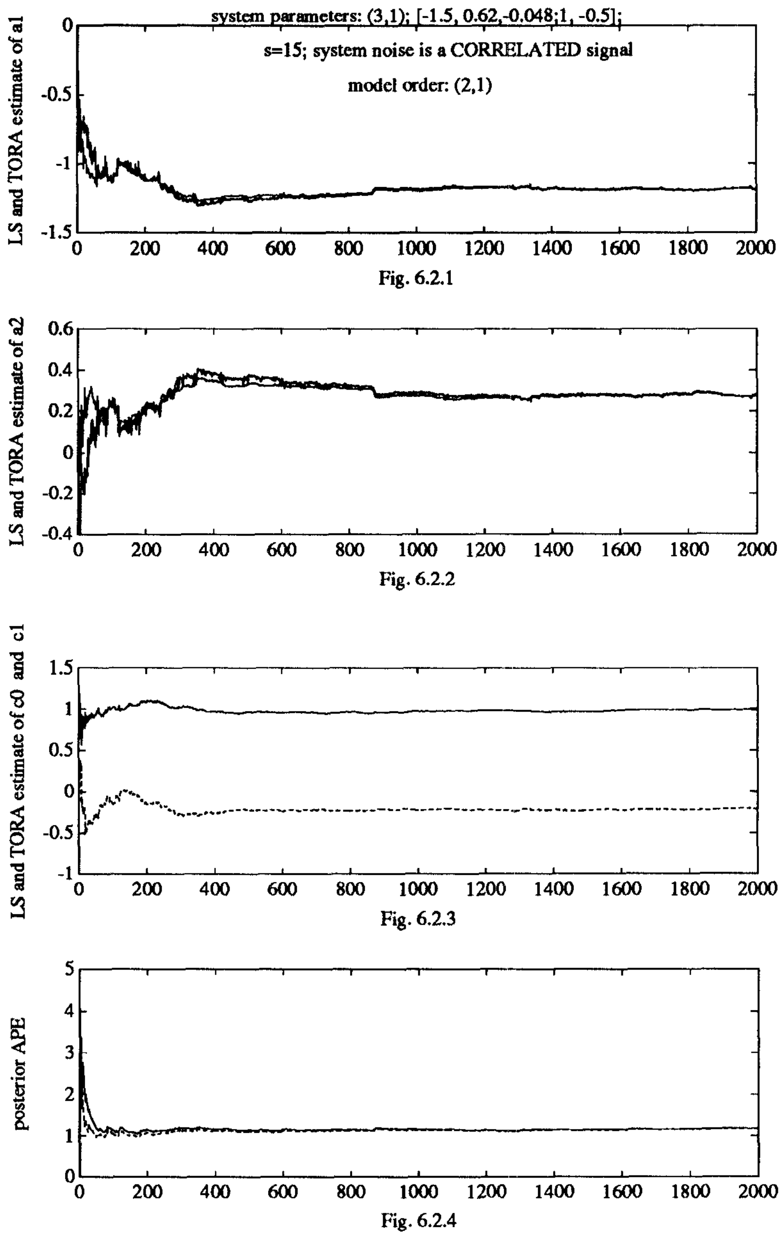


Figure 6.2: A *reduced-order* LS adaptive IIR filter $\hat{C}_n(z^{-1})/\hat{A}_n(z^{-1})$ and its modified version $\hat{C}_n(z^{-1})/\tilde{A}_n(z^{-1})$, which is obtained by using the proposed stabilizing algorithm, are compared with each other. The coefficients of the denominator polynomials $\hat{A}_n(z^{-1}) = 1 + \hat{a}_{n,1}z^{-1} + \hat{a}_{n,2}z^{-2}$ and $\tilde{A}_n(z^{-1}) = 1 + \tilde{a}_{n,1}z^{-1} + \tilde{a}_{n,2}z^{-2}$ are shown, respectively, in Figures 6.2.1 - 6.2.2, where the "jumpier" curves represent $\tilde{a}_{n,1}$ and $\tilde{a}_{n,2}$. The parameters of the common numerator are shown in Figure 6.2.3. The APE of $\hat{C}_n(z^{-1})/\hat{A}_n(z^{-1})$ and $\hat{C}_n(z^{-1})/\tilde{A}_n(z^{-1})$ are expressed in the solid and dashed curves, respectively, in Figure 6.2.4.

stabilizing algorithm for general adaptive IIR filters, which could be applied to adaptive model predictive control. Notice that instantaneous stability is often accepted as BIBO stability in practice of adaptive signal processing. Therefore, the results contained in this chapter could have applications to communications and signal processing. The asymptotic properties of the proposed stabilizing algorithm need further study. Also Theorem 6.2 needs further investigation with regard to BIBO stability.

Chapter 7

Conclusions

Simultaneous estimation of system order and parameters has been considered from the points of view of adaptive control and adaptive signal processing in this dissertation. The goals have been to enrich knowledge and understanding of order estimation and to develop some tools to implement order estimation concepts on-line for autoregressive systems with exogenous inputs. The new knowledge and algorithms are believed to be useful in treating adaptive control and IIR signal processing problems. The main contributions of this dissertation are summarized as follows.

Chapters 2 and 5 are devoted to narrowing the gap between the theoretical results on order estimation and the engineering requirements coming from adaptive control and IIR signal processing. For instance, the effect of model order in adaptive control was illustrated in terms of parametric and unparametric uncertainties. This indicates potential improvements resulting from the inclusion of model order estimation in robust adaptive control systems. The major approaches for order estimation and the key factors in consistency of order estimation and convergence rate of parameter estimation were also explained. These ideas can help us find innovative approaches to incorporating robust control techniques into the design of adaptive control systems and IIR filters.

One of the most important contributions in this dissertation is the direct exploitation of the Toeplitz structure of submatrices in a Toeplitz submatrix system. In Chapter 3, the fundmental Levinson-Durbin algorithm was generalized from Toeplitz systems to Toeplitz submatrix systems, or from pure AR models to ARX and all-zero models. As a result, a fast order-recursive algorithm was developed for determining all solutions to Toeplitz submatrix systems of different order. This generalization also enables us to have a lattice form of LMMSE IIR filters. In Chapter 4, the computational bottleneck in determining a family of LS ARX models was overcome by the introduction of the time and order recursive algorithm. The convergence of the TORA estimates to LS estimates was proved provided the data used in estimation are bounded. This algorithm provides us with a practical tool for on-line model selection based on application-oriented criteria. In addition, the symmetric treatment of system output and input measurements in the TORA makes it more attractive in some applications. The TORA was applied to simultaneous estimation of ARX system order and parameters in Chapter 5, thereby establishing a fast method. Martingale analysis shows that the new method preserves the strong consistency of the previous method of Hemmerly and Davis which is of high computational complexity. Also, simulation studies manifest that its transient performance is satisfactory.

An implementable stabilizing algorithm for general adaptive IIR filters was developed in Chapter 6 in the sense of instantaneous stability. The algorithm has two distinguished features: it does not require stability monitoring and does not need knowledge of the system which generates the data being processed. These features make the algorithm useful in adaptive IIR filtering.

The future research will be devoted to fast algorithms, simultaneous estimation of feedback system order and parameters and its application to adaptive control, and BIBO stability of IIR filters, with emphasis on robust adaptive control system design. The main problems to be addressed are outlined below.

- Some simple numerical examples need to be designed to show that some conventional adaptive control methods may fail in some normal situations unless order estimation is introduced. This is possible since we have investigated order estimation for feedback systems in a comprehensive way and Rohrs et al. [128][129] has done similar things to adaptive control with respect to robustness. Through these examples, not only would the role of order estimation be shown, also the intrinsic relation between an estimator of system order and parameters and a robust controller would be precisely and clearly illustrated, including how robustness considerations in the controller affect the consistency of order estimation and convergence of parameter estimation. Based on this, an innovative adaptive control approach combining robust control techniques and simultaneous estimation of system order and parameters could be proposed and analysed.

- The numerical robustness of the time and order recursive algorithm needs to be improved. This is because some matrix has to be inversed in the algorithm and the matrix has a larger condition number than the original estimation problem. The technique used in improving the numerical robustness of the Levinson algorithm [100] can also be used for our purpose.

- The TORA is limited by the boundedness of the data involved. QR-based fast LS algorithms for parameter estimation of ARX or ARMAX models and/or for IIR signal processing are still required in certain applications. The introduction of forward/backward output/input predictors in the TORA sheds some new hope for this long-standing problem. This is because, historically, the clever use of forward/backward output predictors resulted in the development of the lattice LS algorithm for AR models.

- There are some questions to be addressed for order estimation criteria. The accumulated prediction error criterion asymptotically has a flat descent surface in the overmodeled case as a function over model order. This causes an overmodeled model to be much harder to reject than an under-modeled model. In other words, many more data points are needed to exclude an overmodeled model from the set of model candidates. It is preferred in applications that the surface of an order estimation criterion be monotonically increasing as the "distance" between a model order and the true system order increases. This nice property could help us in tradeoffs between system performance and model complexity.

- Instantaneous stability does not guarantee BIBO stability for time-varying systems, including adaptive IIR filters, although this is ignored in signal processing. For completeness, the results about instantaneous stability developed in this dissertation need further investigation with regard to BIBO stability.

Bibliography

[1] Akaike, H., "Fitting Autoregressive Models for Prediction," Ann. Inst. Stat. Math., vol. 21, no. 2, pp. 243–247, 1969.

[2] Akaike, H., "Statistical Predictor Identification," Ann. Inst. Stat. Math., vol. 22, no.2 pp. 203–217, 1970.

[3] Akaike, H., "Information Theory and an Extension of the Maximum Likelihood Principle," Proc. 2nd Int. Symp. Inform. Theory, pp. 267–281, 1972.

[4] Akaike, H., "Use of an Information Theoretic Quantity for Statistical Model Identification," Proc. 5th Hawaii Int. Conf. System Theory, pp. 249–250, 1972.

[5] Akaike, H., "A New Look at the Statistical Model Identification," IEEE Trans. AC, vol., AC-19, no. 6, pp. 716–723, 1974.

[6] Akaike, H., "Markovian Representation Of Stochastic Processes By Canonical Variables," SIAM J. Control, Vol 13, No. 1, January 1975.

[7] Akaike, H., "On Entropy Maximisation Principle," in **Application of Statistics** (ed. P. R. Krishnaiah), 27-41. Amsterdam, North-Holland, 1977.

[8] Ammar, G.S. and Gragg, W.B., "Numerical Experience with a Superfast Real Toeplitz Solver," Linear Algebra And Its Applications, 121: 185-206, 1989.

[9] Anderson, B. D. O. et al., **Stability of Adaptive Systems,** The MIT Press, Cambridge, Massachusetts, 1986.

[10] Åström, K.J., Hagander, P. and Sternby, J, "Zeros of Sampled Systems," Automatica, 1984.

[11] Åström, K. J. and Wittenmark, B., **Adaptive Control**, Addison-Wesley Publishing Company, 1986.

[12] Åström, "Automatic Tuning of PID Controllers," Inst. Soc. of America, 1988.

[13] Bai, E.-W., "Adaptive Identifier with a Self-Tuning Filter," Proc. of the 29th CDC, pp.3247–3248, Honolulu, Hawaii, 1990.

[14] Baram, Y. and Porat, B. , "Identification of Minimal Order State Space Models From Stochastic Input-Output Data," SIAM J. Control and Optimization, Vol.26, No.1, January 1988.

[15] Bellanger, M.G., "The FLS-QR Algorithm for Adaptive Filtering," Signal Processing, **17**, 291–304, 1989.

[16] Bistritz, Y., Lev-Ari, H., and Kailath, T., " Immittance-Domain Levinson Algorithms," IEEE Trans. Inform. Theory, Vol. 35, No.3, 675 - 682, May 1989.

[17] Bojanczyk, A.W., Brent, R.P., and Hoog, F.R., "QR Factorization of Toeplitz Matrices," Numer. Math. **49**, 81-94, 1986.

[18] Box, G.E.P. and Jenkins, G.M., **Time Series Analysis — forecasting and control**, Holden-Day Inc., 1976.

[19] Boyd, S.P. and Barratt, C.H., **Linear Controller Design** — Limits of Performance, Prentice Hall, 1991.

[20] Bunch, J.R., "Stability of Methods for Solving Toeplitz Systems of Equations," SIAM Sci. Stat. Comput., Vol. 6, No. 2, 1985.

[21] Caines, P.E., **Linear Stochastic Systems**, John Wiley & Sons, Inc., 1988.

[22] Cadzow, J.A., "Signal Processing via Least Squares Error Modeling," IEEE ASSP Magazine, 12-31, Oct. 1990.

[23] Chen, H.F., " Strong Consistency And Convergence Rate of Least Squares Identification," Sci. Sin. (Ser. A) 25, 771-784, 1982.

[24] Chen, H.F., **Recursive Estimation & Control For Stochastic Systems**, John Wiley & Sons, Inc., 1985.

[25] Chen, H.F. and Guo, L., "Convergence Rate of Least-Squares Identification And Adaptive Control for Stochastic Systems," Int. J. Control, Vol.44, No. 5, pp1459-1476, 1986.

[26] Chen, H.F. and Guo, L., "Asymptotically Optimal Adaptive Control With Consistent Parameter Estimates," SIAM J. Control and Optimization, Vol.25, No.3, May 1987.

[27] Chen, H.F. and Guo, L., "Optimal Adaptive Control and Consistent Parameter Estimates For ARMAX Model With Quadratic Cost," SIAM J. Control and Optimization, Vol.25, No.4, July 1987.

[28] Chen, H.F. and Guo, L., "Consistent Estimation of the Order of Stochastic Control Systems," IEEE Trans. AC, Vol. 32, no. 6,pp.531–535, 1987.

[29] Chen, H.F. and Zhang, J.F., " Identification and Adaptive Control for Systems with Unknown Orders, Delay, and Coefficients," IEEE AC Vol.35 No.8, pp. 866 - 877, 1990.

[30] Chen, H.F. and Zhang, J.F. , "Stochastic Adaptive Control for System with Noise Being an ARMA Process," 1989.

[31] Chun, J., Kailath, T., and Lev-Ari, H., "Fast Parallel Algorithms for QR and Triangular Factorization," SIAM J. Sci. Stat. Comput., Vol. 8, No. 6, 1987.

[32] Cioffi, J.M., "Limited-Precision Effects in Adaptive Filtering," IEEE Trans. Circuits Syst., Vol. CAS-34, No. 7, July 1987.

[33] Cioffi, J.M., "The Fast Adaptive ROTOR's RLS Algorithm," IEEE Trans. ASSP, Vol. 38, No. 4, April 1990.

[34] Clarke, D. W., C. Mohtadi and P. S. Tuffs, "Generalized Predictive Control — Part I. The Basic Algorithm," Automatica, **23**, pp.137–148, 1987.

[35] Clarke, D. W., C. Mohtadi and P. S. Tuffs, "Generalized Predictive Control — Part II. Extensions and Interpretations," Automatica, **23**, pp.149–160, 1987.

[36] Clarke, D. W. and C. Mohtadi, " Properties of Generalized Predictive Control," IFAC world Congress, Munich, 1987.

[37] Cowan, C.F.N. and Grant, P.M., "**Adaptive Filters**," Prentice-Hall, 1985.

[38] Cybenko, G., " The Numerical Stability of the Levinson-Durbin Algorithm for Toeplitz Systems of Equations," SIAM J. Sci. Stat. Comput., Vol 1, No.3, 303 - 319, Sep., 1980.

[39] Davis, M.H.A. and Vinter, R.B., **Stochastic Modelling and Control**, Chapman and Hall, 1984.

[40] Davisson, L.D., "The Prediction Error of Stationary Gaussian Time Series of Unknown Covariance," IEEE Trans. Inform. Theory, vol. IT-19, no. 4, pp. 527–532, Oct. 1965.

[41] Davisson, L.D., "The Adaptive Prediction of Time Series," Proc. Nat. Electronics Conf., vol. 22, pp. 557–561, 1966.

[42] Deller, J.R., "Set Membership Identification in Digital Signal Processing," IEEE ASSP Magazine, vol. 6, no. 4, pp. 4-20, 1989.

[43] Dongarra, J.J., Moler, C.B., Bunch, J.R., and Stewart, G.W., **LINPACK: Users' Guide**, SIAM, Philadelphia, 1979.

[44] Doyle, J.C., and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," IEEE Trans. on AC, Vol. AC-26, No.1, pp. 4–16, 1981.

[45] Durbin, J.,, "The Fitting of Time-Series Models," Rev. Intern. Statist. Anal. Appl., vol. 2, pp. 233–263, 1961.

[46] H. Fan and M. Nayeri, On Reduced Order Identification; Revisiting "On Some System Identification Techniques For Adaptive Filtering", IEEE Trans. Circuits Syst., Vol. 37, No. 9, 1990.

[47] Friedlander, B., Morf, M., Kailath, T., and Ljung, L., "Extended Levinson- and Chandrasekhar-type equations for a General discrete-Time Linear Estimation Problem," IEEE Trans. Automatioc Control, AC-23, pp. 653–659, 1978.

[48] Friedlander, B., Morf, M., Kailath, T., and Ljung, L., "New Inversion Formulas for Matrices Classified in Terms of Their Distance from Toeplitz Matrices," Linear Algebra and Its Applications, vol. 27, pp. 31-60, 1979.

[49] Friedlander, B., "Lattice Filters for Adaptive Processing," Proc. IEEE vol. 70, pp. 829-867, Aug. 1982.

[50] Friedlander, B., "Lattice Methods for Spectral Estimation," Proc. IEEE vol. 70, pp. 990-1017, Aug. 1982.

[51] Friedlander, B. and Smith, J. O., "Analysis and Performance Evaluation of an Adaptive Notch Filter," IEEE Trans. IT, vol. IT-30, no.2, pp. 283–295, 1984.

[52] Gerencer, L., "On Rissanen's Predictive Stochastic Complexity For Stationary ARMA Processes," Proc. of the 28th CDC, pp.1210–1215, Tampa, Florida, 1989.

[53] Giordano, A.A. and Hsu, F.M., **Least Square Estimation with Applications to Digital Signal Processing**, Prentice-Hall, Inc., 1985.

[54] Golub, G.H., "Matrix Decompositions and Statistical Calculations," in **Statistical Computation**, (ed. Milton, R.C. and Nelder, J.A.), Academic Press, pp. 365–397, 1969.

[55] Gooch, R.P. and Shynk, J. J., "Wide-Band Adaptive Array Processing Using Pole-Zero Digital Filters," IEEE Trans. Antennas and Propagation, vol. AP–34, no. 3, pp. 355–367, 1986.

[56] Goodwin, G.C., Ninnes, B., and Salgado, M.E., "Qualification of Uncertainty in Estimation," Proc. ACC, San Diego, Ca, pp.2400–2405, 1990.

[57] Grenander, U. and Szegö, G., "**Toeplitz Forms and Their Applications**," Chelsea Publishing Company, New York, 1984.

[58] Guo, L. and Chen, H.F., "A Method For Adaptive Estimation Of ARMA Processes," Systems And Control Letters, No.11, pp351-356, 1988.

[59] Guo, L. , Chen, H.F. and Zhang, J.F., "Consistent Order Estimation For Linear Stochastic Feedback Control Systems (CARMA Model)," Autimatica, vol. 25, pp.147–151, 1989.

[60] Hannan, E.J. and Quinn, B.G., "The Determination of the Order of an Autoregression," J.R. Stat. Soc., Set-B, **41**, 190–195, 1979.

[61] Hannan, E.J., "Estimating the Dimension of a Linear System," J. Multivariance Anal. **11**, 459-473.hs, 1981.

[62] Hannan, E.J., and Rissanen, J., "Recursive Estimation of Mixed Autoregressive-moving Average Order," Biometrika, 69, 1, pp.81-94, 1982.

[63] Hannan, E.J., Kavalieris, L. and Mackisack, M., "Recursive Estimation Of Linear Systems," Biometrika, 73, 1, pp.119-33, 1986.

[64] Hannan, E.J. and Deistler, E., **The Statistical Theory of Linear Systems**, John Wiley & Sons, 1988.

[65] Hartwig, R.E. and Fisher, M. E., "Asymptotic Behavior of Toeplitz Matrices and Determinants," Arch. Rat. Mech. Anal., 32, pp. 190ff, 1969.

[66] Haykin, S., "**Adaptive Filter Theory**," Prentice-Hall, 1986.

[67] Haykin, S., "**Modern Filters**," Macmillan Publishing Company, 1989.

[68] Heinig, G., and Rost, K., "**Algebraic Methods for Toeplitz-like Matrices and Operators**," Birkhäuser, 1984.

[69] Helmicki, A.J., Jacobson, C.A., and Nett, C.N., "Identification in H_∞: Linear Algorithms," Proc. ACC, San Diego, Ca, pp.2418–2423, 1990.

[70] Helmicki, A.J., Jacobson, C.A., and Nett, C.N., "Identification in H_∞: A Robustly Convergent, Nonlinear Algorithm," Proc. ACC, San Diego, Ca, pp.386–391, 1990.

[71] Hemmerly, E.M. and Davis, M.H.A., " Strong Consistency of the PLS Criterion for Order Determination of Autoregressive Processes," Ann. Statist., Vol. 17, pp.941–946, 1989.

[72] Hemmerly, E.M. and Davis, M.H.A. , "Recursive Order Estimation of Stochastic Control Systems," Math. Systems Theory, **22**, pp.323-346, 1989.

[73] Horn, R.A. and Johnson, C., **Matrix Analysis**, Cambridge Univ. Press, 1985.

[74] Horvat, S., "Adaptive IIR Digital Filters for On-line Time Domain Equalization and Linear Prediction," presented at IEEE arden House Workshop on Dig. Sig. Proc., Harriman, N.Y., 1976.

[75] Hotelling, H., "Relations Between Two Sets of Variables," Biometrika, Vol. 28, pp. 321–377, 1936.

[76] Iohvidov, I.S., "**Hankel and Toeplitz Matrices and Forms–Algebraic Theory**," Birkhäuser, 1983.

[77] Jain, A.K., **Fundamentals of Digital Image Processing**, Prentice-Hall, 1989.

[78] Jayant, N.S. and Noll, P., **Digital Coding of Waveforms: Principles and Applications to Speech and Video**, Prentice-Hall, 1984.

[79] Judge, **The Theory And Practice of Econometrics**, 1985.

[80] Kailath, T., "A View of Three Decades of Linear Filtering Theory," IEEE Trans. Inf. Theory, IT-20(2): 146-181 (1974).

[81] Kailath, T., Kung, S.Y., and Morf, M., "Displacement Ranks of Matrices and Linear Equations," J. Math Anay. App., Vol. 68, pp.395–407, 1979.

[82] Kailath, T., **Linear System**, Prentice-Hall, Inc, 1980.

[83] Kailath, T., Bruckstein A., and Morgan, D., "Fast Matrix Factorizations via Discrete Transmission Lines," Linear Algebra and Its Applications, **75**: 1-25, 1986.

[84] Karaboyas, S., Kalouptsidis, N., and Caroubalos, C., " Efficient ARX Identification Algorithms with Full Parallelism," IEEE Trans on ASSP, Vol. 38, No. 11, Nov. 1990.

[85] Karlsson, E. and Hayes, M.H., "Least Squares ARMA Modeling of Linear Time-Varying Systems: Lattice Filter Structures and Fast RLS Algorithms," IEEE Trans on ASSP, Vol. 35, No. 7, July 1987.

[86] Kay, S.M., **Modern Spectral Estimation**, Prentice-Hall, Inc., 1988.

[87] Knopp, **Infinite Sequences and Series**, 1956.

[88] Kosut, R.L., Lau, M., and Boyd, S., "Identification of Systems with Parametric and Nonparametric Uncertainty," Proc. ACC, San Diego, Ca, pp. 2412–2417, 1990.

[89] Kullback, S., **Information Theory and Statistics**, John Wiley and Sons, 1959.

[90] Kung, S.-Y. and Hu, Y. H., " A highly Concurrent Algorithm and Pipelined Architecture for Solving Toeplitz Systems," IEEE Trans ASSP, Vol. 31, No. 1, Feb. 1983.

[91] Kung, S.-Y., Whitehouse, H.J., and Kailath, T., "**VLSI and Modern Signal Processing**," Prentice-Hall, 1985.

[92] Lai, T.L. and Wei, C.Z., " Least Squares Estimates in Stochastic Regression Models with Applications to Identification and Control of Dynamic Systems," Ann. Statist., Vol.10, pp.154-166, 1982.

[93] Lai, T.L. and Wei, C.Z., "On the Concept of Excitation in Least Squares Identification and Adaptive Control," Statistics, Vol. 16, pp.227-254, 1986.

[94] Landau, I.D., **Adaptive Control: The Model Reference Approach**, Marcel Dekker, Inc., 1979.

[95] Larimore, W.E., "System Identification, Reduced-Order Filtering And Modeling Via Canonical Variate Analysis," Proc. 1983 ACC, June 22-24, San Francisco, Ca.

[96] Larimore, W.E., "Canonical Variate Analysis in Identification, Filtering, and Adaptive Control," Proc. 29th Conf. on Decision and Control, Hawaii, Dec. 1990.

[97] Lau, M.K., Kosut, R. L., and Boyd, S.,"Parameter Set Estimation of Systems with Uncertain Nonparametric Dynamics and Disturbances," Proc. of the 29th CDC, pp.3162–3167, Honolulu, Hawaii, 1990.

[98] Lev-Ari, H., and Kailath, T., "Lattice Filter Parameterization and Modeling of Nonstationary Processes," IEEE Trans. Inform. Theory, Vol. IT-30, No.1, January 1984.

[99] Levinson, N., "The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction," J. Math. Phys., Vol. 25, no.4, pp.261-278, 1947.

[100] Lewis, P.S., "QR-Based Algorithms for Multichannel Adaptive Least Squares Lattice Filters," IEEE Trans. ASSP, Vol. 38, No. 3, March 1990.

[101] Lee, T.H. and Narendra, K.S., "Robust Adaptive Control Of Discrete-Time Systems Using Persistent Excitation," Automatica, Vol. 24, No.6, pp.781-788, 1988.

[102] Ljung, L., **System Identification: Theory for the User**, Prentice Hall, 1987.

[103] Ljung, L. and Wahlberg, B., "Influence of Model Order on Chance Detection in Noise-Free, Complex System," Proc. ACC, San Diego, Ca, pp.2388–2393, 1990.

[104] Ljung, L. "Issues in System Identification," IEEE Control Systems Magazine, Vol. 11, Nu. 1, pp. 25–229, 1991.

[105] Long, G., Shwed, D., and Falconer, D. O., "Study of a Pole-zero Adaptive Echo Canceller," IEEE Trans. Circuits Systems, vol. CAS–34, no. 7, pp. 765–769, 1987.

[106] Middleton, R.H., and Goodwin, G.C., **Digital Control And Estimation — A Unified Approach**, Prentice Hall, 1990.

[107] Markel, J.D. and Gray, A.H., Jr., **Linear Prediction of Speech**, Springer-Verlag, 1976.

[108] K. X. Miao and H. Fan, "BIBO Stability of Adaptive Normalized Lattice IIR System," Proc. of the 29th CDC, pp. 3563–3564, Honolulu, Hawaii, Dec. 1990.

[109] Moler, C., Little, J., Bangert, S., and Kleiman, S., **Pro-Matlab**, The MathWorks, Inc., 1986.

[110] Nehorai, A., "A Minimal Parmeter Adaptive Notch Filter with Constrained Poles and Zeros," IEEE Trans. ASSP, vol. ASSP–33, no. 4, pp. 983–996, 1985.

[111] Pan, J and Levine, W.S., "A Fast Order-Recursive Algorithm for Block-Toeplitz Systems with Applications to Estimation of ARX Systems," Proc. of the 29th CDC, pp.2879–2885, 1990.

[112] Pan, J. and Levine, W.S., " A Fast Time and Order Recursive Parameter Estimation Algorithm for a Set of ARX Models," Proc. of ACC, Boston, MA, pp. 2832–2838, 1991.

[113] Pan, J. and Levine, W.S., "Robust Stability of LS ARX models/IIR Filters," To appear in Proc. of the 1991 Conf. on Inform. Scie. & Sys., Baltimore, Maryland, 1991

[114] Pan, J. and Levine, W.S., "Fast Prediction/Estimation Techniques for Control Systems," Proc. of IEEE Southeastcon '91, pp. 188–192, 1991.

[115] Praly, L., Lin, S.F. and Kumar, P.R., "A Robust Adaptive Minimum Variance Controller," SIAM J. Control And Optimization, Vol.27, No. 2, p.235-266, March 1989.

[116] Prett, D.M., Skrovanek, T.A., and Pollard, J.F., "Process Identification — Past, Present, Future," 79–104, in **The Shell Process Control Workshop**, ed. by Pretty, D. M. and Morari, M., 1987.

[117] Proakis, J. G., **Digital Communications**, McGraw-Hill, 1983.

[118] Regalia, P., "A Hybrid Lattice-QR Fast Algorithm for Least Squares Adaptive Filtering," 1990.

[119] Rissanen J., "Modeling by Shortest Data Description," Automatica, Vol. 14, pp.465-471, 1978

[120] Rissanen J., "Universal Coding, Identification, Prediction, and Estimation," IEEE Trans. IT, Vol. IT-30, No.4, July 1984.

[121] Rissanen J., "Complexity of String in the Class of Markov Sources," IEEE Trans. IT , Vol. IT-32, No. 4, July 1986.

[122] Rissanen J., "A Predictive Least-Squares Principle," IMA J. Math. Contr. Inform., Vol 3, Nos 2–3, pp.211–222, 1986.

[123] Rissanen, J. and Wax, M., "Measures of Mutual and Causal Dependence Between Two Time Series," IEEE Trans. IT, Vol. IT-33, No.4, July 1987.

[124] Rissanen, J., "Stochastic Complexity," J. R. Statist. Soc. **B**(1987), **49**, No. 3, pp. 223–239 and 252–265, 1987.

[125] Rissanen, J., **Stochastic Complexity in Statistical Inquiry**, World Scientific, 1989.

[126] Rivera, D.E., and Pollard, J.F., Sterman, L.E., and Garcia, C.E., "An Industrial Perspective on Control-Relevant Identification," Proc. ACC, San Diego, Ca, pp.2406–2411, 1990.

[127] Robinson,E.A., and Treitel, S., **Geophysical Signal Analysis**, Prentice-Hall, Inc., 1980.

[128] Rohrs, C.E., Valavani, L., Athans, M., and Stein, G., "Robustness of Adaptive Control Algorithms in the Presence of Unmodeled Dynamics," Proc. of the 21st IEEE CDC, Florida, pp. 3-11, 1982.

[129] Rohrs, C.E., Valavani, L., Athans, M., and Stein, G., "Robustness of Continus-time Adaptive Control Algorithms in the Presence of Unmodeled Dynamics," IEEE Trans. AC, No. 9, pp. 881–889, 1985.

[130] Rohrs, C.E., "Rethinking Adaptive Control for the 90's," Proc. of the 29th CDC, pp. 3143–3145, Honolulu, Hawaii, 1990.

[131] Rosenblatt, **Stationary Sequences and Random Field**, 1985.

[132] Sastry, S. and Bodson,M., **Adaptive Control — Stability, Convergence, and Robustness**, Prentice-Hall, Inc., 1989.

[133] Schwarz, G., "Estimating the Dimension of a Model," Ann. Stat., 6 , 461-464, 1978.

[134] Shynk, J.J., "Adaptive IIR Filtering," IEEE ASSP Magazine, April 1989.

[135] SIAM , "Report of The Panel on Future Directions in Control Theory : A Mathematical Perspective," 1989.

[136] Smith, R.S. and Doyle, J.C., "Towards a Methodology for Robust Parameter Identification," Proc. ACC, San Diego, Ca, pp.2394–2399, 1990.

[137] Söderström, T. and Stoica, P., On the Stability of Dynamic Models Obtained by Least-Squares Identification, IEEE Trans. AC, Vol. 26, No. 2, 1981.

[138] Söderström, T. and Stoica, P., **System Identification**, Prentice-Hall, 1989.

[139] Stein, G. and Athans, M., "The LQG/LTR Procedure For Multivariable Feedback Control Design," IEEE Trans. AC , Vol.AC-32 , No.2, February 1987.

[140] Stoica, P., Söderström, T., and Friedlander, B., "Optimal Instrumental Variable Estimates of the AR Parameters of an ARMA Process," IEEE Trans. AC, vol. 30, No. 11, Nov. 1985.

[141] Sweet, D.R., "Fast Toeplitz Orthogonalization," Numer. Math., **43**, 1-21, 1984.

[142] Tugnait, J.K., "Identification of Non-minimum Phase Linear Stochastic Systems," Automatica, Vol.22, No.4, pp.457-464, 1986.

[143] Tzes, A.P., and Yurkovich, S., "On Frequency Domain Loop Shaping for Self-Tuning Control," Proc. ACC, San Diego, Ca, pp.61–66, 1990.

[144] Veres, S.M., "Strong Consistency of the Accumulated Prediction Error Criterion For Multivariate Processes," 1988.

[145] M. Vidyasagar, "Nonlinear Systems Analysis", Prentice-Hall, Englewood Cliffs, N.J., 1978.

[146] Wang, K and Ljung, L., "A Discussion of Adaptive Stabilization And Robust Adaptive Control," Systems And Control Letters, No.12, pp53-56, 1989.

[147] Wasilkowski, G.W., "Information-Based Complexity; An Overview," Proc. ACC, San Diego, Ca, pp.744–379, 1990.

[148] Wax, M., " Order Selection for AR Models by Predictive Least Squares," IEEE Trans. Acoust. Speech Signal Process., Vol. 36, No. 4, pp. 581-588, 1988.

[149] Wei, C.Z. , "Adaptive Prediction By Least Squares Predictors In Stochastic Regression Models with Applications to Time Series," The Analysis of Statistics, Vol.15, No.4 , 1987.

[150] Whittle, P., "On the Fitting of Multivariate Autogressions and the Approximate Canonical Factorization of a Spectral Density Matrix," Biometrika, vol. 50, pp. 129–134, 1963.

[151] Wiggins, R.A. and Robbinson, E.A., "Recursive Solution to the Multichannel Filtering Problem," J. Geophys. Res., vol. 70, pp. 1885–1891, 1965.

[152] Yaglom, A.M., "Outline of Some Topics in Linear Extrapolation of Stationary Random Processes," Proc. Fifth Berkeley Symp. Math. Stat. and Prob., California Press, pp. 259–278, 1970.

[153] Yam, Y., Bayard, D.S., and Scheid, R.E., "Integrated Identification and Robust Control Tuning for Large Space Structures," Proc. ACC, San Diego, Ca, pp.2424–2429, 1990.

[154] Younce, R.C. and Rohrs, C.E., "Identification with Non-Parametric Uncertainty," Proc. of the 29th CDC, pp.3154–3161, Honolulu, Hawaii, 1990.

[155] Zames, G., "Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses," IEEE Trans. AC, Vol. AC-26, No.2, 1981.

[156] Zervos, C., Belanger, P.R. and Dummont, G.A., "On PID Controller Tuning Using Orthonormal Series Identification," Automatica, Vol.24, No.2, pp.165-175, 1988.

Appendix A

The Derivation of the Order-Recursive Algorithm

A.1 The Derivation of the ORA in P

It follows from Eq. (3.30) that the equation of order $(p+1,q)$ can be described as follows:

$$\begin{pmatrix} I & A_1^{p+1,q} & \dots & A_{p+1}^{p+1,q} & -C_0^{p+1,q} & -C_1^{p+1,q} & \dots & -C_q^{p+1,q} \\ B_{p+1}^{p+1,q} & B_p^{p+1,q} & \dots & I & -D_q^{p+1,q} & -D_{q-1}^{p+1,q} & \dots & 0 \\ 0 & -H_1^{p+1,q} & \dots & -H_{p+1}^{p+1,q} & I & G_1^{p+1,q} & \dots & G_q^{p+1,q} \\ -F_{p+1}^{p+1,q} & -F_p^{p+1,q} & \dots & -F_0^{p+1,q} & E_q^{p+1,q} & E_{q-1}^{p+1,q} & \dots & I \end{pmatrix} \times \Gamma(p+1,q)$$

$$= \begin{pmatrix} R^f(p+1,q) & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & R^b(p+1,q) & 0 & 0 & \dots & \nu(p+1,q) \\ \eta(p+1,q) & 0 & \dots & 0 & 0 & V^f(p+1,q) & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & V^b(p+1,q) \end{pmatrix}. \tag{A.1}$$

By rearranging Eq.(A.1), we obtain the following four normal equations:

$$\begin{array}{lllllllll} (& R_{yy}(1) & \dots & R_{yy}(p) & R_{yy}(p+1) & R_{yu}(0) & R_{yu}(1) & \dots & R_{yu}(q) \quad) \\ +(& A_1^{p+1,q} & \dots & A_p^{p+1,q} & A_{p+1}^{p+1,q} & -C_0^{p+1,q} & -C_1^{p+1,q} & \dots & -C_q^{p+1,q} \quad)R^1(p+1,q) \\ =(& 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \quad), \end{array} \tag{A.2}$$

$$
\begin{array}{llllllll}
(& R_{yy}(-p-1) & \dots & R_{yy}(-1) & R_{yu}(-p-1) & \dots & R_{yu}(q-p-2) &) \\
+(& B_{p+1}^{p+1,q} & \dots & B_1^{p+1,q} & -D_q^{p+1,q} & \dots & -D_1^{p+1,q} &)R^2(p+1,q) \\
=(& 0 & \dots & 0 & 0 & \dots & 0 &)
\end{array}
\tag{A.3}
$$

$$
\begin{array}{lllllllll}
(& R_{uy}(1) & \dots & R_{uy}(p) & R_{uy}(p+1) & R_{uu}(1) & \dots & R_{uu}(q) &) \\
+(& -H_1^{p+1,q} & \dots & -H_p^{p+1,q} & -H_{p+1}^{p+1,q} & G_1^{p+1,q} & \dots & G_q^{p+1,q} &)R^3(p+1,q) \\
=(& 0 & \dots & 0 & 0 & 0 & \dots & 0 &),
\end{array}
\tag{A.4}
$$

$$
\begin{array}{lllllllll}
(& R_{uy}(-q) & \dots & R_{uy}(p-q) & R_{uy}(p-q+1) & R_{uu}(-q) & \dots & R_{uu}(-1) &) \\
+(& -F_{p+1}^{p+1,q} & \dots & -F_1^{p+1,q} & -F_0^{p+1,q} & E_q^{p+1,q} & \dots & E_1^{p+1,q} &)R^4(p+1,q) \\
=(& 0 & \dots & 0 & 0 & 0 & \dots & 0 &),
\end{array}
\tag{A.5}
$$

where $R^1(p+1,q)$, $R^2(p+1,q)$, $R^3(p+1,q)$, and $R^4(p+1,q)$ are defined in Definition 3.2. To relate the solution to Eq.(3.19) of order (p,q) with that of order $(p+1,q)$, introduce four sets of partial correlation coefficients (PCC), $K_2^{p,q} \in \mathcal{R}^{m\times m}$, $K_3^{p,q} \in \mathcal{R}^{m\times l}$, $L_1^{p,q} \in \mathcal{R}^{m\times m}$, $L_3^{p,q} \in \mathcal{R}^{m\times l}$, $L_4^{p,q} \in \mathcal{R}^{m\times l}$, $M_2^{p,q} \in \mathcal{R}^{l\times m}$, and $N_2^{p,q} \in \mathcal{R}^{l\times m}$, as the solutions to the equations:

$$
\left.
\begin{array}{l}
K_2^{p,q}R^b(p,q) + K_3^{p,q}\zeta(p,q) = -\alpha(p,q) \\
K_2^{p,q}\delta(p,q) + K_3^{p,q}V^f(p,q) = 0
\end{array}
\right\}
\tag{A.6}
$$

$$
\left.
\begin{array}{l}
L_1^{p,q}R^f(p,q) + L_3^{p,q}\eta^{p,q} = -\beta(p,q) \\
L_3^{p,q}V^f(p,q) = -\delta(p,q) \\
-L_1^{p,q}C_q^{p,q} + L_3^{p,q}G_q^{p,q} + L_4^{p,q} = D_1^{p,q}
\end{array}
\right\}
\tag{A.7}
$$

$$
M_2^{p,q}R^b(p,q) = -\zeta(p,q)
\tag{A.8}
$$

$$
N_2^{p,q}R^b(p+1,q) = -\lambda(p,q)
\tag{A.9}
$$

where $(p,q) \in \mathcal{O}_{0,-1}^{p^*,q^*}$. By Theorem 3.1 and Eq.(3.37), we have the unique partial correlation coefficients which are presented in (3.40). On the other hand, Eqs.(3.19), (A.6)–(A.9), and (A.2)–(A.5) imply:

$$
\begin{array}{rccccccl}
(& R_{yy}(1) & \ldots & R_{yy}(p+1) & R_{yu}(0) & \ldots & R_{yu}(q) &) \\
+\{\,(& A_1^{p,q} & \ldots & A_{p+1}^{p,q} & -C_0^{p,q} & \ldots & -C_q^{p,q} &) \\
+\,K_2^{p,q}\,(& B_p^{p,q} & \ldots & B_0^{p,q} & -D_{q+1}^{p,q} & \ldots & -D_1^{p,q} &) \\
+\,K_3^{p,q}\,(& -H_1^{p,q} & \ldots & -H_{p+1}^{p,q} & G_0^{p,q} & \ldots & G_q^{p,q} &)\}R^1(p+1,q) \\
=(& 0 & \ldots & 0 & 0 & \ldots & 0 &) \\
=(& R_{yy}(1) & \ldots & R_{yy}(p+1) & R_{yu}(0) & \ldots & R_{yu}(q) &) \\
+(& A_1^{p+1,q} & \ldots & A_{p+1}^{p+1,q} & -C_0^{p+1,q} & \ldots & -C_q^{p+1,q} &)R^1(p+1,q),
\end{array}
$$

$$
\begin{array}{rccccccl}
(& R_{yy}(-p-1) & \ldots & R_{yy}(-1) & R_{yu}(-p-1) & \ldots & R_{yu}(q-p-2) &) \\
+\{\,(& B_{p+1}^{p,q} & \ldots & B_1^{p,q} & -D_{q+1}^{p,q} & \ldots & -D_2^{p,q} &) \\
+\,L_1^{p,q}\,(& A_0^{p,q} & \ldots & A_p^{p,q} & -C_0^{p,q} & \ldots & C_{q-1}^{p,q} &) \\
+\,L_3^{p,q}\,(& -H_1^{p,0} & \ldots & -H_p^{p,q} & G_0^{p,q} & \ldots & G_{q-1}^{p,q} &) \\
+\,L_4^{p,q}\,(& -F_p^{p,q} & \ldots & -F_0^{p,q} & E_q^{p,q} & \ldots & E_1^{p,q} &)\,\}R^2(p+1,q) \\
=(& 0 & \ldots & 0 & 0 & \ldots & 0 &) \\
(& R_{yy}(-p-1) & \ldots & R_{yy}(-1) & R_{yu}(-p-1) & \ldots & R_{yu}(q-p-2) &) \\
+(& B_{p+1}^{p+1,q} & \ldots & B_1^{p+1,q} & -D_q^{p+1,q} & \ldots & -D_1^{p+1,q} &)R^2(p+1,q),
\end{array}
$$

$$
\begin{array}{rccccccl}
(& R_{uy}(1) & \ldots & R_{uy}(p+1) & R_{uu}(1) & \ldots & R_{uu}(q) &) \\
+\{\,(& -H_1^{p,q} & \ldots & -H_{p+1}^{p,q} & G_1^{p,q} & \ldots & G_q^{p,q} &) \\
+\,M_2^{p,q}\,(& B_p^{p,q} & \ldots & B_1^{p,0} & -D_q^{p,q} & \ldots & -D_1^{p,q} &)\,\}R^3(p+1,q) \\
=(& 0 & \ldots & 0 & 0 & \ldots & 0 &) \\
=(& R_{uy}(1) & \ldots & R_{uy}(p+1) & R_{uu}(1) & \ldots & R_{uu}(q) &) \\
+(& -H_1^{p+1,q} & \ldots & -H_{p+1}^{p+1,q} & G_1^{p+1,q} & \ldots & G_q^{p+1,q} &)R^3(p+1,q),
\end{array}
$$

$$
\begin{array}{lcccccccc}
 & (& R_{uy}(-q) & \ldots & R_{uy}(p-q+1) & R_{uu}(-q) & \ldots & R_{uu}(-1) &) \\
+\{ & (& -F_p^{p,q} & \ldots & -F_{-1}^{p,q} & E_q^{p,q} & \ldots & E_1^{p,q} &) \\
+N_2^{p,q} & (& B_{p+1}^{p+1,q} & \ldots & B_1^{p+1,q} & -D_q^{p+1,q} & \ldots & -D_1^{p+1,q} &)\}R^4(p+1,q) \\
= & (& 0 & \ldots & 0 & 0 & \ldots & 0 &) \\
= & (& R_{uy}(-q) & \ldots & R_{uy}(p-q+1) & R_{uu}(-q) & \ldots & R_{uu}(-1) &) \\
+ & (& -F_{p+1}^{p+1,q} & \ldots & -F_0^{p+1,q} & E_q^{p+1,q} & \ldots & E_1^{p+1,q} &)R^4(p+1,q),
\end{array}
$$

where some fictitious parameters defined in Definition 3.3 have been used. Thus, the following parameter recursions are immediately obtained because of the nonsingularity of the normal matrices appearing in Eqs. (A.2) – (A.5):

$$
\begin{array}{lcccccccc}
 & (& A_1^{p+1,q} & \ldots & A_{p+1}^{p+1,q} & -C_0^{p+1,q} & \ldots & -C_q^{p+1,q} &) \\
= & (& A_1^{p,q} & \ldots & A_{p+1}^{p,q} & -C_0^{p,q} & \ldots & -C_q^{p,q} &) \\
+K_2^{p,q} & (& B_p^{p,q} & \ldots & B_0^{p,q} & -D_{q+1}^{p,q} & \ldots & -D_1^{p,q} &) \\
+K_3^{p,q} & (& -H_1^{p,q} & \ldots & -H_{p+1}^{p,q} & G_0^{p,q} & \ldots & G_q^{p,q} &),
\end{array}
\tag{A.10}
$$

$$
\begin{array}{lcccccccc}
 & (& B_{p+1}^{p+1,q} & \ldots & B_1^{p+1,q} & -D_q^{p+1,q} & \ldots & -D_1^{p+1,q} &) \\
= & (& B_{p+1}^{p,q} & \ldots & B_1^{p,q} & -D_{q+1}^{p,q} & \ldots & -D_2^{p,q} &) \\
+L_1^{p,q} & (& A_0^{p,q} & \ldots & A_p^{p,q} & -C_0^{p,q} & \ldots & C_{q-1}^{p,q} &) \\
+L_3^{p,q} & (& -H_1^{p,0} & \ldots & -H_p^{p,q} & G_0^{p,q} & \ldots & G_{q-1}^{p,q} &) \\
+L_4^{p,q} & (& -F_p^{p,q} & \ldots & -F_0^{p,q} & E_q^{p,q} & \ldots & E_1^{p,q} &),
\end{array}
\tag{A.11}
$$

$$
\begin{array}{lcccccccc}
 & (& -H_1^{p+1,q} & \ldots & -H_{p+1}^{p+1,q} & G_1^{p+1,q} & \ldots & G_q^{p+1,q} &) \\
= & (& -H_1^{p,q} & \ldots & -H_{p+1}^{p,q} & G_1^{p,q} & \ldots & G_q^{p,q} &) \\
+M_2^{p,q} & (& B_p^{p,q} & \ldots & B_1^{p,0} & -D_q^{p,q} & \ldots & -D_1^{p,q} &)\}
\end{array}
\tag{A.12}
$$

$$
\begin{array}{lcccccccc}
 & (& -F_{p+1}^{p+1,q} & \ldots & -F_0^{p+1,q} & E_q^{p+1,q} & \ldots & E_1^{p+1,q} &) \\
= & (& -F_p^{p,q} & \ldots & -F_{-1}^{p,q} & E_q^{p,q} & \ldots & E_1^{p,q} &) \\
+N_2^{p,q} & (& B_{p+1}^{p+1,q} & \ldots & B_1^{p+1,q} & -D_q^{p+1,q} & \ldots & -D_1^{p+1,q} &).
\end{array}
\tag{A.13}
$$

Eqs.(A.10) – (A.13) immediately result in Eq.(3.39).

A.2 The Derivation of the ORA in Q

The derivation of the ORA in q is very similar to that of the ORA in p. By careful checking Eq.(3.32) and Eq.(3.38) of order $(p, q+1)$, we have the following normal equations:

$$
\begin{array}{rcccccccl}
(& R_{yy}(1) & \dots & R_{yy}(p) & R_{yu}(0) & \dots & R_{yu}(q+1) &) & \\
+(& A_1^{p,q+1} & \dots & A_p^{p,q+1} & -C_0^{p,q+1} & \dots & -C_{q+1}^{p,q+1} &) & R^1(p,q+1) \\
=(& 0 & \cdots & 0 & 0 & \cdots & 0 &) & \\
=(& 0 & \cdots & 0 & 0 & \cdots & \gamma(p,q) &) & \\
+P_3^{p,q}(& 0 & \dots & 0 & V^f(p,q) & \cdots & \kappa(p,q) &) & \\
+P_4^{p,q}(& 0 & \dots & 0 & \xi(p-1,q) & \cdots & V^b(p-1,q) &) & \\
=(& R_{yy}(1) & \dots & R_{yy}(p) & R_{yu}(0) & \dots & R_{yu}(q+1) &) & \\
+\{(& A_1^{p,q} & \cdots & A_p^{p,q} & -C_0^{p,q} & \cdots & -C_{q+1}^{p,q} &) & \\
+P_3^{p,q}(& -H_1^{p,q} & \cdots & -H_p^{p,q} & G_0^{p,q} & \cdots & G_{q+1}^{p,q} &) & \\
+P_4^{p,q}(& -F_{p-1}^{p-1,q} & \cdots & -F_0^{p-1,q} & E_{q+1}^{p-1q} & \cdots & E_0^{p-1,q} &)\} & R^1(p,q+1),
\end{array}
$$

$$
\begin{array}{rcccccccl}
(& R_{uy}(1) & \dots & R_{uy}(p) & R_{uu}(1) & \dots & R_{uu}(q+1) &) & \\
+(& -H_1^{p,q+1} & \dots & -H_p^{p,q+1} & G_1^{p,q+1} & \dots & G_{q+1}^{p,q+1} &) & R^3(p,q+1) \\
=(& 0 & \cdots & 0 & 0 & \cdots & 0 &) & \\
=(& 0 & \cdots & 0 & 0 & \cdots & \kappa(p,q) &) & \\
+S_4^{p,q}(& 0 & \dots & 0 & 0 & \cdots & V^b(p-1,q) &) & \\
=(& R_{uy}(1) & \dots & R_{uy}(p) & R_{uu}(1) & \dots & R_{uu}(q+1) &) & \\
+\{(& -H_1^{p,q} & \cdots & -H_p^{p,q} & G_1^{p,q} & \cdots & G_{q+1}^{p,q} &) & \\
+S_4^{p,q}(& -F_{p-1}^{p-1,q} & \cdots & -F_0^{p-1,q} & E_q^{p-1,q} & \cdots & E_0^{p-1,q} &)\} & R^3(p,q+1),
\end{array}
$$

$$
\begin{array}{rcccccccl}
(& R_{uy}(-q-1) & \dots & R_{uy}(p-q-2) & R_{uu}(-q-1) & \dots & R_{uu}(-1) &) & \\
+(& -F_{p-1}^{p-1,q+1} & \dots & -F_0^{p-1,q+1} & E_{q+1}^{p-1,q+1} & \dots & E_1^{p-1,q+1} &) & R^2(p,q+1) \\
=(& 0 & \cdots & 0 & 0 & \cdots & 0 &) & \\
=(& \mu(p-1,q) & \cdots & 0 & \xi(p-1,q) & \cdots & 0 &) & \\
+T_1^{p,q}(& R^f(p,q) & \cdots & 0 & 0 & \cdots & 0 &) & \\
+T_2^{p,q}(& 0 & \cdots & 0 & 0 & \cdots & 0 &) & \\
+T_3^{p,q}(& \eta(p,q) & \cdots & 0 & V^f(p,q) & \cdots & 0 &) & \\
=(& R_{uy}(-q-1) & \dots & R_{uy}(p-q-2) & R_{uu}(-q-1) & \dots & R_{uu}(-1) &) & \\
+\{(& -F_p^{p-1,q} & \cdots & -F_0^{p-1,q} & E_q^{p-1,q} & \cdots & E_1^{p-1,q} &) & \\
+T_1^{p,q}(& A_0^{p,q} & \cdots & A_{p-1}^{p,q} & -C_0^{p,q} & \cdots & -C_q^{p,q} &) & \\
+T_2^{p,q}(& B_p^{p,q+1} & \cdots & B_1^{p,q+1} & -D_{q+1}^{p,q+1} & \cdots & -D_1^{p,q+1} &) & \\
+T_3^{p,q}(& -H_0^{p,q} & \cdots & -H_{p-1}^{p,q} & G_0^{p,q} & \cdots & G_q^{p,q} &)\} & R^2(p,q+1),
\end{array}
$$

$$
\begin{array}{llllllll}
(& R_{yy}(-p) & \cdots & R_{yy}(-1) & R_{yu}(-p) & \cdots & R_{yu}(-p+q+1) &) \\
+(& B_p^{p,q+2} & \cdots & B_1^{p,q+2} & -D_{q+2}^{p,q+2} & \cdots & -D_1^{p,q+2} &)R^4(p,q+2) \\
=(& 0 & \cdots & 0 & 0 & \cdots & 0 &) \\
=(& B_p^{p,q+1} & \cdots & B_1^{p,q+1} & -D_{q+1}^{p,q+1} & \cdots & -D_0^{p,q+1} &) \\
+(& -F_{p-1}^{p-1,q+1} & \cdots & -F_0^{p-1,q+1} & E_{q+1}^{p-1,q+1} & \cdots & E_0^{p-1,q+1} &)R^4(p,q+2),
\end{array}
$$

where $R^1(p,q+1), R^2(p,q+1)$, $R^3(p,q+1)$, and $R^4(p,q+2)$ are defined in Definition 3.2. To relate the solution to Eq. (3.32) with that of order $(p,q+1)$, we introduce the partial correlation coefficients, $P_3^{p,q} \in \mathcal{R}^{m\times l}$, $P_4^{p,q} \in \mathcal{R}^{m\times l}$, $S_4^{p,q} \in \mathcal{R}^{l\times l}$, $T_1^{p,q} \in \mathcal{R}^{l\times m}$, $T_2^{p,q} \in \mathcal{R}^{l\times m}$, $T_3^{p,q} \in \mathcal{R}^{l\times l}$, and $Q_4^{p,q} \in \mathcal{R}^{m\times l}$, that are defined as the solution to the following equations:

$$
\left.\begin{array}{l}
P_3^{p,q}V^f(p,q) + P_4^{p,q}\xi(p-1,q) = 0 \\
P_3^{p,q}\kappa(p,q) + P_4^{p,q}V^b(p-1,q) + \gamma(p,q) = 0
\end{array}\right\} \tag{A.14}
$$

$$
S_4^{p,q}V^b(p-1,q) + \kappa(p,q) = 0, \tag{A.15}
$$

$$
\left.\begin{array}{l}
T_1^{p,q}R^f(p,q) + T_3^{p,q}\eta(p,q) + \mu(p-1,q) = 0 \\
T_3^{p,q}V^f(p,q) + \xi(p-1,q) = 0 \\
T_2^{p,q} + T_1^{p,q}A_p^{p,q} - T_3^{p,q}H_p^{p,q} - F_0^{p-1,q} = 0
\end{array}\right\} \tag{A.16}
$$

$$
Q_4^{p,q}V^b(p-1,q+1) + \nu(p,q+1) = 0, \tag{A.17}
$$

where $(p,q) \in \Xi(p^*,q^*)$. It follows from Theorem 3.1 that if matrices $\Gamma(p,q)$, $(p,q) \in \Xi(p^*,q^*)$, are nonsingular, there exist unique solutions to Eqs.(A.14)–(A.17) which are expressed in Eq.(3.51). Based on some arguments similar to those in Section A.1 and Theorem 3.1, we have the ORA in q described in Eq.(3.50).

Appendix B

Some Proofs on the Relation between TORA and LS Parameter Estimates

The proof of Lemmas 4.1, 4.2, and 4.3 is lengthy and tedious. The only technique involved in the proof is algebraic manipulation. The proof will be carried out by considering two cases: $p \geq q$ and $p < q$. The ideas for proving the lemmas are same in both cases. On the other hand, each of the lemmas expresses four identities. For each lemma, the proof of one identity is similar to that of the other three. Therefore, for simplicity, we will only present a detailed proof of one identity for each lemma and only the case of $p \geq q$ is considered in the proofs.

B.1 The Proof of Lemma 4.1

The aim of this section is to prove

$$\begin{aligned}\tilde{R}^{1}_{\mathrm{N}}(p,q) &= \frac{1}{N}\sum_{k=1}^{N}\phi_{\mathrm{f,k\text{-}1}}(p,q)\phi^{\mathrm{T}}_{\mathrm{f,k\text{-}1}}(p,q)+\frac{1}{Nr_0}I+\frac{1}{N}\sum_{i=0}^{p-1}f_{\mathrm{N,1}}(i,p,q)f^{\mathrm{T}}_{\mathrm{N,1}}(i,p,q)\\ &= \frac{1}{N}(P^{y}_{\mathrm{f,N}}(p,q))^{-1}+\frac{1}{N}\sum_{i=0}^{p-1}f_{\mathrm{N,1}}(i,p,q)f^{\mathrm{T}}_{\mathrm{N,1}}(i,p,q).\end{aligned} \tag{B.1}$$

Actually, it suffices to prove the first equality because the second one comes immediately from the first equality and (4.16). Before proving the identity (B.1), let us consider an example of $\tilde{R}^1_{\mathrm{N}}(3,2)$. By (4.17),

$$\tilde{R}^1_{\mathrm{N}}(3,2)=\frac{1}{Nr_0}I+\frac{1}{N}\sum_{k=1}^{N}\begin{pmatrix} \mathbf{y}_k\mathbf{y}_k^T & \mathbf{y}_k\mathbf{y}_{k-1}^T & \mathbf{y}_k\mathbf{y}_{k-2}^T & \mathbf{y}_{k-1}\mathbf{u}_k^T & \mathbf{y}_k\mathbf{u}_k^T & \mathbf{y}_k\mathbf{u}_{k-1}^T \\ \mathbf{y}_{k-1}\mathbf{y}_k^T & \mathbf{y}_k\mathbf{y}_k^T & \mathbf{y}_k\mathbf{y}_{k-1}^T & \mathbf{y}_{k-2}\mathbf{u}_k^T & \mathbf{y}_{k-1}\mathbf{u}_k^T & \mathbf{y}_k\mathbf{u}_k^T \\ \mathbf{y}_{k-2}\mathbf{y}_k^T & \mathbf{y}_{k-1}\mathbf{y}_k^T & \mathbf{y}_k\mathbf{y}_k^T & \mathbf{y}_{k-3}\mathbf{u}_k^T & \mathbf{y}_{k-2}\mathbf{u}_k^T & \mathbf{y}_{k-1}\mathbf{u}_k^T \\ \mathbf{u}_k\mathbf{y}_{k-1}^T & \mathbf{u}_k\mathbf{y}_{k-2}^T & \mathbf{u}_k\mathbf{y}_{k-3}^T & \mathbf{u}_k\mathbf{u}_k^T & \mathbf{u}_k\mathbf{u}_{k-1}^T & \mathbf{u}_k\mathbf{u}_{k-2}^T \\ \mathbf{u}_k\mathbf{y}_k^T & \mathbf{u}_k\mathbf{y}_{k-1}^T & \mathbf{u}_k\mathbf{y}_{k-2}^T & \mathbf{u}_{k-1}\mathbf{u}_k^T & \mathbf{u}_k\mathbf{u}_k^T & \mathbf{u}_k\mathbf{u}_{k-1}^T \\ \mathbf{u}_{k-1}\mathbf{y}_k^T & \mathbf{u}_k\mathbf{y}_k^T & \mathbf{u}_k\mathbf{y}_{k-1}^T & \mathbf{u}_{k-2}\mathbf{u}_k^T & \mathbf{u}_{k-1}\mathbf{u}_k^T & \mathbf{u}_k\mathbf{u}_k^T \end{pmatrix}.$$

This matrix can be decomposed into the sum of a normal matrix

$$\frac{1}{N}\sum_{k=1}^{N}\phi_{\mathrm{f},\mathrm{k-1}}(3,2)\phi^T_{\mathrm{f},\mathrm{k-1}}(3,2)$$

and some other matrices, which will converge to zero as the sampling time approaches infinity. Specifically,

$$\tilde{R}^1_{\mathrm{N}}(3,2)=\tfrac{1}{Nr_0}I+\tfrac{1}{N}\sum_{k=1}^{N}\begin{pmatrix} \mathbf{y}_{k-1}\mathbf{y}_{k-1}^T & \mathbf{y}_{k-1}\mathbf{y}_{k-2}^T & \mathbf{y}_{k-1}\mathbf{y}_{k-3}^T & \mathbf{y}_{k-1}\mathbf{u}_k^T & \mathbf{y}_{k-1}\mathbf{u}_{k-1}^T & \mathbf{y}_{k-1}\mathbf{u}_{k-2}^T \\ \mathbf{y}_{k-2}\mathbf{y}_{k-1}^T & \mathbf{y}_{k-2}\mathbf{y}_{k-2}^T & \mathbf{y}_{k-2}\mathbf{y}_{k-3}^T & \mathbf{y}_{k-2}\mathbf{u}_k^T & \mathbf{y}_{k-2}\mathbf{u}_{k-1}^T & \mathbf{y}_{k-2}\mathbf{u}_{k-2}^T \\ \mathbf{y}_{k-3}\mathbf{y}_{k-1}^T & \mathbf{y}_{k-3}\mathbf{y}_{k-2}^T & \mathbf{y}_{k-3}\mathbf{y}_{k-3}^T & \mathbf{y}_{k-3}\mathbf{u}_k^T & \mathbf{y}_{k-3}\mathbf{u}_{k-1}^T & \mathbf{y}_{k-3}\mathbf{u}_{k-2}^T \\ \mathbf{u}_k\mathbf{y}_{k-1}^T & \mathbf{u}_k\mathbf{y}_{k-2}^T & \mathbf{u}_k\mathbf{y}_{k-3}^T & \mathbf{u}_k\mathbf{u}_k^T & \mathbf{u}_k\mathbf{u}_{k-1}^T & \mathbf{u}_k\mathbf{u}_{k-2}^T \\ \mathbf{u}_{k-1}\mathbf{y}_{k-1}^T & \mathbf{u}_{k-1}\mathbf{y}_{k-2}^T & \mathbf{u}_{k-1}\mathbf{y}_{k-3}^T & \mathbf{u}_{k-1}\mathbf{u}_k^T & \mathbf{u}_{k-1}\mathbf{u}_{k-1}^T & \mathbf{u}_{k-1}\mathbf{u}_{k-2}^T \\ \mathbf{u}_{k-2}\mathbf{y}_{k-1}^T & \mathbf{u}_{k-2}\mathbf{y}_{k-2}^T & \mathbf{u}_{k-2}\mathbf{y}_{k-3}^T & \mathbf{u}_{k-2}\mathbf{u}_k^T & \mathbf{u}_{k-2}\mathbf{u}_{k-1}^T & \mathbf{u}_{k-2}\mathbf{u}_{k-2}^T \end{pmatrix}$$

$$+\tfrac{1}{N}\begin{pmatrix} \mathbf{y}_N\mathbf{y}_N^T & \mathbf{y}_N\mathbf{y}_{N-1}^T & \mathbf{y}_N\mathbf{y}_{N-2}^T & \mathbf{0} & \mathbf{y}_N\mathbf{u}_N^T & \mathbf{y}_N\mathbf{u}_{N-1}^T \\ \mathbf{y}_{N-1}\mathbf{y}_N^T & \mathbf{y}_N\mathbf{y}_N^T & \mathbf{y}_N\mathbf{y}_{N-1}^T & \mathbf{0} & \mathbf{y}_{N-1}\mathbf{u}_N^T & \mathbf{y}_N\mathbf{u}_N^T \\ \mathbf{y}_{N-2}\mathbf{y}_N^T & \mathbf{y}_{N-1}\mathbf{y}_N^T & \mathbf{y}_N\mathbf{y}_N^T & \mathbf{0} & \mathbf{y}_{N-2}\mathbf{u}_N^T & \mathbf{y}_{N-1}\mathbf{u}_N^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{u}_N\mathbf{y}_N^T & \mathbf{u}_N\mathbf{y}_{N-1}^T & \mathbf{u}_N\mathbf{y}_{N-2}^T & \mathbf{0} & \mathbf{u}_N\mathbf{u}_N^T & \mathbf{u}_N\mathbf{u}_{N-1}^T \\ \mathbf{u}_{N-1}\mathbf{y}_N^T & \mathbf{u}_N\mathbf{y}_N^T & \mathbf{u}_N\mathbf{y}_{N-1}^T & \mathbf{0} & \mathbf{u}_{N-1}\mathbf{u}_N^T & \mathbf{u}_N\mathbf{u}_N^T \end{pmatrix}$$

$$+\tfrac{1}{N}\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{y}_{N-1}\mathbf{y}_{N-1}^T & \mathbf{y}_{N-1}\mathbf{y}_{N-2}^T & \mathbf{0} & \mathbf{0} & \mathbf{y}_{N-1}\mathbf{u}_{N-1}^T \\ \mathbf{0} & \mathbf{y}_{N-2}\mathbf{y}_{N-1}^T & \mathbf{y}_{N-1}\mathbf{y}_{N-1}^T & \mathbf{0} & \mathbf{0} & \mathbf{y}_{N-2}\mathbf{u}_{N-1}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{u}_{N-1}\mathbf{y}_{N-1}^T & \mathbf{u}_{N-1}\mathbf{y}_{N-2}^T & \mathbf{0} & \mathbf{0} & \mathbf{u}_{N-1}\mathbf{u}_{N-1}^T \end{pmatrix}$$

$$+\frac{1}{N}\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{y}_{N-2}\mathbf{y}_{N-2}^{T} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

$$-\frac{1}{N}\begin{pmatrix} \mathbf{y}_0\mathbf{y}_0^T & \mathbf{y}_0\mathbf{y}_{-1}^T & \mathbf{y}_0\mathbf{y}_{-2}^T & \mathbf{0} & \mathbf{y}_0\mathbf{u}_0^T & \mathbf{y}_0\mathbf{u}_{-1}^T \\ \mathbf{y}_{-1}\mathbf{y}_0^T & \mathbf{y}_0\mathbf{y}_0^T+\mathbf{y}_{-1}\mathbf{y}_{-1}^T & \mathbf{y}_0\mathbf{y}_{-1}^T+\mathbf{y}_{-1}\mathbf{y}_{-2}^T & \mathbf{0} & \mathbf{y}_{-1}\mathbf{u}_0^T & \mathbf{y}_0\mathbf{u}_0^T+\mathbf{y}_{-1}\mathbf{u}_{-1}^T \\ \mathbf{y}_{-2}\mathbf{y}_0^T & \mathbf{y}_{-1}\mathbf{y}_0^T+\mathbf{y}_{-2}\mathbf{y}_{-1}^T & \mathbf{y}_0\mathbf{y}_0^T+\mathbf{y}_{-1}\mathbf{y}_{-1}^T+\mathbf{y}_{-2}\mathbf{y}_{-2}^T & \mathbf{0} & \mathbf{y}_{-2}\mathbf{u}_0^T & \mathbf{y}_{-1}\mathbf{u}_0^T+\mathbf{y}_{-2}\mathbf{u}_{-1}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{u}_0\mathbf{y}_0^T & \mathbf{u}_0\mathbf{y}_{-1}^T & \mathbf{u}_0\mathbf{y}_{-2}^T & \mathbf{0} & \mathbf{u}_0\mathbf{u}_0^T & \mathbf{u}_0\mathbf{u}_{-1}^T \\ \mathbf{u}_{-1}\mathbf{y}_0^T & \mathbf{u}_0\mathbf{y}_0^T+\mathbf{u}_{-1}\mathbf{y}_{-1}^T & \mathbf{u}_0\mathbf{y}_{-1}^T+\mathbf{u}_0\mathbf{y}_{-2}^T & \mathbf{0} & \mathbf{u}_{-1}\mathbf{u}_0^T & \mathbf{u}_0\mathbf{u}_0^T+\mathbf{u}_{-1}\mathbf{u}_{-1}^T \end{pmatrix}.$$

The last matrix is equal to a zero matrix because of the one-sided assumption that

$$\mathbf{y}_{-k} = 0, k \geq 0$$

and

$$\mathbf{u}_{-k} = 0, k \geq 0.$$

Thus, it follows from the definition of $f_{N,1}(i,p,q)$ at the begining of Section 4.2.3 that

$$\begin{aligned} \tilde{R}_N^1(3,2) &= \frac{1}{Nr_0}I \\ &+\frac{1}{N}\sum_{k=1}^{N}(\ \mathbf{y}_{k-1}^T\ \mathbf{y}_{k-2}^T\ \mathbf{y}_{k-3}^T\ \mathbf{u}_k^T\ \mathbf{u}_{k-1}^T\ \mathbf{u}_{k-2}^T\ )^T(\ \mathbf{y}_{k-1}^T\ \mathbf{y}_{k-2}^T\ \mathbf{y}_{k-3}^T\ \mathbf{u}_k^T\ \mathbf{u}_{k-1}^T\ \mathbf{u}_{k-2}^T\ ) \\ &+\frac{1}{N}(\ \mathbf{y}_N^T\ \mathbf{y}_{N-1}^T\ \mathbf{y}_{N-2}^T\ \mathbf{0}^T\ \mathbf{u}_N^T\ \mathbf{u}_{N-1}^T\ )^T(\ \mathbf{y}_N^T\ \mathbf{y}_{N-1}^T\ \mathbf{y}_{N-2}^T\ \mathbf{0}^T\ \mathbf{u}_N^T\ \mathbf{u}_{N-1}^T\ ) \\ &+\frac{1}{N}(\ \mathbf{0}^T\ \mathbf{y}_N^T\ \mathbf{y}_{N-1}^T\ \mathbf{0}^T\ \mathbf{0}^T\ \mathbf{u}_N^T\ )^T(\ \mathbf{0}^T\ \mathbf{y}_N^T\ \mathbf{y}_{N-1}^T\ \mathbf{0}^T\ \mathbf{0}^T\ \mathbf{u}_N^T\ ) \\ &+\frac{1}{N}(\ \mathbf{0}^T\ \mathbf{0}^T\ \mathbf{y}_N^T\ \mathbf{0}^T\ \mathbf{0}^T\ \mathbf{0}^T\ )^T(\ \mathbf{0}^T\ \mathbf{0}^T\ \mathbf{y}_N^T\ \mathbf{0}^T\ \mathbf{0}^T\ \mathbf{0}^T\ ) \\ &= \frac{1}{N}\sum_{k=1}^{N}\phi_{f,k-1}(3,2)\phi_{f,k-1}^T(3,2) + \frac{1}{N}\sum_{i=0}^{3-1} f_{N,1}(i,3,2)f_{N,1}^T(i,3,2). \end{aligned}$$

Now let us begin the proof for the general case of $p \geq q$. By (4.7),

$\tilde{R}^1_{\rm N}(p,q) = \frac{1}{Nr_0} I$

$$+\frac{1}{N}\sum_{k=1}^{N}\begin{pmatrix} \mathbf{y}_k\mathbf{y}_k^T & \cdots & \mathbf{y}_k\mathbf{y}_{k-q+1}^T & \cdots & \mathbf{y}_k\mathbf{y}_{k-p+1}^T & \mathbf{y}_{k-1}\mathbf{u}_k^T & \cdots & \mathbf{y}_k\mathbf{u}_{k-q+1}^T \\ \vdots & \ddots & \vdots & & \vdots & \vdots & & \vdots \\ \mathbf{y}_{k-q+1}\mathbf{y}_k^T & \cdots & \mathbf{y}_k\mathbf{y}_k^T & \cdots & \mathbf{y}_k\mathbf{y}_{k-p+q}^T & \mathbf{y}_{k-q}\mathbf{u}_k^T & \cdots & \mathbf{y}_k\mathbf{u}_k^T \\ \vdots & & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \mathbf{y}_{k-p+1}\mathbf{y}_k^T & \cdots & \mathbf{y}_{k-p+q}\mathbf{y}_k^T & \cdots & \mathbf{y}_k\mathbf{y}_k^T & \mathbf{y}_{k-p}\mathbf{u}_k^T & \cdots & \mathbf{y}_{k-p+q}\mathbf{u}_k^T \\ \mathbf{u}_k\mathbf{y}_{k-1}^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-q}^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-p}^T & \mathbf{u}_k\mathbf{u}_k^T & \cdots & \mathbf{u}_k\mathbf{u}_{k-q}^T \\ \mathbf{u}_k\mathbf{y}_k^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-q+1}^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-p+1}^T & \mathbf{u}_{k-1}\mathbf{u}_k^T & \cdots & \mathbf{u}_k\mathbf{u}_{k-q+1}^T \\ \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{k-q+1}\mathbf{y}_k^T & \cdots & \mathbf{u}_k\mathbf{y}_k^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-p+q}^T & \mathbf{u}_{k-q}\mathbf{u}_k^T & \cdots & \mathbf{u}_k\mathbf{u}_k^T \end{pmatrix}.$$

To relate $\tilde{R}^1_{\rm N}(p,q)$ with the normal matrix $\frac{1}{N}\sum_{k=1}^{N}\phi_{f,k-1}(p,q)\phi^{\rm T}_{f,k-1}(p,q)$, decompose the matrix $\tilde{R}^1_{\rm N}(p,q)$ into:
$\tilde{R}^1_{\rm N}(p,q) = \frac{1}{Nr_0} I$

$$+\frac{1}{N}\sum_{k=1}^{N}\begin{pmatrix} \mathbf{y}_{k-1}\mathbf{y}_{k-1}^T & \cdots & \mathbf{y}_{k-1}\mathbf{y}_{k-q}^T & \cdots & \mathbf{y}_{k-1}\mathbf{y}_{k-p}^T & \mathbf{y}_{k-1}\mathbf{u}_k^T & \cdots & \mathbf{y}_{k-1}\mathbf{u}_{k-q}^T \\ \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{k-q}\mathbf{y}_{k-1}^T & \cdots & \mathbf{y}_{k-q}\mathbf{y}_{k-q}^T & \cdots & \mathbf{y}_{k-q}\mathbf{y}_{k-p}^T & \mathbf{y}_{k-q}\mathbf{u}_k^T & \cdots & \mathbf{y}_{k-q}\mathbf{u}_{k-q}^T \\ \vdots & & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \mathbf{y}_{k-p}\mathbf{y}_{k-1}^T & \cdots & \mathbf{y}_{k-p}\mathbf{y}_{k-q}^T & \cdots & \mathbf{y}_{k-p}\mathbf{y}_{k-p}^T & \mathbf{y}_{k-p}\mathbf{u}_k^T & \cdots & \mathbf{y}_{k-p}\mathbf{u}_{k-q}^T \\ \mathbf{u}_k\mathbf{y}_{k-1}^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-q}^T & \cdots & \mathbf{u}_k\mathbf{y}_{k-p}^T & \mathbf{u}_k\mathbf{u}_k^T & \cdots & \mathbf{u}_k\mathbf{u}_{k-q}^T \\ \mathbf{u}_{k-1}\mathbf{y}_{k-1}^T & \cdots & \mathbf{u}_{k-1}\mathbf{y}_{k-q}^T & \cdots & \mathbf{u}_{k-1}\mathbf{y}_{k-p}^T & \mathbf{u}_{k-1}\mathbf{u}_k^T & \cdots & \mathbf{u}_{k-1}\mathbf{u}_{k-q}^T \\ \vdots & \ddots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{k-q}\mathbf{y}_{k-1}^T & \cdots & \mathbf{u}_{k-q}\mathbf{y}_{k-q}^T & \cdots & \mathbf{u}_{k-q}\mathbf{y}_{k-p}^T & \mathbf{u}_{k-q}\mathbf{u}_k^T & \cdots & \mathbf{u}_{k-q}\mathbf{u}_{k-q}^T \end{pmatrix}$$

$$-\frac{1}{N}\begin{pmatrix} \mathbf{y}_0\mathbf{y}_0^T & \cdots & \mathbf{y}_0\mathbf{y}_{1-q}^T & \cdots & \mathbf{y}_0\mathbf{y}_{1-p}^T & 0 & \mathbf{y}_0\mathbf{u}_0^T & \cdots & \mathbf{y}_0\mathbf{u}_{1-q}^T \\ \vdots & \ddots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{1-q}\mathbf{y}_0^T & \cdots & \sum_{i=1}^{q}\mathbf{y}_{i-q}\mathbf{y}_{i-q}^T & \cdots & \sum_{i=1}^{q}\mathbf{y}_{i-q}\mathbf{y}_{i-p}^T & 0 & \mathbf{y}_{1-q}\mathbf{u}_0^T & \cdots & \sum_{i=1}^{q}\mathbf{y}_{i-q}\mathbf{u}_{i-q}^T \\ \vdots & & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ \mathbf{y}_{1-p}\mathbf{y}_0^T & \cdots & \sum_{i=1}^{q}\mathbf{y}_{i-p}\mathbf{y}_{i-q}^T & \cdots & \sum_{i=1}^{p}\mathbf{y}_{i-p}\mathbf{y}_{i-p}^T & 0 & \mathbf{y}_{1-p}\mathbf{u}_0^T & \cdots & \sum_{i=1}^{q}\mathbf{y}_{i-p}\mathbf{u}_{i-q}^T \\ 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \mathbf{u}_0\mathbf{y}_0^T & \cdots & \mathbf{u}_0\mathbf{y}_{1-q}^T & \cdots & \mathbf{u}_0\mathbf{y}_{1-p}^T & 0 & \mathbf{u}_0\mathbf{u}_0^T & \cdots & \mathbf{u}_0\mathbf{u}_{1-q}^T \\ \vdots & \ddots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{1-q}\mathbf{y}_0^T & \cdots & \sum_{i=1}^{q}\mathbf{u}_{i-q}\mathbf{y}_{i-q}^T & \cdots & \sum_{i=1}^{q}\mathbf{u}_{i-q}\mathbf{y}_{i-p}^T & 0 & \mathbf{u}_{1-q}\mathbf{u}_0^T & \cdots & \sum_{i=1}^{q}\mathbf{u}_{i-q}\mathbf{u}_{i-q}^T \end{pmatrix}$$

$$+\begin{pmatrix} M_{N,1}^{1,1}(p,q) & M_{N,1}^{1,2}(p,q) \\ M_{N,1}^{2,1}(p,q) & M_{N,1}^{2,2}(p,q) \end{pmatrix},$$

where

$$M_{N,1}^{1,1}(p,q) = \frac{1}{N}\begin{pmatrix} \mathbf{y}_N\mathbf{y}_N^T & \cdots & \mathbf{y}_N\mathbf{y}_{N-q+1}^T & \cdots & \mathbf{y}_N\mathbf{y}_{N-p+1}^T \\ \vdots & \ddots & \vdots & & \vdots \\ \mathbf{y}_{N-q+1}\mathbf{y}_N^T & \cdots & \sum_{i=0}^{q-1}\mathbf{y}_{N-i}\mathbf{y}_{N-i}^T & \cdots & \sum_{i=0}^{q-1}\mathbf{y}_{N-i}\mathbf{y}_{N-i-p+q}^T \\ \vdots & & \vdots & \ddots & \vdots \\ \mathbf{y}_{N-p+1}\mathbf{y}_N^T & \cdots & \sum_{i=0}^{q-1}\mathbf{y}_{N-i-p+q}\mathbf{y}_{N-i}^T & \cdots & \sum_{i=0}^{p-1}\mathbf{y}_{N-i}\mathbf{y}_{N-i}^T \end{pmatrix}$$

$$M_{N,1}^{1,2}(p,q) = \frac{1}{N}\begin{pmatrix} 0 & \mathbf{y}_N\mathbf{u}_N^T & \cdots & \mathbf{y}_N\mathbf{u}_{N-q+1}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathbf{y}_{N-q+1}\mathbf{u}_N^T & \cdots & \sum_{i=0}^{q-1}\mathbf{y}_{N-i}\mathbf{u}_{N-i}^T \\ \vdots & \vdots & & \vdots \\ 0 & \mathbf{y}_{N-p+1}\mathbf{u}_N^T & \cdots & \sum_{i=0}^{q-1}\mathbf{y}_{N-i-p+q}\mathbf{u}_{N-i}^T \end{pmatrix}$$

$$M_{N,1}^{2,1}(p,q) = \frac{1}{N}\begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \mathbf{u}_N\mathbf{y}_N^T & \cdots & \mathbf{u}_N\mathbf{y}_{N-q+1}^T & \cdots & \mathbf{u}_N\mathbf{y}_{N-p+1}^T \\ \vdots & \ddots & \vdots & & \vdots \\ \mathbf{u}_{N-q+1}\mathbf{y}_N^T & \cdots & \sum_{i=0}^{q-1}\mathbf{u}_{N-i}\mathbf{y}_{N-i}^T & \cdots & \sum_{i=0}^{q-1}\mathbf{u}_{N-i}\mathbf{y}_{N-i-p+q}^T \end{pmatrix}$$

$$M_{N,1}^{2,2}(p,q) = \frac{1}{N}\begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \mathbf{u}_N\mathbf{u}_N^T & \cdots & \mathbf{u}_N\mathbf{u}_{N-q+1}^T \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathbf{u}_{N-q+1}\mathbf{u}_N^T & \cdots & \sum_{i=0}^{q-1}\mathbf{u}_{N-i}\mathbf{u}_{N-i}^T \end{pmatrix}.$$

For ease of writing, we introduce the operator that computes the outer product for a column vector $\mathbf{x}$, which is defined as $\mathcal{M}[\mathbf{x}^{\mathrm{T}}] \triangleq \mathbf{x}\mathbf{x}^{\mathrm{T}}$. Thus, by the one-sided assumption, we have

$$\begin{aligned} \tilde{R}_{\mathrm{N}}^{1}(p,q) &= \frac{1}{N}\sum_{k=1}^{N}\mathcal{M}[(\mathbf{y}_{\mathrm{k-1}}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{k-q}}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{k-p}}^{\mathrm{T}} \ \mathbf{u}_{\mathrm{k}}^{\mathrm{T}} \ \mathbf{u}_{\mathrm{k-1}}^{\mathrm{T}} \ \cdots \ \mathbf{u}_{\mathrm{k-q}}^{\mathrm{T}})] + \frac{1}{Nr_0}I \\ &+\frac{1}{N}\mathcal{M}[(\mathbf{y}_{\mathrm{N}}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{N-q+1}}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{N-p+1}}^{\mathrm{T}} \ \mathbf{0}^{\mathrm{T}} \ \mathbf{u}_{\mathrm{N}}^{\mathrm{T}} \ \cdots \ \mathbf{u}_{\mathrm{N-q+1}}^{\mathrm{T}})] \\ &+\cdots \\ &+\frac{1}{N}\mathcal{M}[(\mathbf{0}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{N}}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{N-p+q}}^{\mathrm{T}} \ \mathbf{0}^{\mathrm{T}} \ \mathbf{0}^{\mathrm{T}} \ \cdots \ \mathbf{u}_{\mathrm{N}}^{\mathrm{T}})] \\ &+\cdots \\ &+\frac{1}{N}\mathcal{M}[(\mathbf{0}^{\mathrm{T}} \ \cdots \ \mathbf{0}^{\mathrm{T}} \ \cdots \ \mathbf{y}_{\mathrm{N}}^{\mathrm{T}} \ \mathbf{0}^{\mathrm{T}} \ \mathbf{0}^{\mathrm{T}} \ \cdots \ \mathbf{0}^{\mathrm{T}})]. \end{aligned}$$

Thus, by (4.14) and the definition of $f_{\mathrm{N},1}(i,p,q)$, we have (B.1). ∎

B.2 The Proof of Lemma 4.2

The purpose of the section is to prove the identity

$$\tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)\tilde{R}^2_{\mathrm{N}}(p,q) = -\frac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k\text{-}p}}\phi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q) - \frac{1}{N}\sum_{i=0}^{p-1}\mathbf{y}_{\mathrm{N\text{-}(p\text{-}1)+i}}f^{\mathrm{T}}_{\mathrm{N},2}(i,p,q). \quad \text{(B.2)}$$

To get some intuition, let us consider a special case of $p = 5$ and $q = 3$. By (4.18), we have

$$\begin{aligned}
\tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(5,3)\tilde{R}^2_{\mathrm{N}}(5,3) &= -\tfrac{1}{N}\sum_{k=1}^{N}(\ \mathbf{y}_{\mathrm{k\text{-}5}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}4}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}3}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}2}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}1}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}5}}\mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}4}}\mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}3}}\mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ )\\
&= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k\text{-}5}}(\ \mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{k\text{-}1}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{k\text{-}2}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{k\text{-}3}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{k\text{-}4}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{k\text{-}1}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{k\text{-}2}}\ )\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}4}}(\ 0\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}2}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}3}}\ \ 0\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ )\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}3}}(\ 0\ \ 0\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}2}}\ \ 0\ \ 0\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ )\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}2}}(\ 0\ \ 0\ \ 0\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}1}}\ \ 0\ \ 0\ \ 0\ )\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}1}}(\ 0\ \ 0\ \ 0\ \ 0\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \ 0\ \ 0\ \ 0\ ),
\end{aligned}$$

where we have used the one-sided assumption. By the definition of $\phi_{\mathrm{b,k\text{-}1}}(p,q)$ in (4.14) and $f_{\mathrm{N},2}(i,p,q)$ at the begining of Section 4.2.3, we have

$$\tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(5,3)\tilde{R}^2_{\mathrm{N}}(5,3) = -\frac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k\text{-}5}}\phi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(5,3) + \frac{1}{N}\sum_{i=0}^{5-2}\mathbf{y}_{\mathrm{N\text{-}(5\text{-}1)+i}}f^{\mathrm{T}}_{\mathrm{N},2}(i,5,3).$$

For the general case, we have $\tilde{\theta}^{\mathrm{T}}_{\mathrm{b,N}}(p,q)\tilde{R}^2_{\mathrm{N}}(p,q)$

$$\begin{aligned}
&= -\tfrac{1}{N}\sum_{k=1}^{N}(\ \mathbf{y}_{\mathrm{k\text{-}p}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \cdots\ \mathbf{y}_{\mathrm{k\text{-}p+q\text{-}1}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \cdots\ \mathbf{y}_{\mathrm{k\text{-}1}}\mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \mathbf{y}_{\mathrm{k\text{-}p}}\mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ \cdots\ \mathbf{y}_{\mathrm{k\text{-}p+q\text{-}1}}\mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ )\\
&= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k\text{-}p}}(\ \mathbf{y}^{\mathrm{T}}_{\mathrm{k}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{k\text{-}(q\text{-}1)}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{k\text{-}(p\text{-}1)}}\ \mathbf{u}^{\mathrm{T}}_{\mathrm{k}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{k\text{-}(q\text{-}1)}}\ )\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}(p\text{-}1)}}(\ \mathbf{0}^{\mathrm{T}}\ \ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}(q\text{-}2)}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}(p\text{-}2)}}\ \ \mathbf{0}^{\mathrm{T}}\ \ \mathbf{u}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N\text{-}(q\text{-}2)}}\ )\\
&\qquad -\cdots\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}(p\text{-}q)}}(\ \mathbf{0}^{\mathrm{T}}\ \ \mathbf{0}^{\mathrm{T}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N\text{-}(p\text{-}q\text{-}1)}}\ \ \mathbf{0}^{\mathrm{T}}\ \ \mathbf{0}^{\mathrm{T}}\ \cdots\ \mathbf{u}^{\mathrm{T}}_{\mathrm{N}}\ )\\
&\qquad -\cdots\\
&\quad -\tfrac{1}{N}\mathbf{y}_{\mathrm{N\text{-}1}}(\ \mathbf{0}^{\mathrm{T}}\ \ \mathbf{0}^{\mathrm{T}}\ \cdots\ \mathbf{0}^{\mathrm{T}}\ \cdots\ \mathbf{y}^{\mathrm{T}}_{\mathrm{N}}\ \ \mathbf{0}^{\mathrm{T}}\ \ \mathbf{0}^{\mathrm{T}}\ \cdots\ \mathbf{0}^{\mathrm{T}}\ )\\
&= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k\text{-}p}}\phi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q) - \tfrac{1}{N}\sum_{i=0}^{p-2}\mathbf{y}_{\mathrm{N\text{-}(p\text{-}1)+i}}f^{\mathrm{T}}_{\mathrm{N},2}(i,p,q)\\
&= -\tfrac{1}{N}\sum_{k=1}^{N}\mathbf{y}_{\mathrm{k\text{-}p}}\phi^{\mathrm{T}}_{\mathrm{b,k\text{-}1}}(p,q) - \tfrac{1}{N}\sum_{i=0}^{p-1}\mathbf{y}_{\mathrm{N\text{-}(p\text{-}1)+i}}f^{\mathrm{T}}_{\mathrm{N},2}(i,p,q).
\end{aligned}$$

The last equality comes from the fact that $f^{\mathrm{T}}_{\mathrm{N},2}(p-1,p,q)=0$. ∎

B.3 The Proof of Lemma 4.3

Comparing the right side of the second normal equation in (4.22) with the right side of Eq. (B.2) yields immediately the second identity in (4.23). ∎

Appendix C

Some Useful Tools for Martingale Analysis

The martingale technique has increasingly wide applications to linear modeling. Many useful results on linear modeling has been obtained assuming that a system can be described by a linear regression model with martingale stochastic noise. The martingale assumption on the noise is less restrictive to modeling problems, especially those related with feedback systems, than the stationary or ergodic assumption on the output of the systems in question. The stationary or ergodic assumption on output is usually not satisfied for feedback control systems. However, the assumption of martingale stochastic noise is true for many closed-loop systems. For example, consider a stochastic control system:

$$\mathbf{y}_n + A_1\mathbf{y}_{n-1} + \cdots + A_p\mathbf{y}_{n-p} = C_0\mathbf{u}_n + C_1\mathbf{u}_{n-1} + \cdots + C_q\mathbf{u}_{n-q} + \omega_n \qquad (C.1)$$

with initial condition: $\{\mathbf{y}_{-1}\ldots\mathbf{y}_{-p}\ \mathbf{u}_0\ \mathbf{u}_{-1}\ldots\mathbf{u}_{-q}\}$. The system input $\mathbf{u}_n$ at time $n \geq 1$ is a measurable function of past system outputs and the current reference signals, i.e., $\mathbf{u}_n = \mathbf{c}(\{\mathbf{y}_t\}_{t=-p}^{n-1}; \mathbf{r}_n)$, where $\mathbf{r}_n, n \geq 1$, are reference signals. Suppose that (i). the initial condition is independent of noise $\{\omega_n\}, n \in Z_+$. (ii). the reference signals $\mathbf{r}_n, n \in Z_+$, are independent of noise $\{\omega_n\}, n \in Z_+$. It follows from Eq.(C.1) that

$$\mathbf{y}_t = \mathbf{f}_t(\mathbf{y}_{-1}\ldots\mathbf{y}_{-p}\ \mathbf{u}_0\ldots\mathbf{u}_{-q}\ \omega_t\ldots\omega_0; \mathbf{r}_t\ldots\mathbf{r}_0), \quad 0 \leq t \leq n \qquad (C.2)$$

and

$$\mathbf{u}_t = \mathbf{g}_t(\mathbf{y}_{-1} \dots \mathbf{y}_{-p}\ \mathbf{u}_0 \dots \mathbf{u}_{-q}\ \omega_t \dots \omega_0; \mathbf{r}_t \dots \mathbf{r}_0), \quad 1 \leq t \leq n, \tag{C.3}$$

where $\mathbf{f}_t$ and $\mathbf{g}_t$ are measurable functions. Therefore,

$$\begin{aligned} E\omega_{n+1}|\mathcal{F}_n) &= E\omega_{n+1}|\mathcal{F}_n(\mathbf{y}_n \dots \mathbf{y}_{-p}\ \mathbf{u}_n \dots \mathbf{u}_{-q}\ \omega_n \dots \omega_0)) \\ &= E\omega_{n+1} \qquad \text{(by (C.2) and (C.3) and assumptions } (i) \text{ and } (ii)) \\ &= 0. \end{aligned}$$

Thus, a sequence of *independent* random variables $\{\omega_n\}$ with zero mean is a martingale difference process w.r.t. an increasing family of σ-fields $\{\mathcal{F}_n\}$ generated by available output/input measurements at time n and noise $\omega_i,\ i \leq n$.

Remark C.1 The regression vector $\phi_{n-1}^T = (\mathbf{y}_{n-1} \dots \mathbf{y}_{n-p}\ \mathbf{u}_n \dots \mathbf{u}_{n-q})$ is measurable w.r.t. $\mathcal{F}_{n-1},\ n \in Z_1$. ∎

Remark C.2 A sequence of independent random variables with zero-mean is a martingale difference process. But a sequence of *uncorrelated* random variables with zero-means is usually not. ∎

So far, no unified approach exists for solving linear modeling problems via martingale analysis. Many methods have been developed to solve various estimation or identification problems via martingale analysis. However, there are some conclusions (lemmas) which are often used in deriving those methods. To know them is very helpful for people to understand previous martingale analysis results and establish new results. In this paper we will use the l_2-norm for vectors and the spectral norm for matrices($\|M\|_2 \triangleq \lambda_{max}(M^T M)$). As a convention in mathematical analysis, we denote $\lim_{t\to\infty} a_t/b_t = 1$ by $a_t \sim b_t$, $\lim_{t\to\infty} a_t/b_t = 0$ by $a_t = \mathbf{o}(b_t)$, and $\lim_{t\to\infty} \sup a_t/b_t < \infty$ by $a_t = \mathbf{O}(b_t)$, where $\{a_t\}_{t=1}^{\infty}$ and $\{b_t\}_{t=1}^{\infty}$ are any two sequences. Specifically, $a_t = \mathbf{o}(1)$ implies $\lim_{t\to\infty} a_t = 0$ and $a_t = \mathbf{O}(1)$ represents $\lim_{t\to\infty} \sup a_t < \infty$.

C.1 Algebraic Tools

Lemma C.1 [Lai and Wei]

1. *Let $V_t, t \in Z_+$, be matrices and $\phi_t, t \in Z_+$, be vectors. If $V_t = V_{t-1} + \phi_{t-1}\phi_{t-1}^T$ is nonsingular for some $t \in Z_+$, then*

$$\phi_{t-1}^T V_t^{-1} \phi_{t-1} = \frac{\det(V_t) - \det(V_{t-1})}{\det(V_t)}. \tag{C.4}$$

2. *Let $\phi_0, \phi_1, \ldots$ be $p \times 1$ vectors and $V_t = \sum_{i=1}^{t} \phi_{i-1}\phi_{i-1}^T$. Let $\lambda_{max}(t)$ denote the maximum eigenvalue of matrix V_t. Assume that V_t is nonsingular for some N_z. Then $\lambda_{max}(t)$ is nondecreasing and V_t is nonsingular for all $t \geq N_z$. Moreover,*

$$\sum_{t=N_z}^{\infty} \phi_{t-1}^T V_t^{-1} \phi_{t-1} < \infty \tag{C.5}$$

if $\lim_{t\to\infty} \lambda_{max}(t) < \infty$. *On the other hand,*

$$\sum_{t=N_z}^{\infty} \phi_{t-1}^T V_t^{-1} \phi_{t-1} = \mathbf{O}(\log \lambda_{max}(t)) \tag{C.6}$$

if $\lim_{t\to\infty} \lambda_{max}(t) = \infty$.

∎

Lemma C.2 [Grenander, [24]]

1. *Let $\{a_t, t \in Z_+\}$ be a sequence of positive numbers. If the partial sums $\{b_t = \sum_{i=0}^{t} a_i, t \in Z_+\}$ are divergent, then*

$$\sum_{t=0}^{\infty} a_t / b_t = \infty \tag{C.7}$$

with $\sum_{t=0}^{n} a_t/b_t = \mathbf{O}(\log b_n)$. *In addition,*

$$\sum_{t=0}^{\infty} a_t / b_t^{1+\epsilon} < \infty \tag{C.8}$$

for $\forall \epsilon > 0$.

2. *Let $\{A_t, t \in Z_+\}$ be a sequence of matrices such that $\sum_{t=0}^{\infty} b_t^{-1} A_t$ converges to a finite limit, where $\{b_t\}$ is a divergent sequence of positive numbers. Then,*

$$\frac{1}{b_n} \sum_{t=1}^{n} A_t \to 0 \quad \text{as} \quad n \to \infty. \tag{C.9}$$

∎

Lemma C.3 *Let $\{a_t, t \in Z_+\}$ be a sequence of positive numbers. If the partial sums $\{b_t = \sum_{i=0}^{t} a_i\}$ are divergent and $\{x_t, t \in Z_+\}$ is a sequence of positive numbers which converges to zero, then*

$$\sum_{t=0}^{n} x_t a_t = \mathbf{o}(\sum_{t=0}^{n} a_t) \tag{C.10}$$

∎

Proof: For any given $\epsilon > 0$, $\exists \epsilon', N > 0$ such that $x_n < \epsilon' < \epsilon$, $n \geq N$. Hence, for any $n > N$,

$$0 \leq \frac{\sum_{t=0}^{n} x_t a_t}{\sum_{t=0}^{n} a_t} \leq \frac{\sum_{t=0}^{N} x_t a_t}{\sum_{t=0}^{n} a_t} + \frac{\epsilon' \sum_{t=N+1}^{n} a_t}{\sum_{t=0}^{n} a_t} \leq \frac{\sum_{t=0}^{N} x_t a_t}{\sum_{t=0}^{n} a_t} + \epsilon'.$$

Letting n big enough, we have $0 \leq \frac{\sum_{t=0}^{N} x_t a_t}{\sum_{t=0}^{n} a_t} < \epsilon$ since $\sum_{t=0}^{n} a_t \to \infty$ as $n \to \infty$.
∎

C.2 Statistical Tools

The following results deduced from Chow's theorem [1] are extensively used in martingale analysis.

Lemma C.4 [Hall and Heyde, [21]] *Suppose that $\{x_t, \mathcal{F}_t\}$ is a martingale difference process and $\{u_t\}$ is measurable w.r.t. $\mathcal{F}_t$. Then,*

(1)

$$\sum_{t=1}^{\infty} u_t^{-1} x_t \quad \text{converges} \quad a.s. \tag{C.11}$$

on the set $\Omega_1 = \{\sum_{t=1}^{\infty} u_t^{-p} E|x_t|^p|\mathcal{F}_{t-1}) < \infty\}$ if $1 \leq p \leq 2$.

(2)

$$\lim_{n\to\infty} u_n^{-1} \sum_{t=1}^{n} x_t = 0 \quad a.s. \qquad \text{or} \qquad \sum_{t=1}^{n} x_t = \mathbf{o}(u_n) \quad a.s \tag{C.12}$$

on the set $\Omega_2 = \{\lim_{n\to\infty} u_n = \infty,\ \sum_{t=1}^{\infty} u_t^{-p} E|x_t|^p|\mathcal{F}_{t-1}) < \infty\}$ if $1 \leq p \leq 2$.

(3)

$$\sum_{t=1}^{\infty} u_t^{-1} x_t \ \text{ converges } \ a.s. \tag{C.13}$$

and

$$\lim_{n\to\infty} u_n^{-1} \sum_{t=1}^{n} x_t = 0 \ \ a.s. \tag{C.14}$$

on the set $\Omega_3 = \{\sum_{t=1}^{\infty} u_t^{-1} < \infty, \ \ \sum_{t=1}^{\infty} u_t^{-1-p/2} E|x_t|^p|\mathcal{F}_{t-1}) < \infty\}$ *if* $2 < p < \infty$. ∎

The statistical tools which are often used in martingale analysis are stated below.

Lemma C.5 [Wei, [149]] *Let* $\{\omega_t\}$ *be a martingale difference process with respect to an increasing sequence of* σ*-fields* $\{\mathcal{F}_t\}$ *such that* $\sup_t E\omega_t^2|\mathcal{F}_{t-1}) < \infty$ *a.s.. Let* u_t *be an* $\mathcal{F}_t$*-measurable random variable for every* t*. Then*

$$\sum_{t=1}^{n} u_t\omega_t \ \text{ converges } \ a.s. \tag{C.15}$$

on the set $\{\sum_{t=1}^{\infty} u_t^2 < \infty\}$*. And*

$$(\sum_{t=1}^{n} u_t\omega_t)/\{(\sum_{t=1}^{n} u_t^2)^{1/2}(\log(\sum_{t=1}^{n} u_t^2))^{\eta}\} \to 0 \ \ a.s. \tag{C.16}$$

on the set $\{\sum_{t=1}^{\infty} u_t^2 = \infty\}$ *for every* $\eta > 1/2$*, and consequently with probability 1*

$$\sum_{t=1}^{n} u_t\omega_t = \mathbf{o}(\sum_{t=1}^{n} u_t^2) + \mathbf{O}(1). \tag{C.17}$$

Moreover,

$$\sum_{t=1}^{n} |u_t|\omega_t^2 < \infty \ \ a.s. \tag{C.18}$$

on the set $\{\sum_{t=1}^{\infty} |u_t| < \infty\}$*, and*

$$(\sum_{t=1}^{n} |u_t|\omega_t^2)/(\sum_{t=1}^{n} |u_t|)^{\rho} \to 0 \ \ a.s. \tag{C.19}$$

on the set $\{\sum_{t=1}^{\infty} |u_t| = \infty\}$ *for every* $\rho > 1$*. If* $\sup_t E\omega_t^{\alpha}|\mathcal{F}_{t-1}) < \infty$ *a.s. for some* $\alpha > 2$*, then (C.19) can be strengthened into*

$$\lim_{n\to\infty} \sup_n (\sum_{t=1}^{n} |u_t|\omega_t^2)/(\sum_{t=1}^{n} |u_t|) < \infty \ \ a.s. \tag{C.20}$$

on the set $\{\sup_t |u_t| < \infty, \ \ \sum_{t=1}^{\infty} |u_t| = \infty\}$ ∎

Lemma C.6 [Wei, [149]] *Let $\{\omega_t, \mathcal{F}_t\}$ be a sequence of martingale differences such that* $\sup_t E|\omega_t|^\alpha|\mathcal{F}_{t-1}\} < \infty$ *a.s. for some $\alpha \geq 2$. Let u_t be $\mathcal{F}_{t-1}$-measurable random variable, $s_n^2 = \sum_{t=1}^n u_t^2$ and f be a nondecreasing function such that*

$$\int_1^\infty (xf^\alpha(x))^{-1}dx < \infty \tag{C.21}$$

Then on the set $\{s_n^2 \to \infty\}$,

$$S_n = \sum_{t=1}^n u_t\omega_t = \mathbf{o}(s_n f(s_n^2)) \quad a.s. \tag{C.22}$$

∎

Appendix D

The Proof of Theorem 6.1

It follows from the definition of $\hat{A}(z^{-1})$ in (6.1) that

$$\hat{A}(z^{-1}) = I + \hat{A}_1 z^{-1} + \cdots + \hat{A}_p z^{-p}.$$

So,

$$\det \hat{A}(z^{-1}) \neq 0 \quad |z| \geq 1 \quad \Longleftrightarrow \quad \det z^p \hat{A}(z^{-1}) \neq 0 \quad |z| \geq 1.$$

It is well known (see e.g. [82]) that $\det z^p \hat{A}(z^{-1})$ is the characteristic polynomial of the matrix

$$A_c = \begin{pmatrix} -\hat{A}_1^T & I & 0 & \cdots & 0 \\ -\hat{A}_2^T & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\hat{A}_{p-1}^T & 0 & 0 & \cdots & I \\ -\hat{A}_p^T & 0 & 0 & \dots & 0 \end{pmatrix},$$

where $\hat{A}_i \in \mathcal{R}^{m\times m}, i = 1, \cdots, p$. In fact, A_c is a companion matrix of the matrix polynomial $z^p \hat{A}(z^{-1})$. Hence, the polynomial $\det \hat{A}(z^{-1})$ has its all roots within the unit circle if and only if all the eigenvalues of A_c lie within the unit circle. Let λ and χ denote an arbitrary eigenvalue of A_c and its associated eigenvector, respectively, so that

$$A_c \chi = \lambda \chi \tag{D.1}$$

Partitioning the non-zero vector χ into $\chi^T \equiv (\chi_1^T \cdots \chi_p^T)$, $\chi_i \in \mathcal{R}^m$, one can rewrite Eq.(A1) in a more detailed form:

$$\begin{aligned}
\chi_2 - \hat{A}_1^T \chi_1 &= \lambda \chi_1 \\
&\vdots \\
\chi_p - \hat{A}_{p-1}^T \chi_1 &= \lambda \chi_{p-1} \\
-\hat{A}_p^T \chi_1 &= \lambda \chi_p
\end{aligned}$$

It is easy to see from the above equations that $\chi_1 \neq 0$; otherwise $\chi = 0$. Thus, an alternative and compact expression of Eq.(D.1) is

$$\begin{pmatrix} \chi \\ 0 \end{pmatrix} = \begin{pmatrix} I \\ \theta_A \end{pmatrix} \chi_1 + \lambda \begin{pmatrix} 0 \\ \chi \end{pmatrix}, \tag{D.2}$$

where $\theta_A^T \triangleq (\hat{A}_1 \cdots \hat{A}_p)$. Note that the Yule-Walker equation described in Eq.(6.4) can be written in a compact manner:

$$(\, I \quad \theta_A^T \quad \theta_C^T \,)\, \Gamma(p+1, q) = (\, R^f \quad 0 \cdots 0 \quad 0 \cdots 0 \,) \tag{D.3}$$

where $\theta_C^T \triangleq (\, -\hat{C}_0 \quad \cdots \quad -\hat{C}_q \,)$. Define two column vectors $\delta \in R^l$ and $\alpha = (\, \alpha_1^T \ \cdots \ \alpha_q^T \,)^T \in R^{ql}$ as the solution to the equation: $\begin{pmatrix} \alpha \\ 0 \end{pmatrix} = \theta_C \chi_1 + \lambda \begin{pmatrix} \delta \\ \alpha \end{pmatrix}$. It is easy to see that the equation always has a solution. Hence, it follows immediately from (D.2) that

$$\begin{pmatrix} \chi \\ 0 \\ \cdots \\ \alpha \\ 0 \end{pmatrix} = \begin{pmatrix} I \\ \theta_A \\ \cdots \\ \theta_C \end{pmatrix} \chi_1 + \lambda \begin{pmatrix} 0 \\ \chi \\ \cdots \\ \delta \\ \alpha \end{pmatrix}. \tag{D.4}$$

Using (D.3), (D.4), and (6.4) and exploiting the Toeplitz structure of matrix $\Gamma(p+1, q)$, one can have that

$$\begin{aligned}
(\chi^* \quad \alpha^* \quad 0^*)\Gamma(p,q) \begin{pmatrix} \chi \\ \alpha \\ 0 \end{pmatrix} &= (\chi^* \quad 0^* \quad \alpha^* \quad 0^*)\Gamma(p+1,q) \begin{pmatrix} \chi \\ 0 \\ \alpha \\ 0 \end{pmatrix} \\
&= \chi_1^* R^f(p,q) \chi_1 + |\lambda|^2 (\, 0^* \quad \chi^* \quad \delta^* \quad \alpha^* \,)\, \underline{R}^1(p,q) \begin{pmatrix} 0 \\ \chi \\ \delta \\ \alpha \end{pmatrix},
\end{aligned} \tag{D.5}$$

where $*$ represents the complex conjugate transpose.

Partition $\Gamma(p+1,q)$ into four Toeplitz submatrices so that

$$\Gamma(p+1,q) = \begin{pmatrix} T_{1,1}(p+1,q) & T_{1,2}(p+1,q) \\ T_{2,1}(p+1,q) & T_{2,2}(p+1,q) \end{pmatrix},$$

where

$$T_{1,1}(p+1,q) \equiv \begin{pmatrix} R_{yy}(0) & R_{yy}(1) & \cdots & R_{yy}(p) \\ R_{yy}(-1) & R_{yy}(0) & \cdots & R_{yy}(p-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_{yy}(-p+1) & R_{yy}(-p+2) & \cdots & R_{yy}(1) \\ R_{yy}(-p) & R_{yy}(-p+1) & \cdots & R_{yy}(0) \end{pmatrix}$$

$$T_{1,2}(p+1,q) \equiv \begin{pmatrix} R_{yu}(0) & R_{yu}(1) & \cdots & R_{yu}(q) \\ R_{yu}(-1) & R_{yu}(0) & \cdots & R_{yu}(q-1) \\ \vdots & \vdots & & \vdots \\ R_{yu}(1-p) & R_{yu}(2-p) & \cdots & R_{yu}(q-p+1) \\ R_{yu}(-p) & R_{yu}(1-p) & \cdots & R_{yu}(q-p) \end{pmatrix}$$

$$T_{2,1}(p+1,q) \equiv [T_{1,2}(p+1,q)]^T$$

$$T_{2,2}(p+1,q) \equiv \begin{pmatrix} R_{uu}(0) & R_{uu}(1) & \cdots & R_{uu}(q) \\ R_{uu}(-1) & R_{uu}(0) & \cdots & R_{uu}(q-1) \\ \vdots & \vdots & \ddots & \vdots \\ R_{uu}(-q+1) & R_{uu}(-q+2) & \cdots & R_{uu}(1) \\ R_{uu}(-q) & R_{uu}(-q+1) & \cdots & R_{uu}(0) \end{pmatrix}.$$

where $\{R_{yy}(k), R_{uu}(k), R_{yu}(k), R_{uy}(k), k = 0, 1, 2\cdots\}$ are a sequence of autocorrelation matrices and cross-correlation matrices of the output and input processes. Note that all four submatrices are block-Toeplitz matrices and $R_{uy}(k) = 0$ and $R_{uu}(k) = 0$ for any $k > 0$ because of the assumptions of Theorem 6.1. So, matrix $\Gamma(p+1,q)$ can also be partitioned into: $\Gamma(p+1,q) =$

$$\left(\begin{array}{c|cc} T_{1,1}(p+1,q-1) & T_{1,2}(p+1,q-1) & \vdots \; B^T(p,q) \\ \hline T_{2,1}(p+1,q-1) & T_{2,2}(p+1,q-1) & \vdots \; 0 \\ \cdots\cdots & \cdots\cdots & . \; \cdots\cdots \\ B(p,q) & 0 & \vdots \; R_{uu}(0) \end{array}\right)$$

$$
= \left(\begin{array}{cc|cc}
R_{yy}(0) \;\vdots & D(p) & R_{yu}(0) \;\vdots & E(q) \\
\cdots\cdots \;\cdot & \cdots\cdots\cdots & \cdots\cdots \;\cdot & \cdots\cdots\cdots \\
D^T(p) \;\vdots & T_{1,1}(p,q-1) & 0 \;\vdots & T_{1,2}(p,q-1) \\
\hline
R_{uy}(0) \;\vdots & 0 & R_{uu}(0) \;\vdots & 0 \\
\cdots\cdots \;\cdot & \cdots\cdots\cdots & \cdots\cdots \;\cdot & \cdots\cdots\cdots \\
E^T(q) \;\vdots & T_{2,1}(p,q-1) & 0 \;\vdots & T_{2,2}(p,q-1)
\end{array}\right) \tag{D.6}
$$

where

$$
\begin{aligned}
B(p,q) &= (\, R_{uy}(-q) \quad R_{uy}(-q+1) \quad \cdots \quad R_{uy}(-q+p)\,), \\
D(p) &= (\, R_{yy}(1) \quad \cdots \quad R_{yy}(p-1) \quad R_{yy}(p)\,), \\
E(q) &= (\, R_{yu}(1) \quad \cdots \quad R_{yu}(q-1) \quad R_{yu}(q)\,).
\end{aligned}
$$

So, the second term on the right side of Eq.(D.5) is equal to:

$$
\|\lambda^2\|\,(\, 0^* \quad \chi^* \quad \delta^* \quad \alpha^*\,)
\left(\begin{array}{cc|cc}
R_{yy}(0) \;\vdots & D(p) & R_{yu}(0) \;\vdots & E(q) \\
\cdots\cdots \;\cdot & \cdots\cdots\cdots & \cdots\cdots \;\cdot & \cdots\cdots\cdots \\
D^T(p) \;\vdots & T_{1,1}(p,q-1) & 0 \;\vdots & T_{1,2}(p,q-1) \\
\hline
R_{uy}(0) \;\vdots & 0 & R_{uu}(0) \;\vdots & 0 \\
\cdots\cdots \;\cdot & \cdots\cdots\cdots & \cdots\cdots \;\cdot & \cdots\cdots\cdots \\
E^T(q) \;\vdots & T_{2,1}(p,q-1) & 0 \;\vdots & T_{2,2}(p,q-1)
\end{array}\right)
\begin{pmatrix} 0 \\ \chi \\ \delta \\ \alpha \end{pmatrix}
$$

$$
= \|\lambda^2\|\,(\, \chi^* \quad \alpha^* \;\vdots\; \delta^*\,)
\left(\begin{array}{c|c}
 & 0 \\
\Gamma(p,q-1) & \cdots \\
 & 0 \\
\hline
0 \;\vdots\; 0 & R_{uu}(0)
\end{array}\right)
\begin{pmatrix} \chi \\ \alpha \\ \cdots \\ \delta \end{pmatrix}. \tag{D.7}
$$

and the left side of Eq.(D.5) is equal to:

$$
(\, \chi^* \quad \alpha^* \quad 0^*\,)
\left(\begin{array}{c|cc}
T_{1,1}(p,q-1) & T_{1,2}(p,q-1) \;\vdots & B^T(p,q) \\
\hline
T_{2,1}(p,q-1) & T_{2,2}(p,q-1) \;\vdots & 0 \\
\cdots\cdots & \cdots\cdots \;\cdot & \cdots\cdots \\
B(p,q) & 0 \;\vdots & R_{uu}(0)
\end{array}\right)
\begin{pmatrix} \chi \\ \alpha \\ 0 \end{pmatrix}
$$

$$
= (\, \chi^* \quad \alpha^*\,)\,\Gamma(p,q-1) \begin{pmatrix} \chi \\ \alpha \end{pmatrix}. \tag{D.8}
$$

Hence, substituting (D.7) and (D.8) into (D.5) generates

$$|\lambda|^2 = 1 - \frac{\chi_1^* R^f(p,q)\chi_1}{(\chi^* \quad \alpha^*)\Gamma(p,q-1)\begin{pmatrix}\chi \\ \alpha\end{pmatrix} + \delta^* R_{uu}(0)\delta}.$$

Note that the assumptions of Theorem 6.1 imply that $\Gamma(p,q-1) > 0$, $R^f(p,q) > 0$, and $R_{uu}(0) > 0$. Therefore, we have $|\lambda|^2 < 1$. ∎

The proof of Corollary 1: The arguments in [138] (Section C5.1) show that the matrix $\Gamma(p+1,q)$ is positive definite. Hence, the corollary immediately follows from Theorem 6.1. ∎