Spring 1998

System Identification and Control of Cavity Noise Reduction

Chien-Hsun Kuo

SYSTEM IDENTIFICATION AND CONTROL OF CAVITY NOISE REDUCTION

by

Chien-Hsun Kuo
M.E. Dec. 1994, Old Dominion University
B.S. June 1990, Feng-Chia University, Taiwan

A Dissertation submitted to the Faculty of Old Dominion University in Partial Fulfillment of the Requirement for the Degree of

DOCTOR OF PHILOSOPHY
ENGINEERING MECHANICS

Old Dominion University
May 1998

Approved by:

Jen-Kuang Huang (Director)

Chuh Mei (Member)

Sebastian Bawab (Member)

Keith M. Williamson (Member)

ABSTRACT

SYSTEM IDENTIFICATION AND CONTROL OF CAVITY NOISE REDUCTION

Chien-Hsun Kuo
Old Dominion University, 1998
Director: Dr. Jen-Kuang Huang

This dissertation first presents indirect closed-loop system identification through residual whitening, then identifies the cavity noise system and applies controllers to reduce the noise. High speed air flow over the cavity produces a complex oscillatory flow-field and induces pressure oscillations within the cavity. The existence of cavities induces large pressure fluctuations which generate undesirable and loud noise. This may have an adverse effect on the objects, such as reducing the stability and performance of overall system, or damaging the sensitive instruments within the cavity.

System identification is the process of building mathematical models of dynamical systems based on the available input and output data from the systems. The indirect system identification by residual whitening is used to improve the accuracy of the identification result, and the optimal properties of the Kalman filter could be enforced for a finite set of data through residual whitening. Linear Quadratic Gaussian (LQG) and deadbeat controllers are applied to obtain the desired system performance. Linear Quadratic Gaussian (LQG) control design is the technique of combining the linear quadratic regulator (LQR) and Kalman filter together, namely, state feedback (LQR) and state estimation (Kalman filter). Deadbeat control design is to bring the output to zero, and both indirect and direct algorithms are applied. For the indirect method, one needs to calculate the finite difference model coefficient parameters first, then form the control parameters. In the recursive direct algorithms, however, one can compute the control parameters directly. When systems change with time, the system parameters become time-varying. An adaptive predictive control is needed for this situation. Since the system parameters are time-varying, the control parameters need to be updated in order to catch up with the systems' changes. The classical recursive least-squares technique is used for the recursive deadbeat controller, and it could be operated for on-line application.

ACKNOWLEDGEMENTS

I am deeply indebted to my advisor, Dr. Jen-Kuang Huang for his thoughtful guidance, care and motivation. The members of my doctoral committee, Dr. Chuh Mei, Dr. Sebastian Bawab, and Dr. Keith M. Williamson are especially appreciated for all the help they gave me.

This dissertation would not be possible without my parents' love, understanding and support.

This work was supported partially by NASA Langley Research Center (NAS1-19858, No. 30). This support is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Unless otherwise stated the listed symbols are specified as follows.

A, B, C	open-loop system matrices
A_c, B_c, C_c	closed-loop system matrices
A_d, B_d, C_d, D_d	system matrices of dynamic controller
a_i, b_i or c_i, d_i	coefficient matrices of ARX model
H	Hankel matrix
h_i, t_i, s_i	coefficient matrices of ARMAX model
I	identity matrix
K	Kalman filter gain
l	data length
n	number of states
n_i	number of inputs
n_o	number of outputs
p	order of ARMAX model
q	order of ARX model
r	reference input
u	control input
V_{okid} or V_{clid}	input/output data matrix
v	measurement noise
w	process noise
x	system state
Y_d	controller Markov parameter
Y_k	open-loop Kalman filter Markov parameter
Y_{kc}	closed-loop Kalman filter Markov parameter
Y_s	open-loop system Markov parameter

Y_{sc}	closed-loop system Markov parameter
y	system output

Greek Letters

ε	residual after filtering
η	augmented state of open-loop system and controller
θ	augmented coefficient matrix of ARMAX model
θ_{okid} or θ_{clid}	augmented coefficient matrix of ARX model

Subscripts

k	k-th time step
i, j	discrete numbers

Superscripts

T	matrix transpose
-1	matrix inverse

Notation above a Symbol

^	estimate
.	time derivative

CHAPTER I

INTRODUCTION

1.1 Background and Problem Statement

High speed air flow inside a cavity produces a complex oscillatory flowfield and induces pressure oscillations within the cavity. Cavities exist in many aerodynamic configurations, such as the weapons' bay and wheel wells on aircraft, and also the recessed areas on some missile configurations. The existence of cavities induces large pressure fluctuations which generate undesirable and loud noise. This may have an adverse effect on the objects, such as influencing the stability and performance of the overall system, or damaging the sensitive instruments within the cavity. Hence, the noise reduction inside a cavity is an important issue to resolve for avoiding unnecessary damage and cost, and enhancing the system structure and achieving the desired system performance.

Cavities with length-to-depth ratio smaller than 10 are called open cavities. In these the shear layer attaches at or downstream of the rear wall. The closed cavities are those with length-to-depth ratio greater than 13 and the shear layer may attach on the floor of the cavity. In the work of this dissertation I will deal with open cavities. In open cavities, the shear layer, which forms between the high-speed external flow and low-speed recirculatory flow within the cavities, deflects into the cavity and partially blocks the pressure waves which travel back and fourth along the longitudinal direction inside the cavity[1]. A reverse flow is developed by the low-pressure region behind the front wall and the high-pressure region in front of the rear wall. The airflow separates from the front edge and has lower pressure than the free-stream because of the high speed external stream when it enters the cavity. As the airflow approaches the rear wall, the pressure will rise because the airflow will be retarded. Thus a captive eddy is generated within the cavity. For open cavities, the airflow pressure distribution is quite constant over most of the cavity length, except it will rise sharply when approaching the rear wall. In a previous study,

---

The journal mode adapted for this dissertation is American Institute of Aeronautics and Astronautics (AIAA).

Rossiter[2] concluded that the unsteady pressures operating in and around an open cavity in a subsonic and transonic airflow may contain both random and periodic components and, in general, the random component predominates in shallow cavity whose length-to-depth ratio is smaller than 4. Figure 1.1 shows the process of the feedback loop inside the cavity as follows:[3] 1) The pressure wave from a previous trailing-edge disturbance reaches the front wall and reflects. Another wave which already reflected off the front wall approaches the rear wall. At this time, the shear layer is deflected above the rear wall. 2) The shear layer oscillation travels downstream in a wave-like pattern and eventually hits the trailing edge. 3) The shear layer, which is now below the trailing edge at the rear wall, forms an upstream traveling compression wave. 4) The upstream and downstream compression waves meet and interact near the cavity center. 5) After the interaction, the waves continue in their respective directions. 6) The shear layer is now above the trailing edge and the oscillation cycle repeats.

The investigations for the airflow over a cavity have been well studied according to different kinds of length-to-depth ratio and airflow speed, since cavity flows emit strong acoustic tones which may have an adverse effect on the systems. Many experimental and computational studies of cavities with suppression devices have been performed by researchers. Sarohia and Massier[4] used a continuous mass addition at the base of the cavity to stabilize the shear layer and achieved a stabilizing effect on cavity shear flows. Sarno and Franke[5] studied the effects of manipulating the shear layer over the cavity leading edge by using static and oscillating fences and steady and pulstating flow injection at the leading edge. Kim and Chokani[2] implemented the computation of the supersonic turbulent flow over a two-dimensional rectangular cavity with passive venting. The passive venting control is performed by using a porous surface over a vent chamber in the cavity floor. Jeng and Payne[6] proposed a method based on the porous plate idea of Kim and Chokani. They replaced one of the cavity walls (forward and aft bulkhead, floor) with a porous wall and adjoining vent chamber to suppress the pressure fluctuations within the cavity. Baysal et al.[1] performed two dimensional, computational simulations for the transonic, turbulent flows past a cavity with two suppression devices. One with a rear face ramp and the other with a spoiler at the front lip.

System identification or modeling is the process of building mathematical models of dynamical systems based on the available input and output data from the systems. This technique is important in a diversity of fields such as communication, economics, statistics, system dynamics and control. For an unknown system, it is mostly required to identify the system model before one can perform the control design. There are many system identification techniques according to different kinds of need such as the nature of the system and the purpose of identification. Before performing the system identification, one needs to choose a suitable model structure first, and then the model parameters are chosen to minimize a defined cost function which indicates the fitness of the model to the input and output data. In reality when one identifies a system from the input and output data in time domain, the available data length is not long enough and the chosen model order is not sufficiently large, then the residual may not be minimized and white. It may be possible to get a more accurate model by whitening the residual. In the mean time, by some reformulation, one may choose a smaller model order to identify the system.[7]

The aim of studying control is how to determine appropriate control signals so that the system behavior could satisfy a required specification. Mathematically, the specific important signals are represented by a set of equations called the control law. In the Linear Quadratic Regulator (LQR) control design, one requires all of the states' information for performing state feedback. State feedback strategy is a common technique in controlling linear systems. However, in general, not all of the states can be measured directly. One needs to deduce some information from the measured output, which is corrupted by noise, by filtering it. In 1960 Kalman published his famous method for sequential state estimation of discrete systems, known as the Kalman filter, using state space formulation.[8] This paper is a landmark in modern control theory. Two years later, a version of Kalman filter for continuous systems was published.[9] With the Kalman filter, the state estimation can be reconstructed from the output. Thus one can implement the LQR control design through state feedback. The technique of combining state feedback and state estimation is called Linear Quadratic Gaussian (LQG) control design.

For systems with unknown disturbances and considerable uncertainties, one needs to design a controller which can catch up with systems' changes. Adaptive control techniques[10-17] become important in this issue. Since the system parameters are varying with time, a parameter estimation algorithm which has the capability of tracking the time variations is needed. The classical recursive least-squares technique is one of the approaches. When systems are time-invariant, the gain of the algorithm decreases to zero eventually since the system parameters are time constant. For time-varying systems, the gain will adjust itself according to systems variations. Hence, it can be used for real time application.

1.2 Objective

The objective of this dissertation is to identify the noise system by using the whitening residual method, and apply a suitable controller to reduce the noise.

First, system identification techniques are shown for both open-loop and closed-loop system. A residual whitening method is developed for indirect closed-loop system identification. Since the exact model order is usually unknown, there is residual left. By whitening the residual, one can extract more information from the residual to improve the system identification. In the mean time, one can also use a smaller order to identify the system. Through residual whitening, the residual is minimized and whitened. Hence, one can obtain a more accurate model by adjusting the system parameters.

Second, one can apply Linear Quadratic Gaussian (LQG) and adaptive control to obtain the desired system performance, i.e., to reduce the noise. Linear Quadratic Gaussian (LQG) control design is the technique of combining the linear quadratic regulator (LQR) and Kalman filter together, namely, state feedback (LQR) and state estimation (Kalman filter). Using the Kalman filter technique, one can obtain the optimal estimate of states in stochastic systems. Then the estimated states are used for state feedback design which minimizes a performance index. One can identify a steady-state Kalman filter gain from the available input and output data. By choosing suitable weighting matrices through state feedback design, one can obtain the desired system performance.

When systems change with time, the system parameters become time-varying. An adaptive predictive control is needed for this situation. Since the system parameters are time-varying, the control parameters need to be updated in order to catch up with systems changes. A recursive deadbeat controller for systems without direct transmission input term is presented for on-line operation. The recursive deadbeat control design will bring the system output to deadbeat (zero). During the processes, the gain matrices are computed in every sampling interval, so the control parameters are updated for real time application. Here, a classical recursive least-squares method is used for on-line calculation.

1.3 Dissertation Outline

Chapter II introduces the existing open-loop and closed-loop system identification techniques. One can build mathematical models of dynamical systems by analyzing a sufficient number of input and output data through the system identification techniques. The closed-loop system identification is needed for marginally stable or unstable systems which are required to have a feedback control to make the overall system stable. One needs to choose a suitable model according to different needs before identifying the system. Some basic model structures and the least-squares technique for both batch and recursive formulation which will be used for a later chapter are shown.

Chapter III proposes the indirect closed-loop system identification through residual whitening. Normally, when one identifies a system from input-output data in a time domain, it is assumed that the data length is long enough and the autoregressive with exogeneous input (ARX) model order is sufficiently large. In the residual whitening method, one uses the autoregressive moving average with exogeneous input (ARMAX) model which includes the dynamics of noise instead of ARX model to minimize and whiten the residual. The properties of the residual sequence, i.e., the orthogonal conditions, will convert to the optimal properties of the Kalman filter. One can also relax the requirement of the model order to reduce the computation burden, especially for several input and output systems. A simulation of 5 degrees of freedom of the large angle magnetic suspension test facility (LAMSTF) system which is unstable is also provided.

Chapter IV provides the material for Linear Quadratic Gaussian (LQG) control design and Kalman filters. The LQG control design includes continuous-time and discrete-time approaches. The LQG control separates design problems into two stages, namely, state feedback (LQR) and state estimation (Kalman filter). One can use a Kalman filter to estimate the optimal state from output data, then perform the state feedback (LQR) technique to obtain desired system performance. A filter-innovation model is also derived for the steady-state Kalman filter gain.

Chapter V shows the deadbeat controller design without the direct transmission term. First, a multi-step ahead prediction output is derived. After obtaining the ARX coefficient parameters, one can integrate the prediction output equations and solve for the deadbeat control force. This is off-line calculation. Then, a recursive least-squares solution for deadbeat control design is presented. One can perform the adaptive control for on-line operation and the formulation satisfied simultaneously system identification and deadbeat controller design requirements, i.e., the system identification and deadbeat control are built into one formulation.

Chapter VI verifies the control design algorithm by numerical simulation. The simulation uses experimental single input and single output data (SISO), both LQG and adaptive control design are performed. And both control design methods are applicable to multi-input and multi-output cases.

Finally, chapter VII provides conclusions and prospects for the extension of this research.

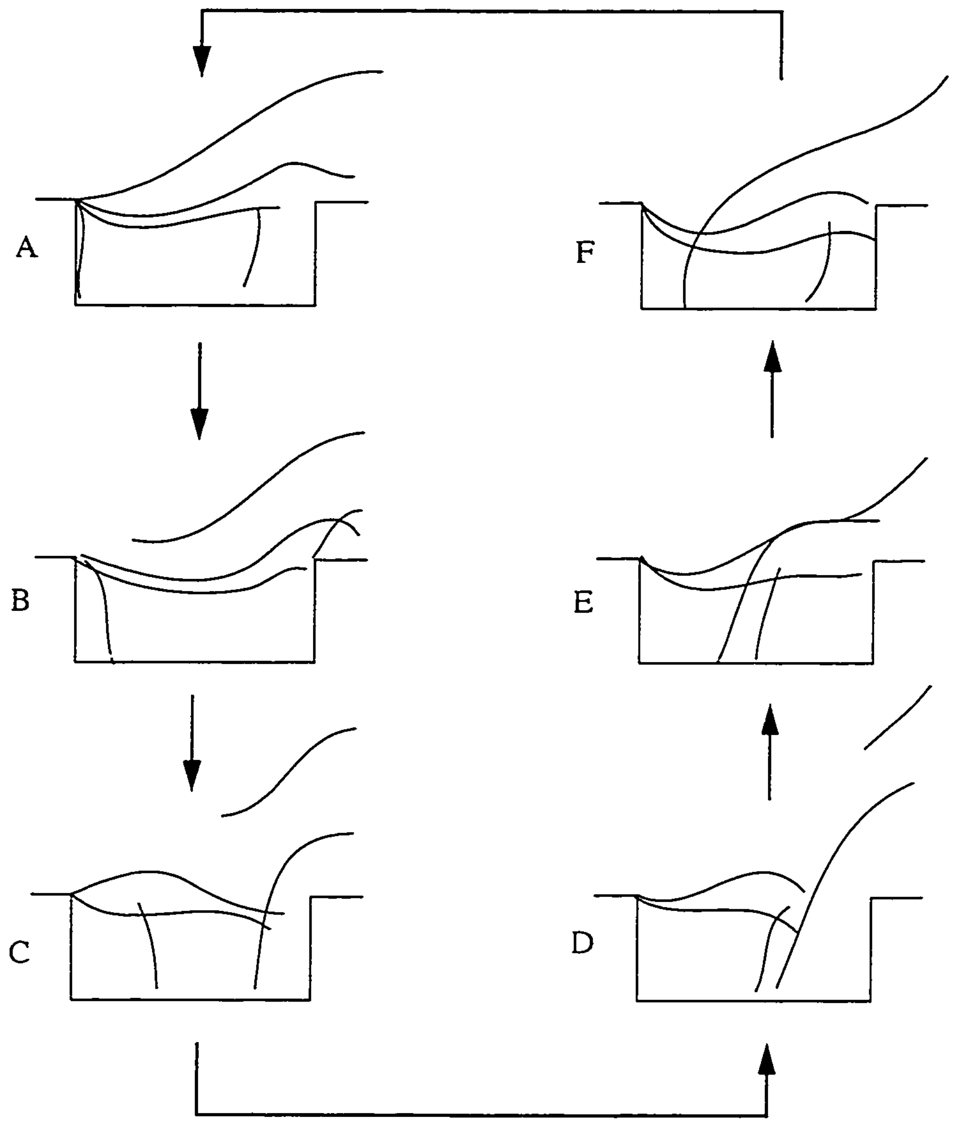


Figure 1.1 Typical Oscillation Cycle

CHAPTER II

DIRECT AND INDIRECT SYSTEM IDENTIFICATION ALGORITHM

2.1 Introduction

In general, one can distinguish between different identification situations for different treatments as follows: First, one distinguishes between linear and nonlinear systems. The linear one is easier to be identified than the nonlinear one because of the linear properties of superposition. Second, one distinguishes between time-invariant and time-varying systems. The latter being systems whose parameters vary with time. Systems may be considered as time-invariant if their parameters vary slowly in comparison with the time needed for adequate identification. Third is a classification of deterministic and stochastic system. In realism all systems contain noise in process and measurement. The knowledge of deterministic systems is helpful to understand the process of stochastic systems. Fourth is a classification of discrete and continuous systems and it is usually straightforward to transform the formulation between continuous and discrete systems.

Basically, we can classify two methods for system identification, the nonparameteric method and the parametric method.[18-20] The earliest methods of control system identification are those based on frequency, step and impulse responses which are called nonparameteric system identification methods since they do not employ a finite-dimensional parameter vector in the search for a best description. These are developed in classical control theory. In modern control, we use the parameter method to attempt to fit a model which best represents the data from the observation of input and output sequences of our system. Therefore, a family set of candidate models is chosen first and then the particular member of this family set which is the most suitable for describing the input and output data is determined. A model of a system is a description of its properties, suitable for a certain purpose. In section 2.2, some different model structures are introduced. Least squares estimation[21-23] is a classical technique. It has been used widely, especially in curve fitting problems where it is desired to determine a function (for example, polynomials, exponential functions, sine and cosine functions) that best fits a set of data points in the sense of minimizing the sum squares differences between the measured data and the estimated, or proposed function. Recursive techniques could reduce storage requirements when estimating the parameters by the least-squares method from updated input and output data, and increase computational efficiency. The batch and classical recursive least squares estimation methods are introduced in section 2.3. In section 2.4, the observer/Kalman filter identification[24-27] (OKID) for stable systems without requiring feedback control is introduced. For identifying marginally stable or unstable systems, however, feedback control is required to ensure the overall system stability. There are generally three classifications to identify systems operating in closed-loop.[28-30] One way is called direct identification which treats the bounded plant input-output data exactly as if they were obtained from an open-loop experiment. Another way is to treat the closed-loop system as a whole. First you have to identify the closed-loop system dynamics, then determine the open-loop system dynamics with the known controller dynamics from the identified closed-loop system dynamics. This is called indirect identification.[31] The other approach is called jointly input-output identification.[32] In this approach, the feedback controller is unknown and the input and output are considered as a joint process and the output of the system is driven by noise only. The indirect method is given in section 2.5. For any dynamics system, although its system Markov parameter is unique, the realized state-space model is in a different coordinate. In order to compare the identified state-space model with the analytical model, both models need to be in the same coordinate. In section 2.6, a unique coordinate transformation matrix is derived to convert the realized state-space model, such that both models, the identified and the analytical models, can be compared in the same coordinate.

2.2 Types of Model Structures

System identification deals with the problem of building the mathematical models of dynamical systems according to the input and output data. The criterion of model quality is normally based on how well the models could perform when they attempt to 'fit' the measured data. A prior knowledge regarding the system would be very helpful in choosing model structure, although some information may be learned from analyzing measured data. In the following, four different types of model structures will be discussed, some of them will be used later on in this work.

2.2.1 AR Model

AR means Auto-Regressive. The output is an autoregressive of itself. The model can be described as follows:

$$y_k = \sum_{i=1}^{q} a_i y_{k-i} + e_k \tag{2.1}$$

where a_i are the AR model parameters and e_k is a random white noise.

2.2.2 ARX Model

The ARX model where AR refers to AutoRegressive part and X refers to the eXogeneous part is commonly used in developing recursive system identification technique. The model can be given as follows:

$$y_k = \sum_{i=1}^{q} a_i y_{k-i} + \sum_{i=1}^{q} b_i u_{k-i} + e_k \tag{2.2}$$

where y_{k-i} are the autoregressive part, u_{k-i} are the exogeneous part, a_i and b_i are the ARX parameter coefficients. The ARX model will be used in open-loop and closed-loop system identification later on.

2.2.3 MA Model

The MA model represents the Moving-Average term. It can be given as follows:

$$y_k = \sum_{i=1}^{q} a_i e_{k-i} + e_k \tag{2.3}$$

where a_i are the moving average parameters and e_{k-i} are white, gaussian noise terms.

2.2.4 ARMAX Model

The ARMAX represents the AutoRegressive Moving Average with eXogeneous input model. It is a combination of ARX and MA models, and will be used later on for whitening residual. The ARMAX model is given as follows:

$$y_k = \sum_{i=1}^{q} a_i y_{k-i} + \sum_{i=1}^{q} b_i u_{k-i} + \sum_{i=1}^{q} c_i e_{k-i} + e_k \tag{2.4}$$

where a_i, b_i, c_i are the ARMAX parameters.

The basic difference between ARX and ARMAX models is that the ARMAX model contains the moving average terms, i.e. noise dynamics.

2.3 Least Squares Method

The method of least squares was first used by Gauss and he applied this approach practically for astronomical computations at the end of eighteen century. He defined "the most probable value of the unknown quantities will be one for which the sum of squares of the differences between the actually observed and computed values multiplied by numbers that measure the degree of precision is a minimum." In a great many different fields including the system identification applications, the least squares method reached a significant achievement and was modified according to different requirements.

2.3.1 Batch Solution of the Least Squares

Let y_k and v_k be the observation and measurement noise at time k, where $k = 1, 2, \ldots, l$, y_k and v_k both are $m \times 1$ vector. Assume that θ is a constant q dimensional vector of parameters to be estimated with the observations linear in θ

$$Y_l = H_l \theta + V \tag{2.5}$$

where $Y_l^T = \begin{bmatrix} y_1 & y_2 & \ldots & y_l \end{bmatrix}$, $H_l^T = \begin{bmatrix} h_1 & h_2 & \ldots & h_l \end{bmatrix}$, $V^T = \begin{bmatrix} v_1 & v_2 & \ldots & v_l \end{bmatrix}$, and Y_l, V, and H_l are $ml \times 1$, $ml \times 1$ vector and $ml \times q$ matrix, respectively.

It is desired to obtain the estimate of θ that minimizes the sum squares of $Y_l - H_l \theta$. Denote the estimate of θ based on the k data samples as $\tilde{\theta}_k$ and introduce the error associated with the $k-th$ measurement as $e_k = y_k - h_k \tilde{\theta}_k$, $k = 1, 2, \ldots l$. This error is normally called the measurement error or the residual. The performance criterion becomes

$$J = (Y_l - H_l\theta)^T R^{-1} (Y_l - H_l\theta) \tag{2.6}$$

where R is a weighting matrix, and J is called a cost function, risk function, etc. For R equals to identity matrix I, the method of minimizing Equation (2.6) is called procedure of least squares(LS). R of general positive definite form is usually called the denotation of weighted least squares. R equals the covariance matrix of noise, the procedure becomes a minimum variance estimation.

The solution of minimizing the quadratic cost function is to differentiate it with respect to the parameters and equate the result to zero. Thus for $R = I$

$$\frac{\partial J}{\partial \theta} = -2H_l^T (Y_l - H_l\theta) = 0$$

and the least squares estimate of θ is

$$\tilde{\theta} = (H_l^T H_l)^{-1} H^T Y_l \tag{2.7}$$

Equation (2.7) is the batch solution of the least squares method. It is noted that the quadratic cost function Equation (2.6) attains an absolute minimum if and only if $\theta = \tilde{\theta}$, where $\tilde{\theta}$ is obtained from Equation (2.7), and the least squares estimate of the matrix of parameter θ is unbiased if the mean values of the components of noise vector V are zeros and if the matrix H_l and V are mutually independent.

2.3.2 Classical Recursive Least Squares

Recursive identification algorithms are of great interest in control and estimation problems, and related areas such as recursive least squares and adaptive methods. Recursive techniques could reduce storage requirements when estimating the parameters by the least squares method from updated input and output data, and increase computational efficiency. It can also be used for on-line system identification in real time if one's data processing device can keep pace with the rate of the data acquisition. A requirement that is often satisfied because of the capabilities of contemporary microprocessors.

Define $Y_{k+1} = \begin{bmatrix} Y_k \\ y_{k+1} \end{bmatrix}$, $H_{k+1} = \begin{bmatrix} H_k \\ h_{k+1} \end{bmatrix}$, where $Y_k^T = \begin{bmatrix} y_1 & y_2 & \dots & y_k \end{bmatrix}$, and

$H_k^T = \begin{bmatrix} h_1 & h_2 & \dots & h_k \end{bmatrix}$ Denote the estimate of θ based on the k observations as $\tilde{\theta}_k$, the least squares estimate of parameter is according to Equation (2.7)

$$\tilde{\theta} = (H_{k+1}^T H_{k+1})^{-1} H_{k+1}^T Y_{k+1}$$

$$= [H_k^T H_k + h_{k+1}^T h_{k+1}]^{-1} [H_k^T Y_k + h_{k+1}^T y_{k+1}]$$

After using matrix inversion lemma, one can get

$$\tilde{\theta}_{k+1} = \tilde{\theta}_k + M_{k+1}(y_{k+1} - h_{k+1}\tilde{\theta}_k) \tag{2.8}$$

where $M_{k+1} = (H_k^T H_k)^{-1} h_{k+1}^T [1 + h_{k+1}(H_k^T H_k)^{-1} h_{k+1}^T]^{-1}$.

The term $h_{k+1}\tilde{\theta}_k$ may be considered as the prediction of y_{k+1} based on the estimate of parameters $\tilde{\theta}_k$ and on the set of measurements h_{k+1}. So, Equation (2.8) could be rewritten as

$$\tilde{\theta}_{k+1} = \tilde{\theta}_k + M_{k+1}(y_{k+1} - \tilde{y}_{k+1}) \tag{2.9}$$

One can see that the estimate parameters $\tilde{\theta}_{k+1}$ is equal to the previous estimate $\tilde{\theta}_k$ corrected by the weighting coefficients M_{k+1} multiplies the error between the observation y_{k+1} and the predicted value $\tilde{y}_{k+1}$. The predicted value $\tilde{y}_{k+1} = h_{k+1}\tilde{\theta}_k$ is equal to the measured data y_{k+1} only if the exact system model with parameters $\tilde{\theta}_k = \theta_{k+1}$ is available and if the noise is absent. In such a case, the correction is zero. To calculate M_{k+1} in a recursive way, one define $P_k = (H_k^T H_k)^{-1}$. After some arrangement, the recursive least squares algorithm could be formulated as follows:

$$M_{k+1} = P_k h_{k+1}^T [1 + h_{k+1} P_k h_{k+1}^T]^{-1} \tag{2.10}$$

$$P_{k+1} = [I - M_{k+1} h_{k+1}^T] P_k \tag{2.11}$$

$$\tilde{\theta}_{k+1} = \tilde{\theta}_k + M_{k+1}(y_{k+1} - \tilde{y}_{k+1}) \tag{2.12}$$

2.4 Open-Loop System Identification Algorithm

In open-loop system identification, we excite the system directly without adding a controller, and identify the system model from the input and output data. Chen et al.[33] introduced a method to identify a state space model from a finite difference model. The difference model, called autoregressive with exogeneous input (ARX), is derived from Kalman filter theories. However, the method required an ARX model of large order which causes intensive computation. In Ref.34 a system is identified through a state observer, which can use a much smaller ARX model order than that derived through Kalman filter, but the derivation is based on deterministic approach. In Ref.35 the projection filters which were originally derived for deterministic systems are developed for the identification of linear open-loop stochastic systems. This approach has proved that least-squares identification of a finite difference model converges to the model derived from the projection filters, which the recursive projection filter of order one is identical to the Kalman filter. The open-loop observer/Kalman filter identification (OKID) is introduced in the following.

2.4.1 Algorithm for Open-Loop System

A finite-dimensional, linear, discrete-time, time invariant stochastic system can be expressed as:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{2.13}$$

$$y_k = Cx_k + v_k \tag{2.14}$$

where is $x_k \in R^{n \times 1}$ the state vector, $u \in R^{ni \times 1}$ is the signal input vector, and $y \in R^{no \times 1}$ is the output vector; $[A, B, C]$ are the system matrices. The sequence of process noise $w \in R^{n \times 1}$ and measurement noise $v \in R^{no \times 1}$ are assumed white, Gaussian, zero mean. The noises w_k and v_k are also assumed uncorrelated with covariance Q and R, respectively.

Equations (2.13), (2.14) can also be expressed as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + AK\varepsilon_k \tag{2.15 a}$$

$$y_k = C\hat{x}_k + \varepsilon_k \tag{2.15 b}$$

where $\varepsilon_k \in R^{no \times 1}$ is the output residual and equals $\varepsilon_k = y_k - \hat{y}_k$; it's zero mean, white, Gaussian. $\hat{x}_k$ and $\hat{y}_k$ are the estimated state and output, respectively. $K \in R^{n \times no}$ is the steady-state Kalman filter gain which satisfies $K = PC^T[R + CPC^T]^{-1}$. $P \in R^{n \times n}$ is the solution of the steady-state algebraic Riccati equation. The existence of K is guaranteed if the system is detectable and $\left(A, Q^{\frac{1}{2}}\right)$ is stabilizable.[10]

$$P = APA^T - APC^T[R + CPC^T]^{-1}CPA^T + Q. \tag{2.16}$$

The system in Equation (2.15) can also be expressed as

$$\hat{x}_{k+1} = (A - AKC)\hat{x}_k + Bu_k + AKy_k \tag{2.17 a}$$

$$y_k = C\hat{x}_k + \varepsilon_k \tag{2.17 b}$$

Then the relation between signal input and output with zero initial condition through Equation (2.17) could be described as

$$y_k = \sum_{i=1}^{\infty} C\bar{A}^{i-1}AKy_{k-i} + \sum_{i=1}^{\infty} C\bar{A}^{i-1}Bu_{k-i} + \varepsilon_k \tag{2.18}$$

where $\bar{A} = A - AKC$. Since $\bar{A}$ is asymptotically stable, $\bar{A}^{i-1} \approx 0$ if $i \geq q$ for a sufficient large number q.[27] Thus Equation (2.18) becomes

$$y_k = \sum_{i=1}^{q} a_i y_{k-i} + \sum_{i=1}^{q} b_i u_{k-i} + \varepsilon_k \tag{2.19}$$

where

$$a_i = C\bar{A}^{i-1}AK,\ b_i = C\bar{A}^{i-1}B,\ i = 1, 2, \ldots, q \tag{2.20}$$

The model described by Equation (2.19) is the open-loop ARX model which directly represents the relationship between the signal input and output of the open-loop system. q is the open-loop system ARX model order, and a_i, b_i, the open-loop ARX model parameters, can be estimated through the least squares method. Suppose that N data points of y_k and u_k, $k = 0, 1, \ldots, N-1$, are given. The batch least squares solution for estimating the parameters a_i, b_i is

$$\theta_{okid} = YV_{okid}^T (V_{okid} V_{okid}^T)^{-1} \tag{2.21}$$

where

$$Y = \begin{bmatrix} y_0 & y_1 & \dots & y_q & \dots & y_{N-1} \end{bmatrix}, \theta_{okid} = \begin{bmatrix} b_1 & a_1 & \dots & b_q & a_q \end{bmatrix}.$$

$$V_{okid} = \begin{bmatrix} 0 & u_0 & \dots & u_{q-1} & \dots & u_{N-2} \\ 0 & y_0 & \dots & y_{q-1} & \dots & y_{N-2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & u_0 & \dots & u_{N-q-1} \\ 0 & 0 & 0 & y_0 & \dots & y_{N-q-1} \end{bmatrix}$$

First, we need to obtain the open-loop system and Kalman filter Markov parameters, since the Markov parameters are uniquely determined for each system. Then we can realize system matrices (A, B, C) from the open-loop system Markov parameters and Kalman filter gain K from the open-loop Kalman filter Markov parameters. The open-loop system Markov parameters $Y_s(k) = CA^{k-1}B$, and the open-loop Kalman filter Markov parameters $Y_k(k) = CA^{k-1}AK$, $k = 1, 2, \dots, q$ can be obtained from the coefficients, a_i, b_i

$$Y_s(k) = b_k + \sum_{i=1}^{k} a_i Y_s(k-i) \tag{2.22}$$

$$Y_k(k) = a_k + \sum_{i=1}^{k-1} a_i Y_k(k-i) \tag{2.23}$$

where $Y_s(0) = 0$, $Y_k(0) = I$ which is an identity matrix.

2.4.2 Recover System Matrices and Kalman Filter Gain

To recover system matrices, an eigensystem realization method is applied. Then the recovered system matrices and the open-loop Kalman filter Markov parameters could be used for identifying Kalman filter gain. A realization is to obtain a set of (A, B, C)

through the computation of the open-loop system Markov parameters $Y_s(k)$, for which the discrete-time model, Equations (2.13) and (2.14), is satisfied. For the first step, form a Hankel matrix $H(k-1)$ from the open-loop system Markov parameters,

$$H(k-1) = \begin{bmatrix} Y_s(k) & Y_s(k+1) & \dots & Y_s(k+\beta-1) \\ Y_s(k+1) & Y_s(k+2) & \dots & Y_s(k+\beta) \\ \dots & \dots & \dots & \dots \\ Y_s(k+\alpha-1) & Y_s(k+\alpha) & \dots & Y_s(k+\alpha+\beta-2) \end{bmatrix}$$

where $Y_s(k)$ is the $k-th$ open-loop Markov parameter.

Using singular value decomposition (SVD) of $H(0)$

$$H(0) = U\Sigma V^T \tag{2.24}$$

A $n-th$ order discrete state space model can be identified as

$$A = \Sigma_n^{-\frac{1}{2}} U_n^T H(1) V_n \Sigma_n^{-\frac{1}{2}} \tag{2.25}$$

$$B = \Sigma_n^{\frac{1}{2}} V_n^T E_{ni} \tag{2.26}$$

$$C = E_{no}^T U_n \Sigma_n^{\frac{1}{2}} \tag{2.27}$$

where Σ_n is the upper left hand $n \times n$ partition of Σ containing the n largest singular values which are in the monotonically non-increasing order along the diagonal. U_n and V_n are the matrices formed by the first n columns of singular vectors associated with the n singular values from U and V, respectively. $E_{ni}^T = \begin{bmatrix} I_{ni} & 0_{ni} & \dots & 0_{ni} \end{bmatrix}$, $E_{no}^T = \begin{bmatrix} I_{no} & 0_{no} & \dots & 0_{no} \end{bmatrix}$, and ni the number of inputs, no the number of outputs.

Once the open-loop A and C are obtained, one can calculate the open-loop Kalman filter gain from the open-loop Kalman filter Markov parameters and A, C as follows:

$$K = (O^T O)^{-1} O^T \begin{bmatrix} Y_k(1) \\ Y_k(2) \\ \dots \\ Y_k(k) \end{bmatrix} \tag{2.28}$$

where $O = \begin{bmatrix} CA \\ CA^2 \\ \dots \\ CA^k \end{bmatrix}$

The identified Kalman filter gain can be used for state estimation.

2.4.3 Computation Steps for OKID

1. Collect input u and output y data from the experiment.
2. Estimate the ARX model parameters a_i, b_i from Equations (2.19) to (2.21) by choosing suitable ARX model order q.
3. Obtain the open-loop system Markov parameters $Y_s(k)$ and Kalman filter Markov parameters $Y_k(k)$ by Equations (2.22) and (2.23).
4. Recover the system matrices by using eigensystem realization method, Equations (2.25-2.27) and the Kalman filter gain by Equation (2.28).

2.5 Closed-Loop System Identification Algorithm

Closed-loop system identification (CLID) collects reference input and output data from a closed-loop system, which is different from an open-loop system identification. There are several instances when we need to use closed-loop system identification. The system may be operating in a closed-loop and only closed-loop data are measurable for identification. For example, marginally or unstable systems need to have a feedback control to ensure overall systems' stability. For such systems, it is not desirable to identify the open-loop system model without adding the controller. The following introduces the indirect method for identifying the open-loop system model from closed-loop data with known feedback dynamics.[31] In the indirect CLID method, we calculate the closed-loop ARX model coefficient matrices in the beginning, from which one can find the closed-loop system Markov parameters. By using closed-loop system Markov parameters with known controller Markov parameters, the open-loop system Markov parameters are identified. Then, one can realize open-loop system matrices by applying the eigensystem realization method, and recover the Kalman filter gain from the identified matrices and open-loop Kalman filter Markov parameters.

2.5.1 Algorithm for Closed-loop System

A finite-dimensional, linear, discrete-time, time invariant stochastic system can be expressed as:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{2.29}$$

$$y_k = Cx_k + v_k \tag{2.30}$$

where $x_k \in R^{n \times 1}$ is the state vector, $u \in R^{ni \times 1}$ is the signal input vector, and $y \in R^{no \times 1}$ is the output vector; $[A, B, C]$ are the system matrices. The sequence of process noise $w \in R^{n \times 1}$ and measurement noise $v \in R^{no \times 1}$ are assumed white, Gaussian, zero mean. The noises w_k and v_k are also assumed uncorrelated with covariance Q and R, respectively.

Equations (2.29), (2.30) can also be expressed as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + AK\varepsilon_k \tag{2.31 a}$$

$$y_k = C\hat{x}_k + \varepsilon_k \tag{2.31 b}$$

where $\varepsilon_k \in R^{no \times 1}$ is the output residual and equals $\varepsilon_k = y_k - \hat{y}_k$; it's zero mean, white, Gaussian. $\hat{x}_k$ and $\hat{y}_k$ are the estimated state and output, respectively. $K \in R^{n \times no}$ is the steady-state Kalman filter gain which satisfies $K = PC^T[R + CPC^T]^{-1}$. $P \in R^{n \times n}$ is the solution of the steady-state algebraic Riccati equation. The existence of K is guaranteed if the system is detectable and $\left(A, Q^{\frac{1}{2}}\right)$ is stabilizable.

Here, a dynamic feedback controller is added to ensure the system's stability. A dynamic feedback controller can be expressed as:

$$p_{k+1} = A_d p_k + B_d y_k \tag{2.32}$$

$$u_k = C_d p_k + D_d y_k + r_k \tag{2.33}$$

where p_k, r_k, $\begin{bmatrix} A_d & B_d & C_d & D_d \end{bmatrix}$ are the controller state, reference input to the closed-loop system and state-space matrices of the controller, respectively.

Taking Equations (2.31), (2.32), (2.33) together, the augmented closed-loop system dynamics becomes

$$\eta_{k+1} = A_c \eta_k + B_c r_k + A_c K_c \varepsilon_k \tag{2.34}$$

$$y_k = C_c \eta_k + \varepsilon_k \tag{2.35}$$

where

$$\eta_k = \begin{bmatrix} \hat{x}_k \\ p_k \end{bmatrix}, \quad \mathrm{A}_c = \begin{bmatrix} A + BD_d C & BC_d \\ B_d C & A_d \end{bmatrix}, \quad \mathrm{B}_c = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$A_c K_c = \begin{bmatrix} AK + BD_d \\ B_d \end{bmatrix}, \quad \text{and } \mathrm{C}_c = \begin{bmatrix} C & 0 \end{bmatrix}$$

It is noted that K_c can be considered as the Kalman filter gain for the closed-loop system and is guaranteed exist when the closed-loop system A_c is nonsingular. Substituting (2.35) into (2.34) yields

$$\eta_{k+1} = \bar{A}_c \eta_k + B_c r_k + A_c K_c y_k \tag{2.36}$$

where $\bar{A}_c = A_c - A_c K_c C_c$ is guaranteed to be asymptotically stable since the steady-state Kalman filter gain K_c exists. The z-transform of (2.36) and (2.35) yields

$$\eta(z) = (z - \bar{A}_c)^{-1}(A_c K_c y(z) + B_c r(z)) \tag{2.37}$$

$$y(z) = C_c \eta(z) + \varepsilon(z) \tag{2.38}$$

Substituting (2.37) into (2.38) yields

$$y(z) = C_c (z - \bar{A}_c)^{-1}(A_c K_c y(z) + B_c r(z)) + \varepsilon(z) \tag{2.39}$$

Taking the inverse z-transform of Equation (2.39) with $(z-\bar{A}_c)^{-1} = \sum_{i=1}^{\infty} \bar{A}_c^{i-1} z^{-i}$, one could get the relation between reference input and output with zero initial condition as

$$y_k = \sum_{i=1}^{\infty} C_c \bar{A}_c^{i-1} A_c K_c y_{k-i} + \sum_{i=1}^{\infty} C_c \bar{A}_c^{i-1} B_c r_{k-i} + \varepsilon_k \tag{2.40}$$

Since $\bar{A}_c$ is asymptotically stable, $\bar{A}_c^{i-1} \approx 0$ if $i \ge q$ for a sufficient large number q. Thus Equation (2.40) becomes

$$y_k = \sum_{i=1}^{q} c_i y_{k-i} + \sum_{i=1}^{q} d_i r_{k-i} + \varepsilon_k \tag{2.41}$$

where

$$c_i = C_c \bar{A}_c^{i-1} A_c K_c,\ d_i = C_c \bar{A}_c^{i-1} B_c,\ i = 1, 2, \ldots, q \tag{2.42}$$

Equation (2.41) is the closed-loop ARX model which directly represents the relationship between the reference input and output of the closed-loop system. q is the closed-loop system ARX model order, and c_i, d_i are the closed-loop ARX model parameters and they are used to find the closed-loop system and Kalman filter Markov parameters. Suppose that there are N data points of y_k and r_k, $k = 0, 1, \ldots, N-1$, are given. The batch least squares solution for estimating the parameters c_i, d_i is,

$$\theta_{clid} = Y V_{clid}^T (V_{clid} V_{clid}^T)^{-1} \tag{2.43}$$

where

$$Y = \begin{bmatrix} y_0 & y_1 & \ldots & y_q & \ldots & y_{N-1} \end{bmatrix},\ \theta_{clid} = \begin{bmatrix} d_1 & c_1 & \ldots & d_q & c_q \end{bmatrix}$$

$$V_{clid} = \begin{bmatrix} 0 & r_0 & \ldots & r_{q-1} & \ldots & r_{N-2} \\ 0 & y_0 & \ldots & y_{q-1} & \ldots & y_{N-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & r_0 & \ldots & r_{N-q-1} \\ 0 & 0 & 0 & y_0 & \ldots & y_{N-q-1} \end{bmatrix}$$

Note that V_{clid} is slightly different from V_{okid} of Equation (2.21), V_{clid} uses reference input r instead of signal input u, and the coefficient matrices a_i, b_i, c_i, d_i represent different parameters. The z-transforms of Equations (2.32) to (2.35) can be used to derive u(z) and y(z)

$$u(z) = \sum_{k=0}^{\infty} Y_d(k) z^{-k} y(z) + r(z) \tag{2.44}$$

$$y(z) = \sum_{k=1}^{\infty} Y_{sc}(k) z^{-k} r(z) + \sum_{k=0}^{\infty} Y_{kc}(k) z^{-k} \varepsilon(z) \tag{2.45}$$

where $Y_d(k) = C_d A_d^{k-1} B_d$ is the controller Markov parameters, $Y_{sc}(k) = C_c A_c^{k-1} B_c$, $Y_{kc}(k) = C_c A_c^{k-1} A_c K_c$ are the closed-loop system and Kalman filter Markov parameters, respectively. It is noted that $Y_d(0) = D_d$, and $Y_{kc}(0) = I$.

Next, the closed-loop system and Kalman filter Markov parameters can be recursively calculated from the estimated coefficient matrices, c_i, d_i of the closed-loop ARX model

$$Y_{sc}(k) = d_k + \sum_{i=1}^{k} c_i Y_{sc}(k-i) \tag{2.46}$$

$$Y_{kc}(k) = \sum_{i=1}^{k} c_i Y_{kc}(k-i) \tag{2.47}$$

It is also noted that $Y_{sc}(0) = 0$, and $Y_{kc}(0) = I$ and $c_i = d_i = 0$ when $i > q$.

Then, by using the closed-loop system Markov parameters $Y_{sc}(k)$ and the Kalman filter Markov parameters $Y_{kc}(k)$, and the known controller Markov parameters $Y_d(k)$; one can derive the open-loop system Markov parameters $Y_s(k) = CA^{k-1}B$ and Kalman filter Markov parameters $Y_k(k) = CA^{k-1}AK$.

$$Y_s(j) = Y_{sc}(j) - \sum_{k=1}^{j} \sum_{i=1}^{k} Y_s(i) Y_d(k-i) Y_{sc}(j-k) \tag{2.48}$$

$$Y_k(j) = Y_{kc}(j) - \sum_{k=1}^{j} \sum_{i=1}^{k} Y_s(i) Y_d(k-i) Y_{kc}(j-k) \tag{2.49}$$

Note that $Y_{sc}(0) = 0$, and $Y_{kc}(0) = I$

After obtaining the open-loop system and Kalman filter Markov parameters, one can use the method mentioned in section 2.4.2 to realize the open-loop system matrices and recover the open-loop Kalman filter gain.

2.5.2 Computation Steps for CLID

1. Collect reference input and output data from the experiment.
2. Estimate the closed-loop ARX model parameters c_i, d_i from Equations (2.41) to (2.43) by choosing suitable ARX model order q.
3. Compute the closed-loop system and Kalman filter Markov parameters from the estimated coefficient matrices c_i, d_i by using (2.46) and (2.47).
4. Calculate the open-loop system and Kalman filter Markov parameters from the closed-loop system Markov parameters, the closed-loop Kalman filter Markov parameters, and the controller Markov parameters computed from the known controller dynamics, by using (2.48) and (2.49).
5. Recover the open-loop system matrices by using the eigensystem realization method, Equations(2.25) to (2.27) and the open-loop Kalman filter gain by Equation (2.28).

2.6 Coordinate Transformation

For any dynamics system, although its system Markov parameter is unique, the realized state-space model is in a different coordinate. In order to compare the identified state-space model with the analytical model, both models need to be in the same coordinate. In this section, a unique coordinate transformation matrix is derived to transform any realized state-space model to be in a form normally used for a structural dynamic system, so that both models, the identified and the analytical models, can be compared in the same coordinate. This transformation matrix, however, exists only if one half of the states can be measured directly. If this condition is not satisfied, other transformation matrices may exist, but they usually are not unique.

Consider a structural dynamic system

$$M\ddot{p} + D\dot{p} + Kp = Gu \tag{2.50}$$

where p is a displacement, u the control force, G the control influence matrix, and M, D, and K are the mass, damping, and stiffness matrices, respectively. One can get state-space model as

$$\dot{x} = A_m x + B_m u \tag{2.51 a}$$

$$y = C_m x \tag{2.51 b}$$

where $A_m = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{bmatrix}$, $B_m = \begin{bmatrix} 0 \\ M^{-1}G \end{bmatrix}$, C_m the output matrix and $x = \begin{bmatrix} p \\ \dot{p} \end{bmatrix}$.

Assume the displacement p can be measured, then $C_m = \begin{bmatrix} I & 0 \end{bmatrix}$. First, one needs to convert the realized discrete-time system matrices $\begin{bmatrix} A & B & C \end{bmatrix}$ to continuous-time system matrices $\begin{bmatrix} A_s & B_s & C \end{bmatrix}$. If A is diagonalized by matrix Θ, then

$$\Theta^{-1} A \Theta = \Lambda \tag{2.52}$$

$$A_s = \Theta \frac{\ln(\Lambda)}{T} \Theta^{-1} \tag{2.53}$$

$$B_s = (A - I)^{-1} A_s B \tag{2.54}$$

where T is the sampling time. It is assumed that the matrix $\begin{bmatrix} C \\ CA_s \end{bmatrix}$ is full rank. Let P be the transformation matrix

$$P = \begin{bmatrix} P_1 & P_2 \end{bmatrix} = \begin{bmatrix} C \\ CA_s \end{bmatrix}^{-1} \tag{2.55}$$

then

$$P^{-1}P = \begin{bmatrix} C \\ CA_s \end{bmatrix} \begin{bmatrix} P_1 & P_2 \end{bmatrix} = \begin{bmatrix} CP_1 & CP_2 \\ CA_sP_1 & CA_sP_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

$$P^{-1}A_sP = \begin{bmatrix} CA_sP_1 & CA_sP_2 \\ CA_s^2P_1 & CA_s^2P_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ X & X \end{bmatrix} \tag{2.56}$$

$$P^{-1}B_s = \begin{bmatrix} CB_s \\ CA_sB_s \end{bmatrix} = \begin{bmatrix} C_mB_m \\ CA_sB_s \end{bmatrix} = \begin{bmatrix} 0 \\ X \end{bmatrix} \tag{2.57}$$

$$CP = \begin{bmatrix} CP_1 & CP_2 \end{bmatrix} = \begin{bmatrix} I & 0 \end{bmatrix} \tag{2.58}$$

Note that $CP = C_m$. According to above, the identified continuous-time model $\begin{bmatrix} A_s & B_s & C \end{bmatrix}$ can be transformed to be $\begin{bmatrix} P^{-1}A_sP & P^{-1}B & CP \end{bmatrix}$ which is in the form of Equation (2.51). So, now both models are in the same coordinate, and can be compared.

CHAPTER III

INDIRECT IDENTIFICATION THROUGH RESIDUAL WHITENING

3.1 Introduction

In theory, when one identifies a system from input-output data in a time domain. it is assumed that the data length is long enough and the ARX model order is sufficiently large. Then the identified observer tends to be the optimal Kalman filter gain in the presence of process and measurement noise. Under these conditions, the resultant residual of the filter is minimized, uncorrelated with the input and output data, and also white. However, in practice, since only a finite set of data is available and choosing the large ARX model order is limited by the computation constraint, the identified system model and observer may not be optimal. In Reference 7, the residual whitening method is used to improve observer/Kalman identification (OKID). In this chapter, the residual whitening method will be performed in closed-loop system identification (CLID) indirect method. Through residual whitening, the optimal properties of the Kalman filter could be enforced for a finite set of data. This means that for a given set of finite data, one can identify the system and observer whose residual is minimized, orthogonal to the time-shifted versions of itself and to the given set of input-output data, instead of being uncorrelated to input-output data and white. The rationale for these conditions is that when the data length tends to infinity, the orthogonal conditions which are imposed on the residual sequence will convert to the optimal properties of the Kalman filter. And one can also relax the requirement of the model order to reduce the computation burden, especially for several input and output systems.

In section 3.2, the ARX model of the CLID indirect method will be converted to ARMAX model by adding and subtracting the term My_k for reducing the requirement of the model order. The basic difference between ARX model and ARMAX model is that the ARMAX model includes dynamics of noise. The residual whitening for ARMAX model is introduced in section 3.3. The orthogonal conditions that the residual is orthogonal to the time-shifted versions of itself and to the given set of input-output data are discussed in section 3.4. In section 3.5, it shows the iterative procedure for residual whitening identification. In section 3.6, how to recover open-loop system and Kalman filter Markov parameters from residual whitening process for the ARMAX model is introduced. After obtaining the open-loop system and Kalman filter Markov parameters, one can use the method mentioned in section 2.4.2 to realize the open-loop system matrices and recover the open-loop Kalman filter gain. Finally, in section 3.7, the NASA Large Angle Magnetic Suspension Test Facility (LAMSTF) is briefly introduced and used as the numerical example for indirect CLID residual whitening. Several tables and figures will be shown in this section.

3.2 Model Structures

The indirect CLID algorithm has been introduced in section 2.5.1. For convenience, some of them will be rewritten in this section. A finite-dimensional, linear, discrete-time, time invariant stochastic system can be expressed as:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{3.1}$$

$$y_k = Cx_k + v_k \tag{3.2}$$

where $x_k \in R^{n \times 1}$ is the state vector, $u \in R^{ni \times 1}$ is the signal input vector, and $y \in R^{no \times 1}$ is the output vector; $[A, B, C]$ are the system matrices. The sequence of process noise $w \in R^{n \times 1}$ and measurement noise $v \in R^{no \times 1}$ are assumed white, Gaussian, zero mean. The noises w_k and v_k are also assumed uncorrelated with covariance Q and R, respectively.

Equations (3.1), (3.2) can also be expressed as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + AK\varepsilon_k \tag{3.3 a}$$

$$y_k = C\hat{x}_k + \varepsilon_k \tag{3.3 b}$$

where $\varepsilon_k \in R^{no \times 1}$ is the output residual and equals $\varepsilon_k = y_k - \hat{y}_k$; it's zero mean, white, Gaussian. $\hat{x}_k$ and $\hat{y}_k$ are the estimated state and output, respectively. $K \in R^{n \times no}$ is the steady-state Kalman filter gain which satisfies $K = PC^T[R + CPC^T]^{-1}$. $P \in R^{n \times n}$ is the solution of the steady-state algebraic Riccati equation. The existence of K is guaranteed if the system is detectable and $\left(A, Q^{\frac{1}{2}}\right)$ is stabilizable.

Here, a dynamic feedback controller is added to ensure the system's stability. A dynamic feedback controller can be expressed as:

$$p_{k+1} = A_d p_k + B_d y_k \tag{3.4}$$

$$u_k = C_d p_k + D_d y_k + r_k \tag{3.5}$$

where p_k, r_k, $\left[A_d \ B_d \ C_d \ D_d\right]$ are the controller state, reference input to the closed-loop system and state-space matrices of the controller, respectively.

Taking Equations (3.3), (3.4), (3.5) together, the augmented closed-loop system dynamics becomes

$$\eta_{k+1} = A_c \eta_k + B_c r_k + A_c K_c \varepsilon_k \tag{3.6}$$

$$y_k = C_c \eta_k + \varepsilon_k \tag{3.7}$$

where

$$\eta_k = \begin{bmatrix} \hat{x}_k \\ p_k \end{bmatrix}, \quad A_c = \begin{bmatrix} A + BD_d C & BC_d \\ B_d C & A_d \end{bmatrix}, \quad B_c = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

$$A_c K_c = \begin{bmatrix} AK + BD_d \\ B_d \end{bmatrix}, \quad \text{and } C_c = \begin{bmatrix} C & 0 \end{bmatrix}$$

In order to reduce the requirement of model order, the term My_k will be added and subtracted to the right hand side Equation (3.6) to yield

$$\begin{aligned} \eta_{k+1} &= A_c \eta_k + B_c r_k + A_c K_c \varepsilon_k + My_k - My_k \\ &= A_c \eta_k + B_c r_k + A_c K_c \varepsilon_k + M(C_c \eta_k + \varepsilon_k) - My_k \\ &= (A_c + MC_c)\eta_k + (A_c K_c + M)\varepsilon_k + B_c r_k - My_k \end{aligned} \tag{3.8}$$

and

$$y_k = C_c\eta_k + \varepsilon_k \tag{3.9}$$

Now the new relationship of reference input and output becomes

$$y_k = \sum_{i=1}^{\infty} C_c\tilde{A}_c^{i-1}(-M)y_{k-i} + \sum_{i=1}^{\infty} C_c\tilde{A}_c^{i-1}B_c r_{k-i} + \sum_{i=1}^{\infty} C_c\tilde{A}_c^{i-1}\overline{M}\varepsilon_{k-i} + \varepsilon_k \tag{3.10}$$

where

$$\tilde{A}_c = A_c + MC_c,\ \overline{M} = M + A_cK_c \tag{3.11}$$

Make $\tilde{A}_c$ asymptotically stable, $\tilde{A}_c^{i-1} \approx 0$ if $i \ge p$ for a sufficient large number p, Equation (3.10) becomes

$$y_k = \sum_{i=1}^{p} h_i y_{k-i} + \sum_{i=1}^{p} t_i r_{k-i} + \sum_{i=1}^{p} s_i\varepsilon_{k-i} + \varepsilon_k \tag{3.12}$$

where

$$h_i = C_c\tilde{A}_c^{i-1}(-M),\ t_i = C_c\tilde{A}_c^{i-1}B_c,\ s_i = C_c\tilde{A}_c^{i-1}\overline{M},\ i = 1, 2, \ldots, p$$

Equation (3.12) is an ARMAX model containing the dynamics of residual which is different with Equation (2.41). To see Equation (3.12) is a particular ARMAX model more clearly, define a delay operator q^{-1}, such that $z_{k-1} = q^{-1}z_k$. Equation (3.12) becomes

$$D(q^{-1})y_k = E(q^{-1})u_k + F(q^{-1})\varepsilon_k \tag{3.13}$$

where the delay operator polynomials are

$$D(q^{-1}) = I + C_cMq^{-1} + C_c\tilde{A}_cMq^{-2} + \ldots + C_c\tilde{A}_c^{p-1}Mq^{-p}$$

$$E(q^{-1}) = C_cB_cq^{-1} + C_c\tilde{A}_cB_cq^{-2} + \ldots + C_c\tilde{A}_c^{p-1}B_cq^{-p}$$

$$F(q^{-1}) = I + C_c\overline{M}q^{-1} + C_c\tilde{A}_c\overline{M}q^{-2} + \ldots + C_c\tilde{A}_c^{p-1}\overline{M}q^{-p}$$

As stated in (3.11), $\tilde{A}_c = A_c + MC_c$, the matrix M is used to make the new system Equations (3.8) (3.9) more stable than the original one in Equations (3.6) and (3.7), especially, since it can be used to reduce the requirement of ARMAX model order p. In the original system (3.6) (3.7), K_c is working as an observer gain which includes Kalman filter gain K. We don't want to lose the information about the Kalman filter by making the observer equation more stable than the filter. The matrix M, however, could be chosen such that $\tilde{A}_c = A_c + MC_c$ is as stable as it could be without losing the Kalman filter information. In other words, we can choose appropriate matrix M to limit the number of ARMAX model order which needn't be several times larger than the true order of the system. In Equation (3.13) the relationship between K_c and M is clearly defined, where K_c appears only in the noise model.

3.3 Residual Whitening

Suppose there are N data points of y_k and u_k are given, $k = 0, 1, \ldots, N-1$. Define

$$\theta = \begin{bmatrix} t_1 & h_1 & \ldots & t_p & h_p \end{bmatrix}, \Psi = \begin{bmatrix} s_1 & s_2 & \ldots & s_p \end{bmatrix}$$

$$W = \begin{bmatrix} 0 & \varepsilon_0 & \varepsilon_1 & \ldots & \varepsilon_{p-1} & \ldots & \varepsilon_{N-2} \\ 0 & 0 & \varepsilon_0 & \ldots & \varepsilon_{p-2} & \ldots & \varepsilon_{N-3} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & 0 & \varepsilon_0 & \ldots & \varepsilon_{N-p-1} \end{bmatrix}, R = \begin{bmatrix} \varepsilon_0 & \varepsilon_1 & \ldots & \varepsilon_{N-1} \end{bmatrix}$$

$$V_{clid} = \begin{bmatrix} 0 & r_0 & \ldots & r_{p-1} & \ldots & r_{N-2} \\ 0 & y_0 & \ldots & y_{p-1} & \ldots & y_{N-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 0 & r_0 & \ldots & r_{N-p-1} \\ 0 & \ldots & 0 & y_0 & \ldots & y_{N-p-1} \end{bmatrix}, Y = \begin{bmatrix} y_0 & y_1 & \ldots & y_{N-1} \end{bmatrix}$$

Then Equation (3.12) could be written in a matrix form

$$Y = \theta V_{clid} + \Psi W + R \tag{3.14}$$

The residual sequence R equals

$$R = Y - \theta V_{clid} - \Psi W$$

$$= Y - \begin{bmatrix} \theta & \Psi \end{bmatrix} \begin{bmatrix} V_{clid} \\ W \end{bmatrix} \tag{3.15}$$

The cost function

$$J = \sum_{k=0}^{N-1} \varepsilon_k^T \varepsilon_k = tr(RR^T) \tag{3.16}$$

Assuming the residual matrix W is available at this moment, then the least squares solution of the parameters $\hat{\theta}$ and $\hat{\Psi}$ that minimize the cost function J is

$$\begin{bmatrix} \hat{\theta} & \hat{\Psi} \end{bmatrix} = Y\begin{bmatrix} V_{clid} \\ W \end{bmatrix}^{+} = Y\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix}\begin{bmatrix} V_{clid}V_{clid}^T & V_{clid}W^T \\ WV_{clid}^T & WW^T \end{bmatrix}^{-1} \tag{3.17}$$

One can derive the parameter $\hat{\theta}$ to be a sum of ordinary least squares indirect CLID and a bias term[36]. To see this, first expand the inverse matrix directly

$$\begin{bmatrix} V_{clid}V_{clid}^T & V_{clid}W^T \\ WV_{clid}^T & WW^T \end{bmatrix}^{-1} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}$$

where

$$Q_{11} = (VV^T)^{-1}[I + VW^T\lambda^{-1}W\Lambda] \tag{3.18 a}$$

$$Q_{12} = -(VV^T)^{-1}VW^T\lambda^{-1} \tag{3.18 b}$$

$$Q_{21} = -\lambda^{-1}W\Lambda \tag{3.18 c}$$

$$Q_{22} = \lambda^{-1} \tag{3.18 d}$$

and

$$\Lambda = V^T(VV^T)^{-1},\ \lambda = W(I - \Lambda V)W^T \tag{3.19}$$

Combining (3.17) and (3.18) together, the estimated parameters could be expressed as

$$\hat{\theta} = Y\Lambda - \hat{\Psi}W\Lambda \tag{3.20}$$

$$\hat{\Psi} = Y(I - \Lambda V)W^T\lambda^{-1} \tag{3.21}$$

Examining the first term for $\hat{\theta}$ in Equation (3.20), it is the same as Equation (2.43) which is the ordinary least squares solution for indirect CLID; and in the second term one may consider it as a bias term. One can write, hence,

$$\hat{\theta} = \theta^{LS} - \theta^{bias} \tag{3.22}$$

where

$$\theta^{LS} = Y\Lambda = YV^T(VV^T)^{-1}$$

$$\theta^{bias} = \Psi\hat{W}\Lambda = \hat{\Psi}WV^T(VV^T)^{-1}$$

Equation(3.20) could also be written as

$$\hat{\Psi} = Y(I - \Lambda V)W^T\lambda^{-1}$$

$$= (Y - \theta^{LS}V)W^T[WW^T - WV^T(VV^T)^{-1}VW^T]^{-1}$$

$$= \hat{e}^{LS}W^T[WW^T - WV^T(VV^T)^{-1}VW^T]^{-1} \tag{3.23}$$

where $\hat{e}^{LS}$ denotes the colored residual between the measurement Y and the least squares estimation from indirect CLID $\theta^{LS}V_{clid}$.

The bias term θ^{bias} in Equation (3.22) is composed of residual. In the deterministic case $W = 0$, the bias equals zero and the parameter $\hat{\Psi}$ will also be zero. The system, here, could be accurately identified regardless of the choice of M as long as $\tilde{A}_c = A_c + MC_c$ is stable enough and the ARMAX model order is chosen large enough. Normally, the dead beat gain M is not related to the Kalman gain K which is contained in K_c. Both M and K_c appear only in the noise model $C(q^{-1})$. The dead beat gain M could be used to make $\tilde{A}_c$ more stable, so we can restrict the number of ARMAX model order. This is the advantage of adding the gain M.

3.4 Properties of the Residual Sequence

The residual sequence, minimizing the cost function $J = \sum_{k=0}^{N-1} \varepsilon_k^T \varepsilon_k$ in the least squares sense, has two other properties.

First property: The residual sequence is orthogonal to the reference input and output data sequence. It is described by the following equations.

$$\sum_{k=0}^{N-1} \varepsilon_k r_k^T = 0, \quad \sum_{k=i}^{N-1} \varepsilon_k r_{k-i} = 0$$

$$\sum_{k=i}^{N-1} \varepsilon_k y_{k-i}^T = 0, \; i = 1, 2, \ldots, p \tag{3.24}$$

Second property: The residual sequence is also orthogonal to the time-shifted versions of itself.

$$\sum_{k=i}^{N-1} \varepsilon_k \varepsilon_{k-i}^T = 0, \; i = 1, 2, \ldots, p \tag{3.25}$$

These properties could be identified by multiplying both sides of Equation (3.15) with $\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix}$, and replace $\begin{bmatrix} \theta & \Psi \end{bmatrix}$ by the least squares solution of Equation (3.17) to yield as the following

$$R\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix} = Y\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix} - Y\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix}$$

$$\times\left(\begin{bmatrix} V_{clid} \\ W \end{bmatrix}\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix}\right)^{-1}\left(\begin{bmatrix} V_{clid} \\ W \end{bmatrix}\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix}\right)$$

$$= Y\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix} - Y\begin{bmatrix} V_{clid}^T & W^T \end{bmatrix} = 0 \tag{3.26}$$

Equation (3.26) indicates $RV_{clid}^T = 0$, $RW^T = 0$ which yield the properties of the residual sequence.

3.5 Iterative Procedure for Identification

In this section, an iterative identification procedure for ARMAX model parameters and residual sequence is shown. The initial estimate of residual sequence could be computed from the ordinary least squares solution of indirect CLID. Then one can identify the ARMAX model parameters. The iterative procedure continues when the residual sequence is updated and a new set of ARMAX model parameters is generated.

Step 1: An initial estimate of the parameter θ, denoted by θ^{LS}, is computed from the ordinary least squares solution. And the other parameter Ψ is assumed to be zero in the beginning.

$$\hat{\theta}^{LS} = YV_{clid}^T(V_{clid}V_{clid}^T)^{-1} \tag{3.27}$$

$$\hat{\Psi}_0 = 0 \tag{3.28}$$

Equation (3.27) is exactly the same as Equation (2.43) which is the ordinary least squares solution for indirect CLID.

Step 2: Calculate the colored residual sequence $\hat{e}^{LS}$ which corresponds to the initial estimate ARMAX model parameter $\hat{\theta}^{LS}$.

$$\hat{e}^{LS} = Y - \hat{\theta}^{LS}V_{clid} \tag{3.29}$$

It is noted that $\hat{e}^{LS}$ is also the initial estimate of white residual sequence denoted by $\hat{e}_0 = \hat{R}_0$.

Step 3: Construct the residual matrix $\hat{W}_0$ which is composed of residual sequence by using the estimated residual sequence $\hat{R}_0$ according to Equation (3.14). Then the parameter $\hat{\Psi}$ could be updated by Equation (3.23).

$$\hat{\Psi}_1 = \hat{e}^{LS}\hat{W}_0^T[\hat{W}_0\hat{W}_0^T - \hat{W}_0V_{clid}^T(V_{clid}V_{clid}^T)^{-1}V_{clid}\hat{W}_0^T]^{-1} \tag{3.30}$$

The updated parameter $\hat{\Psi}_1$ is used to correct the initial bias as follows

$$\hat{\theta}_1^{bias} = \hat{\Psi}\hat{W}_0\Lambda = \hat{\Psi}_1\hat{W}_0V_{clid}^T(V_{clid}V_{clid}^T)^{-1} \tag{3.31}$$

$$\hat{\theta}_1 = \hat{\theta}^{LS} - \hat{\theta}_1^{bias} \tag{3.32}$$

Step 4: Compute the new whitened residual sequence $\hat{R}_1$ by using the estimated parameters $\hat{\theta}_1$ and $\hat{\Psi}_1$ as follows

$$\hat{R}_1 = Y - \hat{\theta}_1V_{clid} - \hat{\Psi}_1\hat{W}_0 \tag{3.33}$$

Step 5: Iterate the procedure from step 3 to step 4 by generating the new residual matrix $\hat{W}_1$ and using the updated parameters $\hat{\theta}_1$ and $\hat{\Psi}_1$. The next cycle is calculated as follows, for example,

$$\hat{\Psi}_2 = \hat{e}^{LS}\hat{W}_1^T[\hat{W}_1\hat{W}_1^T - \hat{W}_1 V_{clid}^T(V_{clid}V_{clid}^T)V_{clid}\hat{W}_1^T]^{-1} \tag{3.34}$$

and

$$\hat{\theta}_2^{bias} = \hat{\Psi}_2\hat{W}_1\Lambda = \hat{\Psi}_2\hat{W}_1 V_{clid}^T(V_{clid}V_{clid}^T)^{-1} \tag{3.35}$$

$$\hat{\theta}_2 = \hat{\theta}^{LS} - \hat{\theta}_2^{bias} \tag{3.36}$$

3.6 Recover System Matrices and Kalman Filter Gain

After having obtained the ARMAX model estimated parameters θ, and Ψ, one can use the estimated coefficients to construct the closed-loop system Markov parameters $Y_{sc}(k) = C_c A_c^{k-1} B_c$, Kalman filter Markov parameters $Y_{kc}(k) = C_c A_c^{k-1} A_c K_c$, and $Y_{mc}(k) = C_c A_c^{k-1} M$, $k = 1, 2, \ldots, p, p+1, \ldots$ It is noted that $Y_{sc}(0) = 0$, $Y_{kc}(0) = I$, and $t_i = h_i = s_i = 0$ when $i > p$, where p is the ARMAX model order, and t_i, h_i, s_i are the estimated ARMAX model parameters.

Remember that

$$\theta = \begin{bmatrix} t_1 \ h_1 \ \ldots \ t_p \ h_p \end{bmatrix}, \ \Psi = \begin{bmatrix} s_1 \ s_2 \ \ldots \ s_p \end{bmatrix}$$

$$\tilde{A}_c = A_c + MC_c, \ \overline{M} = M + A_c K_c$$

$$h_i = C_c\tilde{A}_c^{i-1}(-M), \ t_i = C_c\tilde{A}_c^{i-1}B_c, \ s_i = C_c\tilde{A}_c^{i-1}\overline{M}, \ i = 1, 2, \ldots, p$$

One can first obtain Markov parameters sequence $Y_{sc}(k) = C_c A_c^{k-1} B_c$, and $Y_{mc}(k) = C_c A_c^{k-1} M$ by following

$$Y_{sc}(k) = t_k + \sum_{i=1}^{k} h_i Y_{sc}(k-i) \tag{3.37}$$

$$Y_{mc}(k) = -h_k + \sum_{i=1}^{k-1} h_i Y_{mc}(k-i) \tag{3.38}$$

Next, for the closed-loop Kalman filter Markov parameters $Y_{kc} = C_c A_c^{k-1} A_c K_c$, it's derived from both estimated parameters θ and Ψ. Note that

$$h_1 + s_1 = C_c(-M) + C_c(M + A_c K_c) = C_c A_c K_c$$

which is equal to $Y_{kc}(1)$

$$Y_{kc}(1) = h_1 + s_1 \tag{3.39}$$

To obtain $Y_{kc}(2)$, one can use

$$h_2 + s_2 = C_c(A_c + MC_c)A_c K_c = Y_{kc}(2) - h_1 Y_{kc}(1)$$

to yield

$$Y_{kc}(2) = h_2 + s_2 + h_1 Y_{kc}(1) \tag{3.40}$$

Similarly, one can find $Y_{kc}(3)$ by following

$$\begin{aligned} h_3 + s_3 &= C_c A_c^2 A_c K_c + C_c M C_c A_c A_c K_c + C_c \tilde{A}_c M C_c A_c K_c \\ &= Y_{kc}(3) - h_1 Y_{kc}(2) - h_2 Y_{kc}(1) \end{aligned}$$

to yield

$$Y_{kc}(3) = h_3 + s_3 + h_1 Y_{kc}(2) + h_2 Y_{kc}(1) \tag{3.41}$$

By induction, one can show that the sequence $Y_{kc}(k)$ equals

$$Y_{kc}(k) = h_k + s_k + \sum_{i=1}^{k-1} h_i Y_{kc}(k-i) \tag{3.42}$$

Then, by using the closed-loop system Markov parameters $Y_{sc}(k)$ and the Kalman filter Markov parameters $Y_{kc}(k)$, and the known controller Markov parameters $Y_d(k)$; one can derive the open-loop system Markov parameters $Y_s(k) = CA^{k-1}B$ and Kalman filter Markov parameters $Y_k(k) = CA^{k-1}AK$.

$$Y_s(j) = Y_{sc}(j) - \sum_{k=1}^{j} \sum_{i=1}^{k} Y_s(i) Y_d(k-i) Y_{sc}(j-k) \tag{3.43}$$

$$Y_k(j) = Y_{kc}(j) - \sum_{k=1}^{j} \sum_{i=1}^{k} Y_s(i) Y_d(k-i) Y_{kc}(j-k) \tag{3.44}$$

Note that $Y_s(0) = 0$, and $Y_k(0) = I$

After obtaining the open-loop system and Kalman filter Markov parameters, one can use the method mentioned in section 2.4.2 to realize the open-loop system matrices and recover the open-loop Kalman filter gain.

3.7 Numerical Simulations

In this section the NASA Large Angle Magnetic Suspension Test Facility (LAMSTF) is briefly introduced and used as an example for indirect CLID whitening residual. This facility has been assembled by NASA Langley Research Center for a ground-based experiment that can be used to develop and evaluate the technology issues associated with magnetic suspension at large gaps, accurate suspended-element control at large gaps, and accurate position sensing at large gaps. This technology is applicable to future efforts that range from magnetic suspension of wind-tunnel models to advanced spacecraft experiment isolation and pointing system[37]. The analytical model has been derived in detail in Reference 38 and 39. From open-loop eigenvalues, it has been found that there are three unstable modes and two stable oscillatory modes. Since the system is unstable, a controller is required to ensure overall system's stability.

This facility basically consists of five electromagnets (see figure 3.1) which actively suspend a small cylindrical permanent magnet. The cylinder is a rigid body and has six degrees of freedom, namely, three displacements (x, y and z) and three rotations (pitch, yaw and roll). The roll of the cylinder is uncontrollable and is assumed to be motionless. Five pairs of LEDs and photo detectors are used to indirectly sense the pitch and yaw angles, and three displacements of the cylinder's centroid. The inputs consist of five currents into five electromagnets and the outputs are five voltage (position) signals from the five photo detectors. Very briefly, the currents into the electromagnet generate a magnetic field which produces a net force and torque on the suspended cylinder.

The details of the suspended cylinder, the coils, power amplifiers, and the position sensors are described in the following section[40]. The mathematical mode of the system has been derived in detail in Reference 38 and 39. Only the final system matrices will be provided in a later section.

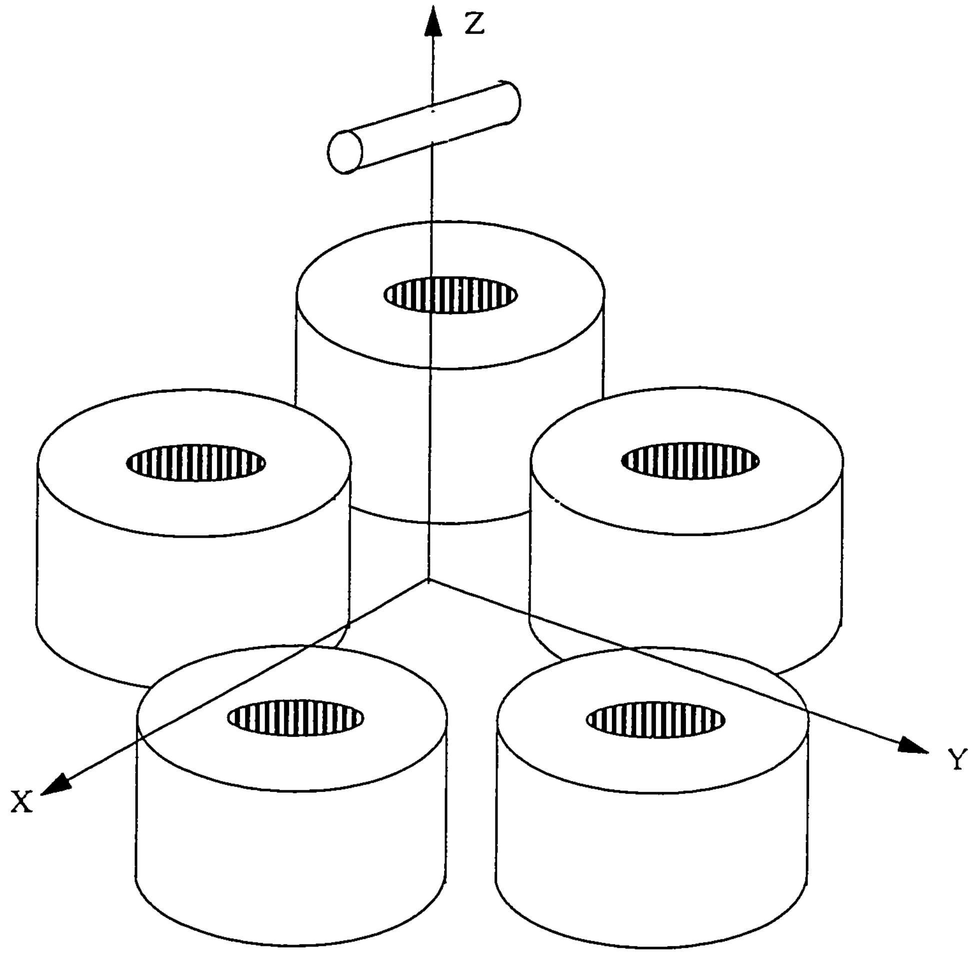


Figure 3.1 Large-Angle Magnetic Suspension Test Facility (LAMSTF) configuration

3.7.1 System Specifications

The suspended cylinder is an aluminum tube filled with 16 wafers of Neodymium-Iron-Boron permanent magnet material. Each magnetic wafer is 0.7963 cm in diameter and 0.3135 cm long, having a magnetization of about 9.5493×10^{-5} A/m. The suspended cylinder will be put at a height of about 10 cm above the coils.

There are five coils mounted on the circumference of a circle of about 13.77 cm radius, at a spacing of 72° apart, on a $1/2''$ thick, square aluminum plate. The current through the coils is controlled by five switching power amplifiers, capable of delivering a maximum of $30A$ continuous and $60A$ peak level. The amplifiers function in a voltage-to-current converter mode and are set in a gain of 3 A/V to have a flat response.

The detection of the suspended cylinder's position is performed by five sets of infrared LEDs and photo detectors. These LED-photo detector pairs are installed in two perpendicular planes (vertical and horizontal), which allow detection of five degrees-of-freedom of the cylinder. The beams from the infrared LEDs, which are incident on the photo detectors, would be partially blocked by the cylinder. The relative position of the cylinder can then be determined from the amount of light received by the photo detectors.

3.7.2 System Model

The analytical model has been derived in detail in Reference 38 and 39. Here, the discrete time state-space parameters of LAMSTF are shown for sampling rate of 250 Hz as follows:

A finite-dimensional, linear, discrete-time, time invariant stochastic system can be expressed as:

$$x_{k+1} = Ax_k + Bu_k + w_k$$

$$y_k = Cx_k + v_k$$

$$A = \begin{bmatrix} A_{11} & A_{12} \end{bmatrix},$$

$$A_{11} = \begin{bmatrix} 1.1687 & 0.0006 & -0.0000 & 0.0000 & 0.0000 \\ -0.0000 & 1.1629 & -0.0000 & -0.0000 & -0.0000 \\ -0.0000 & 0.0001 & 1.0178 & -0.0017 & -0.0037 \\ -0.0000 & 0.0000 & 0.0001 & 1.0051 & 0.0001 \\ 0.0000 & 0.0002 & -0.0004 & 0.0008 & 1.0106 \\ 0.0000 & -0.0000 & -0.0021 & -0.0240 & 0.0005 \\ 0.0000 & -0.0001 & -0.0064 & -0.0001 & -0.0213 \\ -0.0000 & -0.0000 & 0.0109 & -0.0009 & -0.0045 \\ 0.0000 & -0.0000 & -0.0086 & 0.0009 & 0.0032 \\ 0.0000 & -0.0000 & 0.0004 & 0.0002 & 0.0006 \end{bmatrix}$$

$$A_{12} = \begin{bmatrix} 0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0633 \\ -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0396 \\ 0.0021 & 0.0074 & -0.0127 & 0.0112 & 0.0254 \\ 0.0295 & 0.0006 & 0.0015 & -0.0011 & -0.0218 \\ -0.0018 & 0.0223 & 0.0066 & -0.0039 & -0.0242 \\ 0.9908 & 0.0028 & -0.0010 & 0.0003 & -0.0154 \\ -0.0041 & 0.9692 & 0.0064 & 0.0004 & -0.0066 \\ 0.0021 & 0.0050 & 0.9260 & -0.0549 & 0.0625 \\ 0.0009 & 0.0031 & -0.0589 & 0.9125 & 0.1245 \\ 0.0012 & 0.0545 & -0.0002 & -0.0002 & -0.1009 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0035 & 0.0706 & 0.0519 & -0.0363 & -0.0633 \\ -0.0434 & -0.0326 & -0.0340 & -0.0425 & -0.0396 \\ 0.0580 & -0.0454 & 0.0983 & -0.0361 & 0.0254 \\ -0.0926 & -0.0315 & 0.0881 & 0.0865 & -0.0218 \\ 0.1160 & 0.0124 & 0.0263 & 0.0982 & -0.0242 \\ -0.1015 & -0.0368 & 0.1033 & 0.0854 & -0.0154 \\ 0.1373 & 0.0057 & 0.0719 & 0.0859 & -0.0066 \\ -0.0159 & -0.0637 & -0.1326 & 0.1165 & 0.0625 \\ 0.0158 & -0.1531 & -0.0261 & 0.0041 & 0.1245 \\ -0.0484 & -0.0800 & -0.0513 & -0.0553 & -0.1009 \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} \end{bmatrix}$$

$$C_{11} = \begin{bmatrix} -0.0313 & 0.4029 & -0.0469 & 0.2269 & -0.0381 \\ 0.0291 & -0.4213 & 0.0006 & 0.2248 & 0.0290 \\ -0.4423 & 0.1071 & 0.1809 & 0.0553 & 0.0669 \\ -0.4254 & -0.1184 & -0.1787 & -0.0092 & -0.0829 \\ 0.4495 & -0.0763 & 0.0754 & 0.0273 & -0.1861 \\ 0.3889 & 0.1015 & -0.0614 & 0.0085 & 0.1739 \end{bmatrix}$$

$$C_{12} = \begin{bmatrix} -0.1961 & 0.1274 & -0.0363 & 0.0198 & -0.1513 \\ -0.2097 & -0.1079 & -0.0130 & 0.0297 & 0.1502 \\ -0.0618 & -0.0906 & -0.0418 & -0.2228 & -0.0472 \\ 0.0200 & 0.1217 & -0.2197 & -0.0559 & 0.0630 \\ -0.0400 & 0.1239 & 0.2109 & 0.0827 & 0.0464 \\ 0.0012 & -0.1277 & 0.0386 & 0.1913 & -0.0634 \end{bmatrix}$$

For simulation, the discrete-time state-space parameters of dynamics output feedback controller can be modeled as:

$$
\begin{aligned}
p_{k+1} &= A_d p_k + B_d y_k \\
u_k &= C_d p_k + D_d y_k + r_k
\end{aligned}
$$

The matrices are

$$
A_d = \begin{bmatrix} 0.3333 & 0 & 0 & 0 & 0 \\ 0 & 0.3333 & 0 & 0 & 0 \\ 0 & 0 & 0.6000 & 0 & 0 \\ 0 & 0 & 0 & 0.6000 & 0 \\ 0 & 0 & 0 & 0 & 0.6000 \end{bmatrix}
$$

$$
B_d = \begin{bmatrix} -0.0206 & 0.0206 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0098 & -0.0098 & 0.0098 & 0.0098 \\ 0 & 0 & 0.0003 & -0.0003 & -0.0003 & 0.0003 \\ 0 & 0 & -0.0003 & 0.0003 & -0.0003 & 0.0003 \\ 0.0004 & 0.0004 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
C_d = 1.0e+03 \times \begin{bmatrix} 0.0796 & 0.0000 & 7.3872 & 0.0000 & -5.5493 \\ 0.1032 & 0.0716 & -5.9772 & 4.3222 & -1.7160 \\ 0.0886 & 0.0442 & 2.2836 & -6.9917 & 4.4907 \\ 0.0886 & -0.0442 & 2.2836 & 6.9917 & 4.4907 \\ 0.1032 & -0.0716 & -5.9772 & -4.3222 & -1.7160 \end{bmatrix}
$$

$$
D_d = \begin{bmatrix} 10.8171 & 3.9903 & -7.0133 & 7.0133 & 7.0133 & -7.0133 \\ 6.7151 & -2.1362 & 11.2687 & -8.3349 & -3.0144 & 0.0807 \\ -2.1923 & -9.7904 & -7.9381 & 9.7505 & -5.4144 & 3.6020 \\ -2.1923 & -9.7904 & 3.6020 & -5.4144 & 9.7505 & -7.9381 \\ 6.7151 & -2.1362 & 0.0807 & -3.0144 & -8.3349 & 11.2687 \end{bmatrix}
$$

In numerical simulations the number of model order (ARX for indirect CLID and ARMAX for residual whitening) is evaluated at 7, 15, 30, and the process and measurement noise (variance) varies from 0.01% to 20%. The number of data points is 5000. After the indirect CLID (CLID) and its residual whitening (CLID/rw) are performed, both sets of the open-loop system Markov parameters of the identified models are reconstructed. To evaluate the accuracy of the identified result, the reconstructed Markov parameters are compared with the true ones. Because there are five inputs and five outputs in this system, each open-loop system Markov parameter is a five-by-five matrix. To compare two matrices, F-norm is used here. Therefore, the error between the first 30 true and reconstructed open-loop system Markov parameters is defined as

$$\sum_{i=1}^{30} \frac{\left\|\hat{C}\hat{A}^{i-1}\hat{B} - CA^{i-1}B\right\|_F}{\left\|CA^{i-1}B\right\|_F} \tag{3.45}$$

where the head $\hat{}$ denotes the reconstructed ones, and the F-norm is defined as $\sqrt{\sum(diag(X^T \times X))}$, X is a matrix. Table 3.1 to Table 3.3 shows the error result of open-loop system Markov parameters of nonwhitening one (CLID) and whitening one (CLID/rw) compared with the true one. The CLID/rw solution is obtained after 4 cycles of iteration. Figure 3.2 to 3.5 shows the plots of convergence of whitened residual norm and auto-correlation of the whitened residual.

Table 3.1: Error Percentage of the Open-Loop System Markov Parameters (Model Order for ARX or ARMAX=7)

Noise (Variance)	$Y_s^T - Y_s^{CLID}$	$Y_s^T - Y_s^{CLID/rw}$
0.01%	1.0334%	1.0234%
1%	15.4462%	9.9266%
5%	34.4472%	14.3700%
10%	44.7847%	23.7707%
15%	49.2913%	34.8206%
20%	56.5067%	38.4626%

Table 3.2: Error Percentage of the Open-Loop System Markov Parameters (Model Order for ARX or ARMAX=15)

Noise (Variance)	$Y_s^T - Y_s^{CLID}$	$Y_s^T - Y_s^{\frac{CLID}{rw}}$
0.01%	0.9066%	1.0250%
1%	8.8467%	9.5389%
5%	12.3540%	13.0617%
10%	21.4697%	21.0527%
15%	29.3945%	30.8258%
20%	30.9603%	30.8680%

Table 3.3: Error Percentage of the Open-Loop System Markov Parameters (Model Order for ARX or ARMAX=30)

Noise (Variance)	$Y_s^T - Y_s^{CLID}$	$Y_s^T - Y_s^{\frac{CLID}{rw}}$
0.01%	0.9788%	1.2630%
1%	9.4574%	8.7023%
5%	13.6627%	15.8037%
10%	24.8745%	22.3979%
15%	31.2659%	30.8230%
20%	32.9591%	32.2228%

3.8 Concluding Remark

From Table 3.1, one can see the residual whitening one obtains a better result than the one without whitening. In Table 3.2 and 3.3, there is not much difference between these two, since the model order is large enough and the residual is almost white. Actually, for the high level of noise cases, the most significant error is from the original estimated parameters which have been corrupted by the disturbance. Hence, it's desired to design a filter which can reduce the influence of disturbances, then the system identification will have a more accurate model.

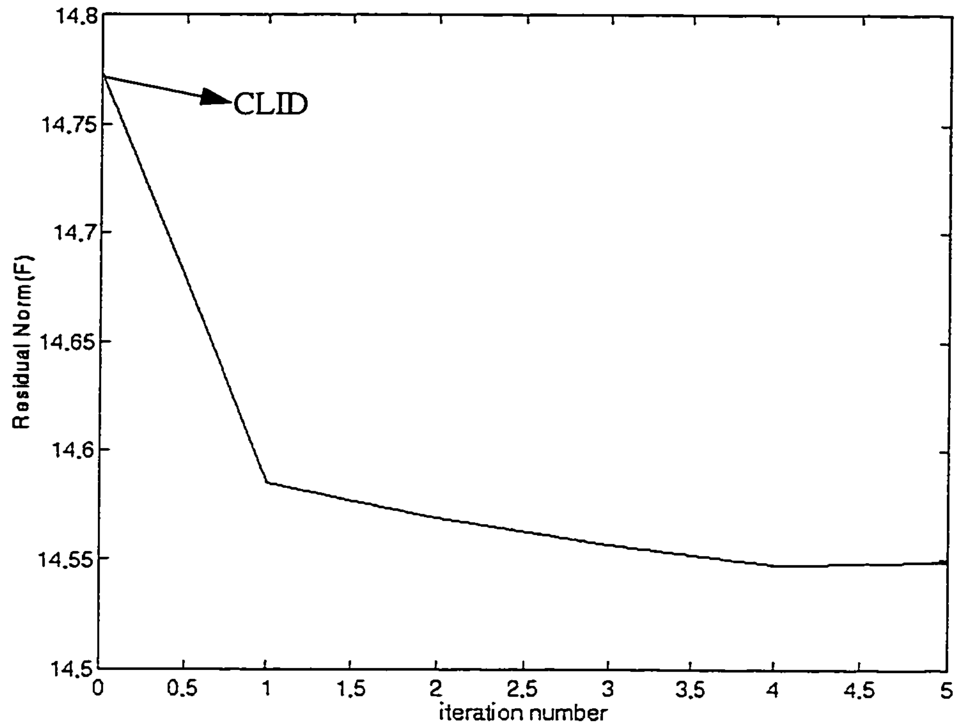


Figure 3.2 Convergence of whitened residual norm for ARX=ARMAX=7.

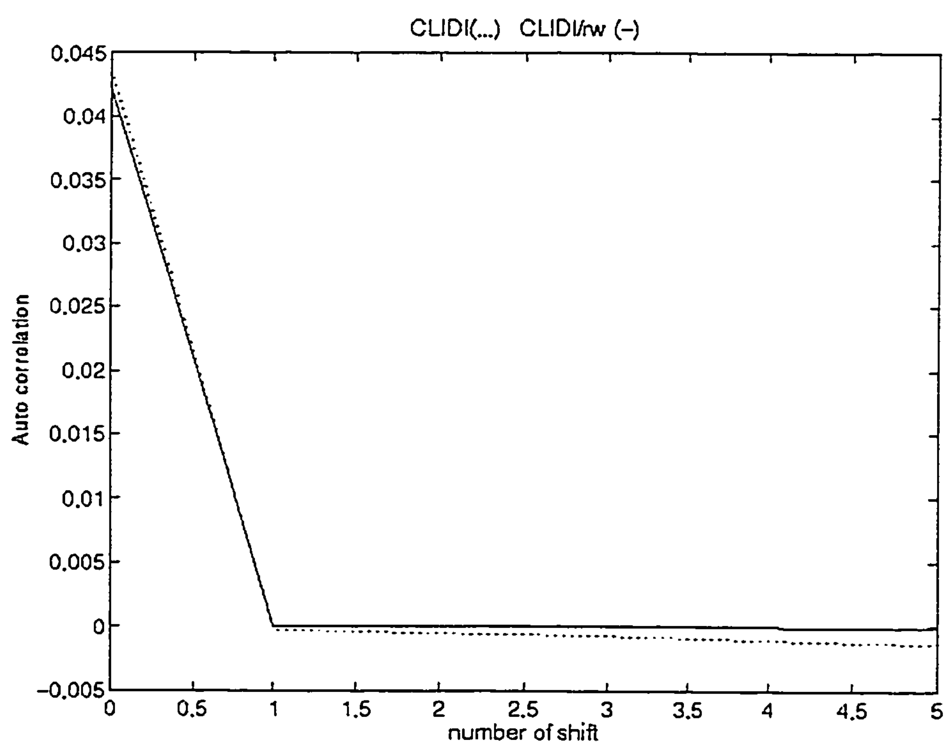


Figure 3.3 Auto-correlation of whitened residual for ARX=ARMAX=7.

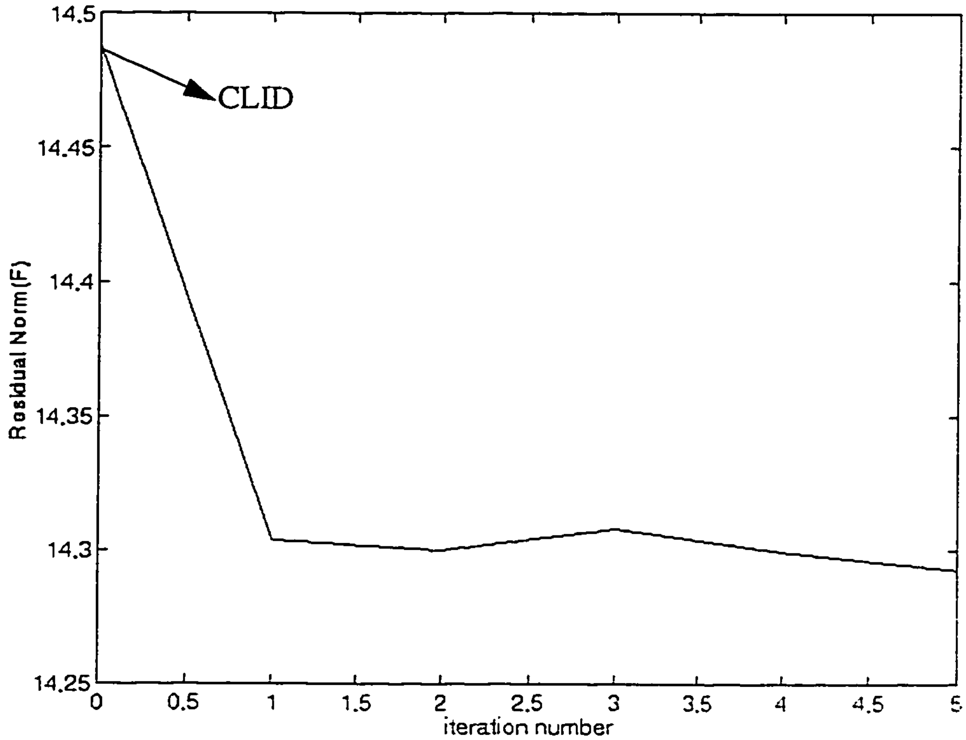


Figure 3.4 Convergence of whitened residual norm for ARX=ARMAX=15

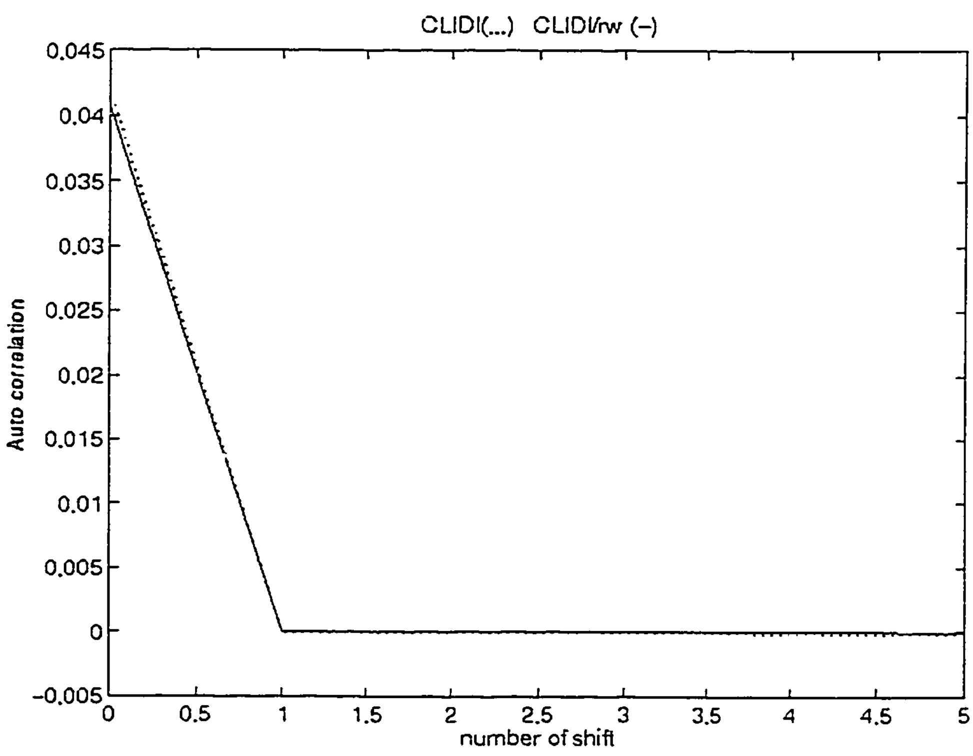


Figure 3.5 Auto-correlation of whitened residual for ARX=ARMAX=15.

CHAPTER IV

THE LQG CONTROL DESIGN

4.1 Introduction

This chapter introduces Linear Quadratic Gaussian (LQG) control design for both continuous and discrete-time domains, a discrete-time version of the Kalman filter for state estimation, and the procedure of the iterative LQG control design through closed-loop system identification.

The linear quadratic regulator (LQR) and the Kalman filter can be combined together to design a dynamic regulator. This procedure is called linear quadratic Gaussian (LQG) design. The performance of a closed-loop system can be arbitrarily adjusted only through full state feedback. In most of the realistic systems, we may not have a sensing system to measure all states. In such a case, output feedback control is required. The LQG control design is the most systematic approach to output feedback design. Another important advantage of LQG is that since the compensator structure is generated from the procedure, it needs not be known in advance. This makes LQG control design useful for controlling the complicated systems (e.g., space structure, aircraft engines), where one may not know the exact compensator structure. The LQG control separates the design into two stages, namely, state feedback (LQR) and state estimation (Kalman filter). Figure 4.1 plots an LQG control system. The system is a stochastic system, since the plant is disturbed by process noise and the output is corrupted by measurement noises. In state estimation the optimal estimate of state is established by using the information of the system output. To achieve this, one needs the statistics of the process and measurement noise to design a steady state Kalman filter gain which is used for state estimation. Then the estimated state through Kalman filter is used for state feedback. The state feedback is designed to minimize a performance index described by system states and inputs with weighting matrices. Basically the LQG uses the output information to accomplish state feedback through state estimation. By selecting proper design parameters of state estimation, the performance of the LQG control can achieve that of a true full state feedback. [41-43]

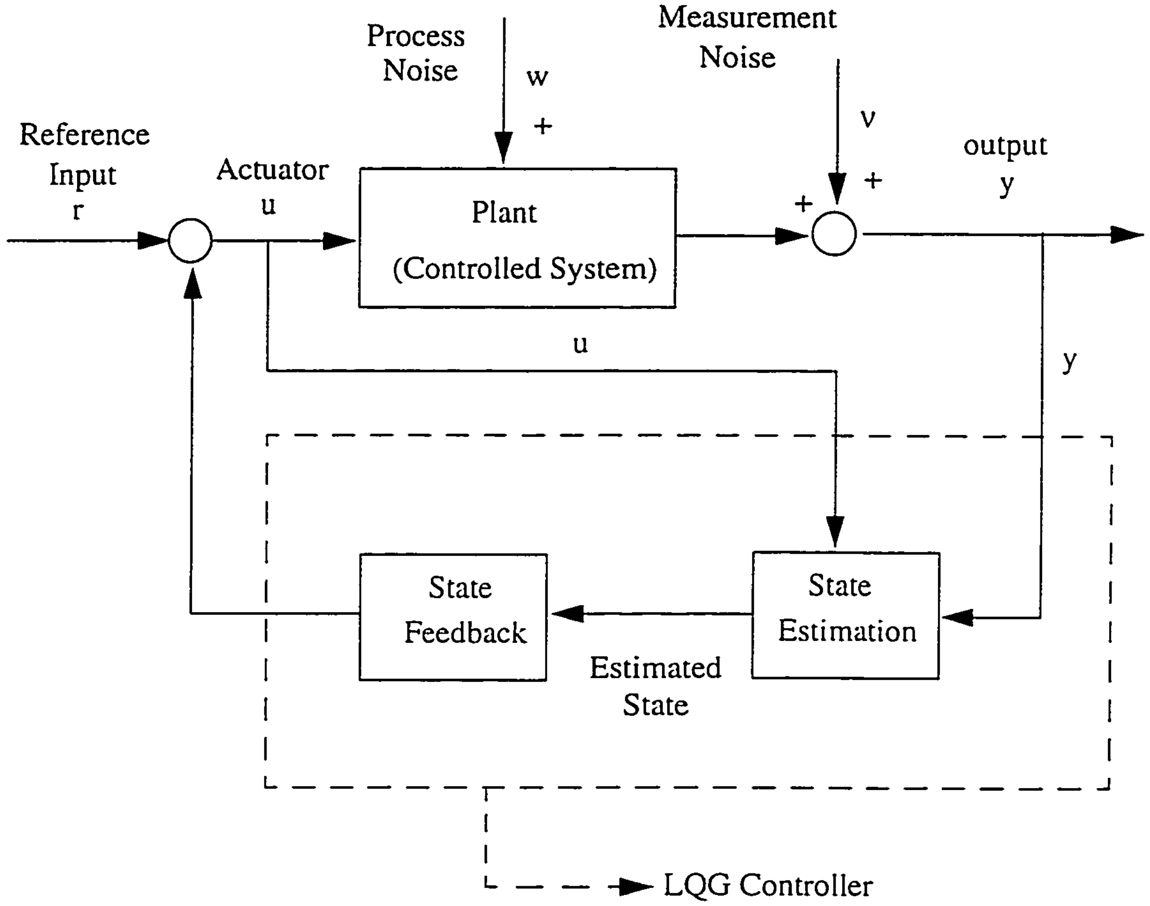


Figure 4.1 LQG Control System

4.2 Continuous Time Approach

A finite-dimensional, linear, discrete-time, time invariant stochastic system can be expressed as:

$$\dot{x} = Ax + Bu + w \tag{4.1}$$

$$y = Cx + v \tag{4.2}$$

where $x \in R^{n \times 1}$ is the state vector, $u \in R^{ni \times 1}$ is the signal input vector, and $y \in R^{no \times 1}$ is the output vector; $[A, B, C]$ are the system matrices. The sequence of process noise

$w \in R^{n \times 1}$ and measurement noise $v \in R^{no \times 1}$ are assumed white, Gaussian, zero mean. The noises w and v are also assumed uncorrelated with covariance Q and R, respectively.

First, for state feedback, LQG control is designed to minimize the performance index associated with state and input vectors

$$J = \int_0^{\infty} (x^T M x + u^T N u) dt \tag{4.3}$$

where M and N are weighting matrices. The matrices M and N determine the relative importance of the error and the expenditure of the energy of the control signals. Generally speaking, a larger M, which makes the error more important, gives less state error so that the system will respond quicker. A larger N, which makes the control signals more important, results in less input so that the system has lower response. In order to insure there is a solution to minimize the performance, M needs to be positive semidefinite and N needs to be positive definite. The optimal feedback control can be determined[44] as follows

$$u = -N^{-1} B^T S x \tag{4.4}$$

where S can be solved from the following algebraic Riccati equation

$$0 = A^T S + SA - SBN^{-1} B^T S + M \tag{4.5}$$

Second, for state estimation, the state estimation law is

$$\dot{\hat{x}} = A\hat{x} + Bu + G(y - C\hat{x}) \tag{4.6}$$

where G is the estimation gain and $\hat{x}$ is the estimated state. The estimated state can be corrected by the measurement output. The estimation gain is designed to minimize the covariance of the error between the true and estimated states

$$\int_0^{\infty} E[(x - \hat{x})(x - \hat{x})^T] \tag{4.7}$$

based on the covariance matrices of the process and measurement noises. The optimal estimation gain can be found[45] as

$$G = LC^T R^{-1} \tag{4.8}$$

where L can be solved from the following Riccati equation

$$0 = AL + LA^T + Q - LC^T R^{-1} CL \tag{4.9}$$

So, one can solve Equation (4.6) to have the estimated states and uses them to find out the optimal control input from Equation (4.4).

4.3 Discrete Time Approach

The recent revolutionary advances in digital computers, such as the great advances made in large-scale integration of semiconductors, the availability of low cost microprocessors and microcomputers which can be used for various control functions, and the development of suitable programming techniques, have increased influence on the techniques of system identification and control. It is easier to maintain and modify computer codes in digital computers than wire connections in analog computers, and also the digital computers provide high flexibility for implementing control laws as compared with analog computers. However, a sampling rate is required to allow digital computers to process control laws. In general, the sampling rate will affect the closed-loop system performance and stability. The sampling and quantizing processes tend to result in more errors which degrade system performance. If a low sampling rate is necessary for the process of a more complex control law, one may need to use discrete time approach for LQG control to maintain the performance and stability of the closed-loop system. Most system identification techniques are based on discrete time systems. The following introduces the discrete time LQG control.

A finite-dimensional, linear, discrete-time, time invariant stochastic system can be expressed as:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{4.10}$$

$$y_k = Cx_k + v_k \tag{4.11}$$

where $x_k \in R^{n \times 1}$ is the state vector, $u \in R^{ni \times 1}$ is the signal input vector, and $y \in R^{no \times 1}$ is the output vector; $[A, B, C]$ are the system matrices. The sequence of process noise $w \in R^{n \times 1}$ and measurement noise $v \in R^{no \times 1}$ are assumed white, Gaussian, zero mean. The noises w_k and v_k are also assumed uncorrelated with covariance Q and R, respectively. Since the procedure of the discrete time is similar to continuous time approach, it is summarized as follows[46]:

1. Performance index:

$$J = \sum_{k=1}^{\infty} x_k^T M x_k + u_k^T N u_k \tag{4.12}$$

2. Optimal state feedback:

$$u_k = -(N + B^T S B)^{-1} B^T S A x_k \tag{4.13}$$

3. Riccati equation for state feedback:

$$S = A^T (S - S B N^{-1} B^T S) A + Q \tag{4.14}$$

4. Optimal state estimation:

$$\hat{x}_{k+1} = A\hat{x}_k + B u_k + A L C^T (C L C^T + R)^{-1} (y_k - c\hat{x}_k) \tag{4.15}$$

5. Riccati equation for state estimation:

$$L = A[L - L C^T (C L C^T + R)^{-1} C L] A^T + Q \tag{4.16}$$

4.4 Kalman Filter

The technique of Kalman filters can be applied to problems such as optimal estimation, prediction, noise filtering, and stochastic optimal control. A Kalman filter can be easily programmed on a digital computer, and can also be applied to stationary and non-stationary processes. The discrete time Kalman filter with stationary, white process and measurement noises which are uncorrelated to each other can be summarized as follows[47]:

a. Assumptions:

$$E[w_k] = 0,\ E[w_k w_j^T] = Q\delta(k-j)$$

$$E[v_k] = 0,\ E[v_k v_j^T] = R\delta(k-j)$$

$$E[x_0] = \hat{x}_0,\ E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] = P_0$$

b. Prediction:

$$\hat{x}_k^- = A\hat{x}_{k-1}^+ + B u_{k-1} \tag{4.17}$$

$$P_k^- = A P_{k-1}^+ A^T + Q \tag{4.18}$$

c. Measurement update:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - c\hat{x}_k^-) \tag{4.19}$$

$$= (I_n - K_k C)\hat{x}_k^- + K_k y_k$$

$$P_k^+ = (I_n - K_k C)P_k^- \tag{4.20}$$

$$K_k = P_k^- C^T (C P_k^- C^T + R)^{-1} \tag{4.21}$$

where Q and R are the covariance matrices of process and measurement noises, respectively; $\hat{x}$ the estimated state vector, P the corresponding estimation error covariance matrix, I_n the n-dimensional identity matrix, K_k the Kalman filter gain and the superscript - and + distinguish the estimates before and after taking account of the current measurement data, respectively.

The inner operation of Kalman filtering can be explained as follows. Given the state, x_{k-1}, at time $k-1$ and its corresponding error covariance, P_{k-1}^+, the Kalman filter propagates the state and the error covariance to the next moment k ((4.17) and (4.18)) using the system model, and the results are x_k^- and P_k^-, respectively. This procedure is called prediction or extrapolation, because the current state is calculated based on previous data. Upon the arrival of the measurement y_k at time k, there are two sources of information about the state at time k: the propagated state with its error covariance and the new measurement with measurement noise covariance. The measurement is related to the state through measurement Equation (4.16). Using a minimum-mean-square estimation error criterion, the Kalman filter provides a method of combining these two sources of information into an optimal estimate of state x_k. This is done by adding a modifying term to the predicted value, where the modifying term is computed by premultiplying the output prediction error (the difference between the real and predicted measurements) with a weighting matrix. This weighting matrix is called the optimal Kalman filter gain, and is given by Equation (4.21). This procedure is called measurement update. After measurement update, the next prediction can be made, and so on. By this method the Kalman filter can use data recursively to yield the optimal estimated state. There is no need to keep a record of previous data.

To combine the prediction and measurement update, one can substitute (4.19) into (4.17) and (4.20) into (4.18) by changing k to $k-1$ in (4.19) and (4.20). The result in an alternative filter formulation which produces the a priori estimated state $\hat{x}_k^-$ and error covariance P_{k-1}^- as follows

$$\hat{x}_k^- = A\hat{x}_{k-1}^- + Bu_{k-1} + AK_{k-1}(y_{k-1} - C\hat{x}_{k-1}^-) \tag{4.22}$$

$$P_k^- = A(P_{k-1}^- - K_{k-1}CP_{k-1}^-)A^T + Q \tag{4.23}$$

Substituting (4.21) into (4.23) by changing k to $k-1$ in (4.21) yields

$$P_k^- = A[P_{k-1}^- - P_{k-1}^-C^T(CP_{k-1}^-C^T + R)^{-1}CP_{k-1}^-]A^T + Q \tag{4.24}$$

Comparing (4.24) and (4.16), one can have

$$\lim_{k \to \infty} P_{k-1}^- = L \tag{4.25}$$

Substituting (4.25) into (4.21) yields

$$\lim_{k \to \infty} K_{k-1} = K = LC^T(CLC^T + R)^{-1} \tag{4.26}$$

This indicates that the steady state Kalman filter gain exists and is used for state estimation in the LQG control. One can write (4.22) with the steady state Kalman filter gain as follows

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + AK(y_k - C\hat{x}_k) \tag{4.27}$$

If one defines the error between the actual output y_k and the estimated output $C\hat{x}_k$ as residual ε_k, one can have

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + AK\varepsilon_k \tag{4.28}$$

$$y_k = C\hat{x}_k + \varepsilon_k \tag{4.29}$$

In a Kalman filter sense, Equations (4.28) and (4.29) are the best description of a stochastic system whose state space model is shown in (4.10) and (4.11). The model using the prediction of (4.28) and (4.29) is called a filter-innovation model[48]. This model has been used in previous chapters. The optimal controller using a Kalman filter can be used even if there are no noise or disturbances in the system. One can identify the Kalman filter gain from the input-output experimental data without estimating the covariance of the process and measurement noises and solving the Riccati equation.

4.5 Iterative LQG Controller Design

In contrast to most existing LQG control designs which solve two separate, but dual problems: the state feedback and state estimation design. Also, an iterative controller design[49] through closed-loop identification is introduced here. In the iterative LQG control design, the Kalman filter gain and open-loop state space model can be obtained simultaneously through closed-loop identification, except one needs to solve the state feedback design which is based on the identified open-loop model. The performance index for state feedback design is

$$J = \sum_{k=1}^{\infty} y_k^T M y_k + u_k^T N u_k = \sum_{k=1}^{\infty} x_k^T C^T M C x_k + u_k^T N u_k \tag{4.30}$$

where weighting matrices M and N are design parameters. The following summarizes the procedure of the iterative LQG controller design:

Step 1:Use the priori open-loop model and arbitrary covariance matrices of the process and measurement noise to design the state feedback and Kalman filter, and calculate the controller Markov parameters. The weighting matrices M and N for the state feedback will remain the same in the following iterations.

Step 2:Excite the closed-loop system with random input and record the input and output data.

Step 3:Perform the closed-loop system identification presented in section 2.5.2 to obtain the identified open-loop system matrices, $\hat{A}$, $\hat{B}$, $\hat{C}$ and the Kalman filter gain $\hat{K}$.

Step 4:Obtain the state feedback gain F by solving the corresponding Riccati equation based on the identified open-loop model.

Step 5:Form the updated LQG controller in Equations (2.32) and (2.33) by using $A_d = \hat{A} - \hat{B}F - \hat{A}\hat{K}\hat{C}$, $B_d = \hat{A}\hat{K}$, $C_d = -F$, and $D_d = 0$.

Step 6:Calculate the updated controller Markov parameters and check the convergence of the controller by

$$\zeta = \sum_{k=0}^{n} \left\| Y_d(k)_{update} - Y_d(k)_{previous} \right\| \tag{4.31}$$

If ζ is greater than a desired value, go back to Step 2, otherwise stop.

CHAPTER V

ADAPTIVE CONTROL DESIGN

5.1 Introduction

The convention active control approach for dynamical systems could be classified by four key steps, namely, system modeling, system identification validation, controller, and finally the resulting system testing. For systems with unknown disturbances and considerable uncertainties, the traditional approach is not quick enough to catch up with the systems changes. As a result, on-line system identification computation and adaptive controller design in real time become important in these situations.

Because of the variation of the system parameters with time, a parameter estimation algorithm which is able to track the time-varying parameters is needed. Many adaptive control techniques[10-17] have been developed for this issue. There are two schemes that are particularly interesting: model reference adaptive control (MRAC)[11,50] and self-tuning regulator (STR)[51]. The basic concept for model reference adaptive system is that of giving a desired reference model and letting the system behave like it. One needs to be concerned the stability of the overall system. The basic idea of a self-tuning regulator is to use a parameter estimation with a minimum variance controller. In reference 17 and 52, Juang proposed a deadbeat controller design using ARX model with direct transmission term. Here, a deadbeat controller design without the direct transmission term is shown and recursive formulation is also provided.

In section 5.2, a multi-step ahead output prediction is derived. One can derive multi- step ahead output predictior from the basic one-step ahead prediction by using the ARX model. Through the recursive formulation, one can calculate the multi-step ahead coefficient parameters. And it turns out that one sequence of the parameters is the system Markov parameters which can be used for identifying system matrices, and the other is the observer gain Markov parameters which can be realized to have a state estimation observer. Section 5.3 shows how to design an indirect predictive control to let the system output go to zero (deadbeat). Using the algorithm derived in section 5.2 for multi-step output prediction, one could set up different control steps to design a deadbeat controller to bring the system output to zero. In this method, one needs to estimate the ARX coefficient parameters first, and use them to design a controller. The step-by-step computation is also provided. In section 5.4, a classical recursive least-squares formulation is presented. This technique combines system identification and predictive control design together. One needn't obtain the estimated ARX coefficient parameters first as one did in section 5.3. The control parameters are updated in every sampling period. There is no matrix inverse involved, and one can use this method for on-line application in real time.

5.2 Multi-Step Ahead Output Predictor

The input and output relation could be described by the ARX model where AR refers to AutoRegressive part and X refers to the eXogeneous part.

$$\begin{aligned} y_k &= \sum_{i=1}^{q} a_i y_{k-i} + \sum_{i=1}^{q} b_i u_{k-i} \\ &= a_1 y_{k-1} + a_2 y_{k-2} + \ldots + a_q y_{k-q} \\ &\quad + b_1 u_{k-1} + b_2 u_{k-2} + \ldots + b_q u_{k-q} \end{aligned} \tag{5.1}$$

where $u \in R^{ni \times 1}$ is the signal input vector, and $y \in R^{no \times 1}$ is the output vector. Note that $a_i \in R^{no \times no}$, $b_i \in R^{no \times ni}$, $i = 1, 2, \ldots, q$ are the ARX coefficient parameters, and q is the ARX model order.

By changing the time step from k to $k+1$, Equation (5.1) can be rewritten as

$$\begin{aligned} y_{k+1} &= a_1 y_k + a_2 y_{k-1} + \ldots + a_q y_{k-q+1} \\ &\quad + b_1 u_k + b_2 u_{k-1} + \ldots + b_q u_{k-q+1} \end{aligned} \tag{5.2}$$

Replace y_k in Equation (5.2) by Equation (5.1), one obtain

$$y_{k+1} = (a_1a_1+a_2)y_{k-1}+(a_1a_2+a_3)y_{k-2}+\ldots+(a_1a_{q-1}+a_q)y_{k-q+1}+a_1a_qy_{k-q}$$
$$+b_1u_k+(a_1b_1+b_2)u_{k-1}+(a_1b_2+b_3)u_{k-2}+\ldots+(a_1b_{q-1}+b_q)u_{k-q+1}+a_1b_qu_{k-q} \tag{5.3}$$

In Equation (5.3), the output at time step $k+1$ is described by the past outputs from $k-1$ to $k-q$ time steps instead of k to $k+q-1$, and the past inputs from k to $k-q$ time steps instead of k to $k-q+1$ time steps.

Define the following symbols

$$\begin{aligned}
a_1^{(1)} &= a_1a_1+a_2,\ b_1^{(1)} = a_1b_1+b_2\\
a_2^{(1)} &= a_1a_2+a_3,\ b_2^{(1)} = a_1b_2+b_3\\
&\ldots \qquad\qquad \ldots\\
a_{q-1}^{(1)} &= a_1a_{q-1}+a_q,\ b_{q-1}^{(1)} = a_1b_{q-1}+b_q\\
a_q^{(1)} &= a_1a_q \qquad\quad ,\ b_q^{(1)} = a_1b_q
\end{aligned} \tag{5.4}$$

and

$$b_0^{(1)} = b_1$$

Then one can write Equation (5.3) as follows

$$\begin{aligned}
y_{k+1} &= a_1^{(1)}y_{k-1}+a_2^{(1)}y_{k-2}+\ldots+a_q^{(1)}y_{k-q}\\
&+b_0^{(1)}u_k+b_1^{(1)}u_{k-1}+b_2^{(1)}u_{k-2}+\ldots+b_q^{(1)}u_{k-q}
\end{aligned} \tag{5.5}$$

By similar way, one could conduct the output at time step $k+j$ as

$$\begin{aligned}
y_{k+j} &= a_1^{(j)}y_{k-1}+a_2^{(j)}y_{k-2}+\ldots+a_q^{(j)}y_{k-q}\\
&+b_0^{(1)}u_{k+j-1}+\ldots+b_0^{(j)}u_k\\
&+b_1^{(j)}u_{k-1}+b_2^{(j)}u_{k-2}+\ldots+b_q^{(j)}u_{k-q}
\end{aligned} \tag{5.6}$$

where

$$
\begin{aligned}
a_1^{(j)} &= a_1^{(j-1)}a_1 + a_2^{(j-1)},\ b_1^{(j)} = a_1^{(j-1)}b_1 + b_2^{(j-1)} \\
a_2^{(j)} &= a_1^{(j-1)}a_2 + a_3^{(j-1)},\ b_2^{(j)} = a_1^{(j-1)}b_2 + b_3^{(j-1)} \\
&\qquad \cdots \qquad\qquad\qquad \cdots \\
a_{q-1}^{(j)} &= a_1^{(j-1)}a_{q-1} + a_q^{(j-1)},\ b_{q-1}^{(j)} = a_1^{(j-1)}b_{q-1} + b_q^{(j-1)}
\end{aligned}
\tag{5.7}
$$

and

$$b_0^{(j)} = b_1^{(j-1)} \tag{5.8}$$

Note that $a_i^{(0)} = a_i$ and $b_i^{(0)} = b_i$, for $i = 1, 2, \ldots$.

With some mathematical operation, Equation (5.8) could be expressed as

$$b_0^{(k)} = b_k + \sum_{i=1}^{k} a_i b_0^{(k-i)},\ k = 1, 2, \ldots, q \tag{5.9}$$

Note $b_0^{(0)} = 0$.

In a similar way, $a_1^{(j)} = a_1^{(j-1)}a_1 + a_2^{(j-1)}$ can also be described as

$$a_1^{(k-1)} = a_k + \sum_{i=1}^{k-1} a_i a_1^{k-i-1},\ k = 1, 2, \ldots, q \tag{5.10}$$

Note that $a_1^{(0)} = a_1$.

Compare Equation (5.9) and Equation (2.22) and one can see that actually $b_0^{(k)}$ is equal to system Markov parameters. On the other hand, Equation (5.10) is the same as Equation (2.23), and $a_1^{(k-1)}$ is the observer gain Markov parameters which can be realized to have a state estimation observer.

5.3 Deadbeat Control Design

In this section, it shows to design a predictive control to let the system output go to zero (deadbeat). Equation (5.6) can produce the following matrix equation, for $j = 0, 1, 2, \ldots, s, s-1, \ldots, l-1$.

$$y_k^l = Tu_k^l + \beta u_{k-q}^q + \alpha y_{k-q}^q \tag{5.11}$$

$$y_k^l = \begin{bmatrix} y_k \\ y_{k+1} \\ \dots \\ y_{k+s} \\ y_{k+s+1} \\ \dots \\ y_{k+l-1} \end{bmatrix}, \quad u_k^l = \begin{bmatrix} u_k \\ u_{k+1} \\ \dots \\ u_{k+s} \\ u_{k+s+1} \\ \dots \\ u_{k+l-1} \end{bmatrix}$$

$$y_{k-q}^q = \begin{bmatrix} y_{k-q} \\ y_{k-q+1} \\ \dots \\ y_{k-1} \end{bmatrix}, \quad u_{k-q}^q = \begin{bmatrix} u_{k-q} \\ u_{k-q+1} \\ \dots \\ u_{k-1} \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & & & & & & \\ b_0^{(1)} & 0 & & & & & \\ \dots & \dots & 0 & & & & \\ b_0^{(s)} & b_0^{(s-1)} & \dots & 0 & & & \\ b_0^{(s+1)} & b_0^{(s)} & \dots & b_0^{(1)} & 0 & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_0^{(l-1)} & b_0^{(l-2)} & \dots & b_0^{(l-s-1)} & b_0^{(l-s-2)} & \dots & 0 \end{bmatrix}$$

$$\beta = \begin{bmatrix} b_q & b_{q-1} & \dots & b_1 \\ b_q^{(1)} & b_{q-1} & \dots & b_1^{(1)} \\ \dots & \dots & \dots & \dots \\ b_q^{(s)} & b_{q-1}^{(s)} & \dots & b_1^{(s)} \\ b_q^{(s+1)} & b_{q-1}^{(s+1)} & \dots & b_1^{(s+1)} \\ \dots & \dots & \dots & \dots \\ b_q^{(l-1)} & b_{q-1}^{(l-1)} & \dots & b_1^{(l-1)} \end{bmatrix}, \quad \alpha = \begin{bmatrix} a_q & a_{q-1} & \dots & a_1 \\ a_q^{(1)} & a_{q-1} & \dots & a_1^{(1)} \\ \dots & \dots & \dots & \dots \\ a_q^{(s)} & a_{q-1}^{(s)} & \dots & a_1^{(s)} \\ a_q^{(s+1)} & a_{q-1}^{(s+1)} & \dots & a_1^{(s+1)} \\ \dots & \dots & \dots & \dots \\ a_q^{(l-1)} & a_{q-1}^{(l-1)} & \dots & a_1^{(l-1)} \end{bmatrix}$$

The quantity y_k^l and u_k^l contain a total of l output and input data points from the time step k to $k+l-1$, respectively. However, y_{k-q}^q and u_{k-q}^q only include q data points from the time step $k-q$ to $k-1$. The matrix T is normally called the Toeplitz matrix which is formed from the system Markov parameters $(b_0^{(1)}, \ldots, b_0^{(l-1)})$. In Equation (5.11) y_k^l is composed of three terms, the 2nd and 3rd terms are the known past input and output vectors from the time step $k-q$ to $k-1$, respectively. The first term, the future input vector, could be designed for deadbeat (zero) feedback control.

If one wants to design a set of future control signals, say, from u_k to u_{k+s-1}, to make the future output sequence $y_{k+s}, y_{k+s+1}, \ldots, \infty$ equal to zero, one needs to design predictive control to satisfy the condition. We assume the control operation only from u_k to u_{k+s-1}, beyond and including u_{k+s} are all zero. The system is open loop before time k and there are s steps of control action which are from the time step k to $k+s-1$. Hence, one can write the following equation according to Equation (5.11) as

$$y_{k+s}^q = T'u_k^s + \beta' u_{k-q}^q + \alpha' y_{k-q}^q \tag{5.12}$$

where

$$y_{k+s}^q = \begin{bmatrix} y_{k+s} \\ y_{k+s+1} \\ \cdots \\ y_{k+s+q-1} \end{bmatrix}, \quad u_k^s = \begin{bmatrix} u_k \\ u_{k+1} \\ \cdots \\ u_{k+s-1} \end{bmatrix}$$

and

$$T' = T(sn_0+1: sn_0+qn_0, 1:sn_i) = \begin{bmatrix} b_0^{(s)} & b_0^{(s-1)} & \cdots & b_0^{(1)} \\ b_0^{(s+1)} & b_0^{(s)} & \cdots & b_0^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ b_0^{(s+q-1)} & b_0^{(s+q-2)} & \cdots & b_0^{(q)} \end{bmatrix}$$

$$\beta' = \beta(sn_0 + 1 : qn_0 + sn_0, :) = \begin{bmatrix} b_q^{(s)} & b_{q-1}^{(s)} & \dots & b_1^{(s)} \\ b_q^{(s+1)} & b_{q-1}^{(s+1)} & \dots & b_1^{(s+1)} \\ \dots & \dots & \dots & \dots \\ b_q^{(s+q-1)} & b_{q-1}^{(s+q-1)} & \dots & b_1^{(s+q-1)} \end{bmatrix}$$

$$\alpha' = \alpha(sn_0 + 1 : qn_0 + sn_0, :) = \begin{bmatrix} a_q^{(s)} & a_{q-1}^{(s)} & \dots & a_1^{(s)} \\ a_q^{(s+1)} & a_{q-1}^{(s+1)} & \dots & a_1^{(s+1)} \\ \dots & \dots & \dots & \dots \\ a_q^{(s+q-1)} & a_{q-1}^{(s+q-1)} & \dots & a_1^{(s+q-1)} \end{bmatrix}$$

Equation (5.12) is a reduced version of (5.11) by cutting its first s equations and the equations beyond $s+q-1$. The matrix $T \in R^{qno \times sni}$ is formed by system Markov parameters, $b_0^{(k)}$. Note that ni and no are the number of inputs and outputs, respectively. q is the ARX model order and s is the number of control steps.

In Equation (5.12) we choose the number of rows for T' to be qn_0. There is a reason for this. If one flips the columns in the left/right direction and preserves the rows of T', it becomes a Hankel matrix of the pulse response. It is known that the Hankel matrix has a maximum rank of n which is the order of the system if $qn_0 \geq n$. By choosing any number greater than qn_0 could not increase the rank of T'. And one also needs to choose $sn_i \geq n$ in order to make sure that T' has rank of n.

To make y_{k+s}^q equal to deadbeat (zero), i.e.,

$$y_{k+s}^q = \begin{bmatrix} y_{k+s} \\ y_{k+s+1} \\ \dots \\ y_{k+s+q-1} \end{bmatrix} = 0$$

the feedback control u_k^s will be

$$u_k^s = -\left[T'\right]^+\left[\beta' u_{k-q}^q + \alpha' y_{k-q}^q\right] \tag{5.13}$$

u_k could be obtained from the first n_i rows of Equation (5.13) as

$$\begin{aligned} u_k &= -first\ n_i\ rows\ of\{[\ T']^+[\beta' u_{k-q}^q + \alpha' y_{k-q}^q]\} \\ &= a_1{}^* y_{k-1} + a_2{}^* y_{k-2} + \ldots + a_q{}^* y_{-q} \\ &+ b_1{}^* u_{k-1} + b_2{}^* u_{k-2} + \ldots + b_q{}^* u_{k-q} \end{aligned} \tag{5.14}$$

where $a_1{}^*, \ldots, a_q{}^*$ and $b_1{}^*, \ldots, b_q{}^*$ are the control parameters to be computed for the feedback control u_k. In theory, the control action will bring the system to rest (zero) after the time step $k + s$. Since the system has input and output uncertainties, however, the control action can only bring the system output down to the level of the uncertainties.

5.3.1 Computation Steps

1. Compute the estimated ARX coefficient parameters, a_i and b_i, $i = 1, 2, \ldots, q$ from given input and output data.
2. Use Equation (5.8) to form the Hankel-like matrix T'. Note that the number of control steps s should be chosen properly to make sure the rank of T' is n or qn_o whichever is the least, where n is the order of the system, q is the order of the ARX model, and n_o is the number of the outputs.
3. Form matrices β' and α' by using Equation (5.7).
4. Calculate the feedback control parameters $a_1{}^*, \ldots, a_q{}^*$ and $b_1{}^*, \ldots, b_q{}^*$ from Equation (5.13).

5.4 Recursive Formula for Real-Time Application

In this section, a classical least-squares recursive formulation will be shown and it satisfies both system identification and deadbeat controller design requirements. The recursive formulation can be used for on-line operation when real time application is required. According to Equation (5.11), one can write down

$$y_{k+s}^q = T'' u_k^{q+s} + \beta' u_{k-q}^q + \alpha' y_{k-q}^q \tag{5.15}$$

where

$$y_{k+s}^{q} = \begin{bmatrix} y_{k+s} \\ y_{k+s+1} \\ \dots \\ y_{k+s+q-1} \end{bmatrix}, u_{k}^{q+s} = \begin{bmatrix} u_{k} \\ u_{k+1} \\ \dots \\ u_{k+s+q-1} \end{bmatrix}$$

$$T'' = \begin{bmatrix} b_0^{(s)} & b_0^{(s-1)} & \dots & b_0^{(1)} & 0 & 0 & \dots & 0 \\ b_0^{(s+1)} & b_0^{(s)} & \dots & b_0^{(2)} & b_0^{(1)} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ b_0^{(s+q-1)} & b_0^{(s+q-2)} & \dots & b_0^{(q)} & b_0^{(q-1)} & b_0^{(q-2)} & \dots & 0 \end{bmatrix}$$

and β', α', u_{k-q}^{q}, y_{k-q}^{q} are the same as in Equation (5.12). Note (5.15) is a slight difference with (5.12) in which we assume the control operation only from u_k to u_{k+s-1}, beyond and including u_{k+s} are all zero; hence, we use $T''u_k^{q+s}$ instead of $T'u_k^{s}$. Later on it will show the same result for feedback control signal by using s steps of control action. Actually, Equation (5.15) must be satisfied for any given input and output sequence, and one can calculate the matrices T'', β' and α'. Examining the matrix T'', one can see T' is a submatrix of T'', i.e., $T' = T''(:, 1:sn_i)$. Let's rewrite Equation (5.15) as following

$$y_{k+s}^{q} = T'u_k^{s} + T_o u_{k+s}^{q} + \beta' u_{k-q}^{q} + \alpha' y_{k-q}^{q} \tag{5.16}$$

where

$$T_o = \begin{bmatrix} 0 & & & \\ b_0^{(1)} & 0 & & \\ \dots & \dots & \dots & \dots \\ b_0^{(q-1)} & b_0^{(q-2)} & \dots & 0 \end{bmatrix}, u_{k+s}^{q} = \begin{bmatrix} u_{k+s} \\ u_{k+s+1} \\ \dots \\ u_{k+s+q-1} \end{bmatrix}$$

One can solve u_k^{s} by following

$$u_k^{s} = (T')^{+}\left[y_{k+s}^{q} - T_o u_{k+s}^{q} - \beta' u_{k-q}^{q} - \alpha' y_{k-q}^{q}\right]$$

or in a simply matrix form

$$u_k^s = \left[-(T')^+\alpha' \; -(T')^+\beta' \; (T')^+ \; -(T')^+T_o\right]\begin{bmatrix} y_{k-q}^q \\ u_{k-q}^q \\ y_{k+s}^q \\ u_{k+s}^q \end{bmatrix} \tag{5.17}$$

Define the following quantities

$$E_c = \left[-(T')^+\alpha' \; -(T')^+\beta'\right], \; E_o = \left[(T')^+ \; -(T')^+T_o\right]$$

and

$$v_{k-q}^q = \begin{bmatrix} y_{k-q}^q \\ u_{k-q}^q \end{bmatrix}, \; v_{k+s}^q = \begin{bmatrix} y_{k+s}^q \\ u_{k+s}^q \end{bmatrix}$$

Then Equation (5.17) becomes

$$u_k^s = \left[E_c \; E_o\right]\begin{bmatrix} v_{k-q}^q \\ v_{k+s}^q \end{bmatrix} \tag{5.18}$$

If one chooses the feedback control signal u_k^s, i.e., the control operation only from u_k to u_{k+s-1} to bring the output to zero, as

$$u_k^s = E_c v_{k-q}^q = \left[-(T')^+\alpha' \; -(T')^+\beta'\right]\begin{bmatrix} y_{k-q}^q \\ u_{k-q}^q \end{bmatrix} \tag{5.19}$$

which is the same as Equation(5.13), i.e., $u_k^s = -\left[T'\right]^+\left[\beta' u_{k-q}^q + \alpha' y_{k-q}^q\right]$. In order to satisfy Equation (5.18), then the following equation should be held

$$E_o v_{k+s}^q = \left[(T')^+ \; -(T')^+T_o\right]\begin{bmatrix} y_{k+s}^q \\ u_{k+s}^q \end{bmatrix} = 0 \tag{5.20}$$

Since u_{k+s}^q has been set to zero, and if T' $(sni \times qno)$ is full rank qno with $sni \geq qno$, then $y^q_{k+s} = 0$, i.e., after s time steps, the feedback control force u^s_k will bring the output to rest (zero).

And the feedback control signal u_k equals

$$u_k = \left[the\ first\ n_i\ rows\ of\ E_c\right] v^q_{k-q}$$

$$= E_{cni} v^q_{k-q} \tag{5.21}$$

One can use the input-output relationship of (5.18) to solve for E_c and E_o. In Equation (5.18)

$$u_k = \left(the\ first\ n_i\ rows\ of \left[E_c\ E_o\right]\right)\begin{bmatrix} v^q_{k-q} \\ v^q_{k+s} \end{bmatrix}$$

$$= \left[E_{cni}\ E_{oni}\right]\begin{bmatrix} v^q_{k-q} \\ v^q_{k+s} \end{bmatrix} \tag{5.22}$$

One can form the following equation to solve for E_{cni} and E_{oni} by using the application of Equation (5.22)

$$U_k = \left[E_{cni}\ E_{oni}\right]\begin{bmatrix} V^q_{k-q} \\ V^q_{k+s} \end{bmatrix} \tag{5.23}$$

where

$$U_k = \left[u_k\ u_{k+1}\ \cdots\ u_{l-q-s+1}\right]$$

$$V^q_{k-q} = \begin{bmatrix} y_{k-q} & y_{k-q+1} & \cdots & y_{l-2q-s+1} \\ u_{k-q} & u_{k-q+1} & \cdots & u_{l-2q-s+1} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ y_{k-1} & y_k & \cdots & y_{l-q-s} \\ u_{k-1} & u_k & \cdots & u_{l-q-s} \end{bmatrix},\ V^q_{k+s} = \begin{bmatrix} y_{k+s} & y_{k+s+1} & \cdots & y_{l-q+1} \\ u_{k+s} & u_{k+s+1} & \cdots & u_{l-q+1} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ y_{k+s+q-1} & y_{k+s+q} & \cdots & y_l \\ u_{k+s+q-1} & u_{k+s+q} & \cdots & u_l \end{bmatrix}$$

And l is the data length. One needs to choose l large enough to make sure the solution exists.

As a result, one can find $\left[E_{cni}\ E_{oni}\right]$ from (5.23)

$$\left[E_{cni}\ E_{oni}\right] = U_k \begin{bmatrix} V_{k-q}^q \\ V_{k+s}^q \end{bmatrix}^+ \tag{5.24}$$

5.4.1 Recursive Algorithm

The following will use the classical recursive least squares method as shown in section 2.3.2 to solve the coefficient parameters $\left[E_{cni}\ E_{oni}\right]$ for on-line operation.

Rewrite Equation (5.22) as

$$u_k = Eh_{k-1} \tag{5.25}$$

where

$$E = \left[E_{cni}\ E_{oni}\right],\ v_k = \begin{bmatrix} y_k \\ u_k \end{bmatrix},\ h_{k-1} = \begin{bmatrix} v_{k-q} \\ \cdots \\ v_{k-1} \\ v_{k+s} \\ \cdots \\ v_{k+s+q-1} \end{bmatrix} \tag{5.26}$$

Then, define the following quantities

$$M_k = h_k^T P_{k-1}\left[1 + h_k^T P_{k-1} h_k\right]^{-1} \tag{5.27}$$

$$\tilde{u}_{k+1} = \tilde{E}_k h_k \tag{5.28}$$

$$P_k = P_{k-1}\left[I - h_k M_k\right] \tag{5.29}$$

$$\tilde{E}_{k+1} = \tilde{E}_k + \left[u_{k+1} - \tilde{u}_{k+1}\right] M_k \tag{5.30}$$

Equations (5.27) to (5.30) are the basic recursive least squares formulations to identify the gain matrix E which includes E_{cni} for the deadbeat controller design and E_{cno} for the need of system identification. The initial values of P_0 and $\tilde{E}_1$ can be obtained by two approaches. One is to execute a small batch calculation after gathering a sufficient number of data. The other is to assign a number to them, one can assign P_0 and $\tilde{E}_1$ as $\kappa I_{2q(no+ni))}$ and $0_{ni \times 2q(ni+no)}$, respectively, where κ is a large positive number.

5.4.2 Computation Steps

1. Form the vector h_k as shown in Equation (5.26) with the new component $v_{k+s+q-1}$ as the last $ni+no$ rows.
2. Calculate the gain vector M_k $(1 \times 2q(ni+no))$ from Equation (5.27). Note that one should compute $h_k^T P_{k-1}$ first, then use the result to calculate $\left[h_k^T P_{k-1}\right] h_k$ in order to save the time for calculation.
3. Compute $\tilde{u}_{k+1}$ from Equation (5.28).
4. Obtain P_k from Equation (5.29) for updating P_{k-1}, and use the values of h_k and M_k obtained in 1^{st} and 2^{nd} step, respectively.
5. Update the desired coefficient parameter, $\tilde{E}_k$, to obtain $\tilde{E}_{k+1}$ from Equation (5.30) by using the new input u_{k+1}, the estimated input $\tilde{u}_{k+1}$ computed from 3^{rd} step, and the gain vector M_k computed from 2^{nd} step.

The recursive algorithm can be used for on-line application. No matrix inverse is needed in the procedure. Only updating P_k and M_k will take more time than the other quantities.

CHAPTER VI

NUMERCIAL VALIDATION

6.1 Introduction

High speed air flow over the cavity produces a complex oscillatory flowfield and induces pressure oscillations within the cavity. The large pressure fluctuations induced from the cavity generate undesirable and loud noise. This may have an adverse effect on the stability and performance of overall system. This chapter uses a set of input and output experimental data from the cavity in the subsonic flow to demonstrate the feasibility of system identification and controller design. The piezeroelectrical material works as an actuator placed on the roof of the front wall of the open cavity. There are five pieces of piezeroeletric material and each is $3'' \times 1''$. The microphone placed on the center floor of the cavity measures noise as output. The wind speed is about 60 m/s and the sampling rate is 5k Hz. The top view of the configuration is shown on Figure 6.1. Our goal is to reduce the noise, namely, let the output down to zero. For LQG control, system matrices for state space model and Kalman filter gain are identified first, then a LQG controller is designed to let the system reach the desired performance. For adaptive control, the recursive least-squares technique is used. The formulation simultaneously satisfied system identification and deadbeat controller design requirements, i.e., the system identification and deadbeat control are built into one formulation. The control parameters are calculated first and the feedback control force is formed to bring the system down to deadbeat (zero).

6.2 Numerical Results

In this section, although the simulation uses single input and output (SISO) data, it is applicable to multiple inputs and outputs (MIMO) cases. Since the system is stable, one could use the open-loop identification algorithm as shown in section 2.4 without adding the controller.

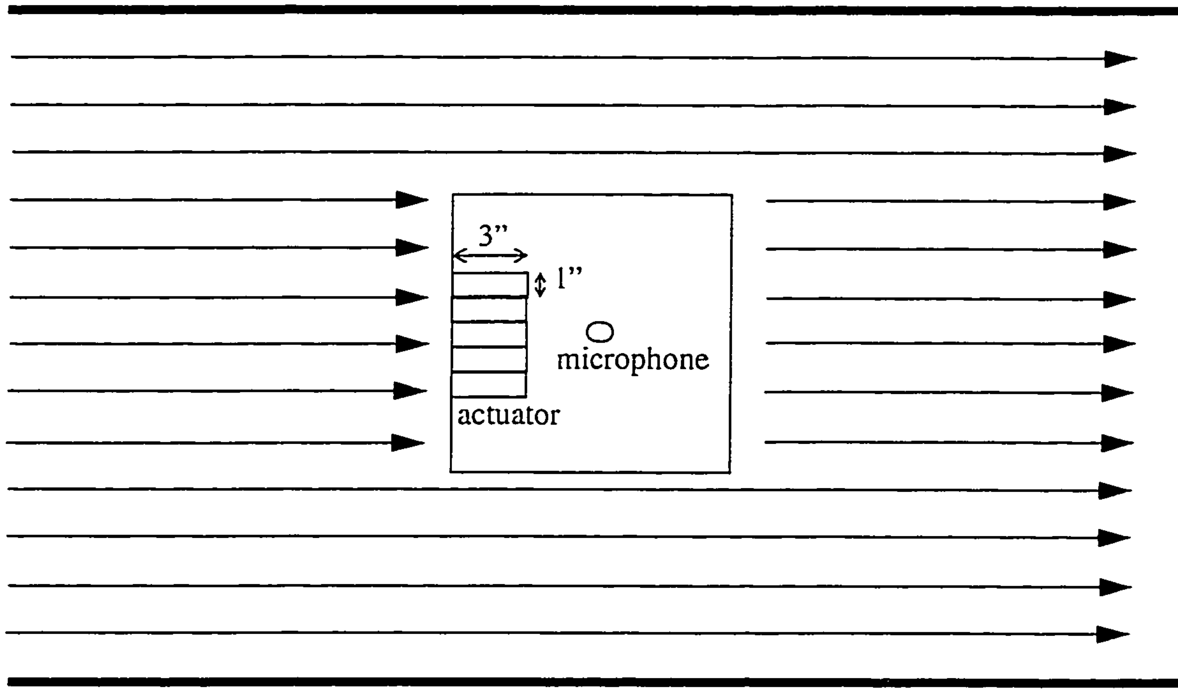


Figure 6.1 Top view of cavity noise system inside wind tunnel

6.2.1 Result for LQG Control

Figures 6.2 and 6.3 show the original input and output data, respectively. Figure 6.4 shows real and predicted output for data from 1 to 1024 points, and Figure 6.5 shows new test data and ARX predicted output from 1025 to 2048 data points. And the identified system matrices and Kalman filter gain are

$$A_i = \begin{bmatrix} 0.9710 & 0.2826 \\ -0.2826 & 0.9418 \end{bmatrix}, B_i = \begin{bmatrix} 1.3719 \\ 9.5415 \end{bmatrix}$$

$$C_i = \begin{bmatrix} 1.3719 & -9.5415 \end{bmatrix}$$

and Kalman filter gain

$$K_i = \begin{bmatrix} 0.3369 \\ -0.1520 \end{bmatrix}$$

In Figures 6.6 to 6.8, the LQG control design is performed to reduce the noise. The controller is added after 1500 data points. Figure 6.6 shows the output after adding the LQG controller. Figures 6.7 and 6.8 depict the plot for dynamic feedback and plot for actuator, respectively. By constrainting the actuator range from 0.1 to 0.25, the plots are shown in

Figures 6.9 to 6.11.

6.2.2 Result for Adaptive Control

Figures 6.12 to 6.20 show the result of using adaptive control by the recursive least-squares technique and the controller is turned on after 1500 data points. The model order is $q = 2$. From Figures 6.12 to 6.14, the number of control steps equals 2, Figure 6.12 shows the original input and the computed control force. Figure 6.13 depicts the overall new input, i.e., the original input plus the control force. Figure 6.14 shows the output after adding the control force. Figures 6.15 to 6.17 use 5 control steps, and the control force is found out to be

$$u_k = -0.3004u_{k-1} - 1.3805u_{k-2} - 0.0080y_{k-1} + 0.0145y_{k-2}$$

Figures 6.18 to 6.20 use 10 control steps.

6.3 Concluding Remarks

One can see from Figures 6.4 and 6.5 that the prediction of the output is very accurate when using ARX model structure. When constraining the actuator range, one can see the result, which is shown in Figures 6.9 to 6.11, is almost the same as the one without constrainting one. For adaptive control, if the control step is choosen to be 2 which is the minimum number of steps as mentioned in section 5.3, one can see there is a peak when the controller is turned on. One can avoid this by increasing the number of control steps. Although increasing the control steps will eliminate the undesired peak, it doesn't mean more control steps will get a better result. This is observed from Figures 6.17 and 6.20, in which the control steps are equal to 5 and 10, respectively.

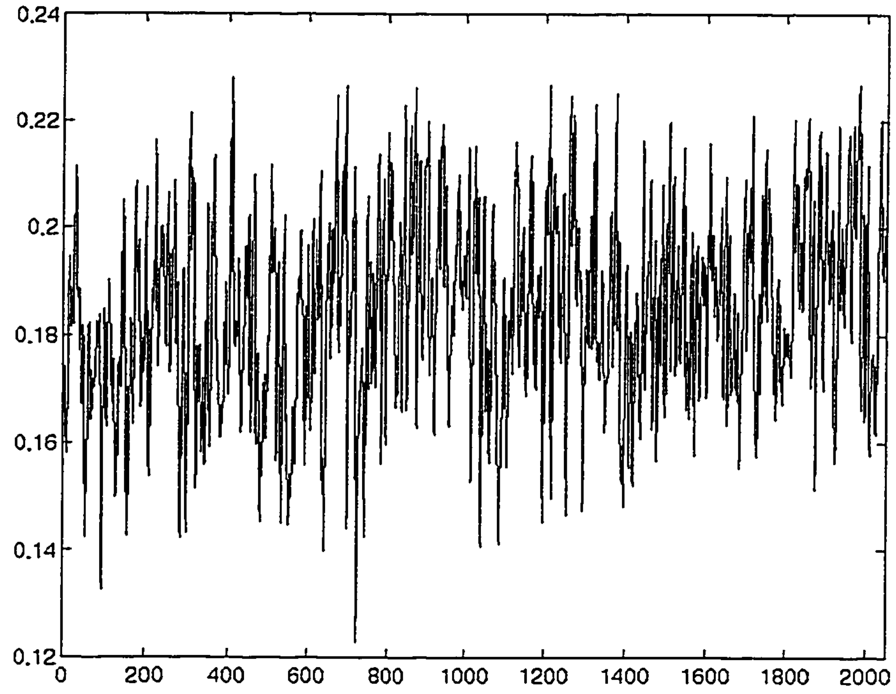


Figure 6.2 Input data

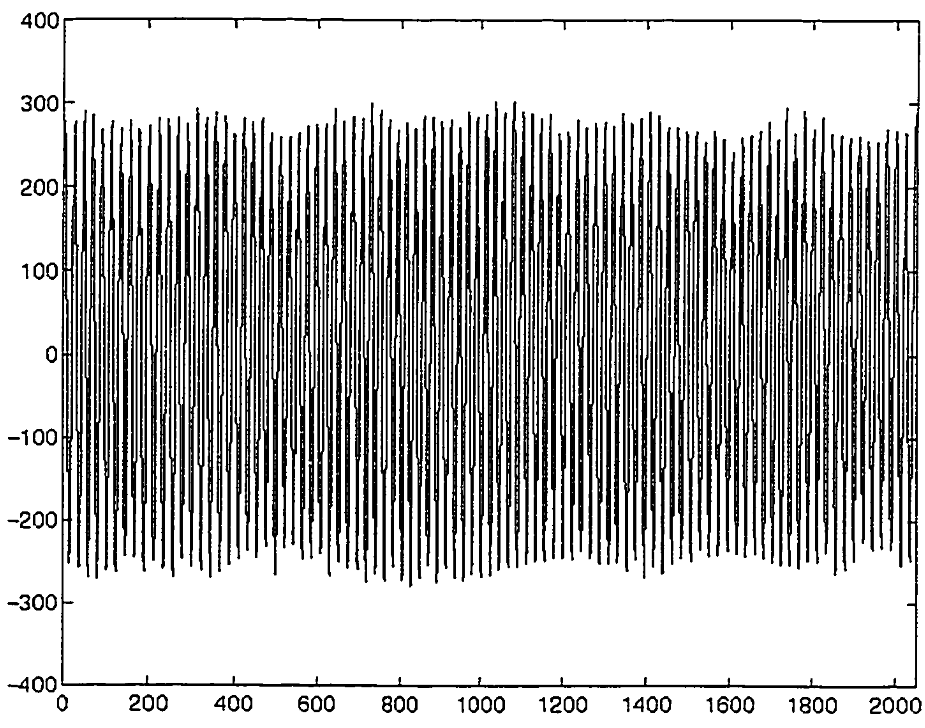


Figure 6.3 Output data

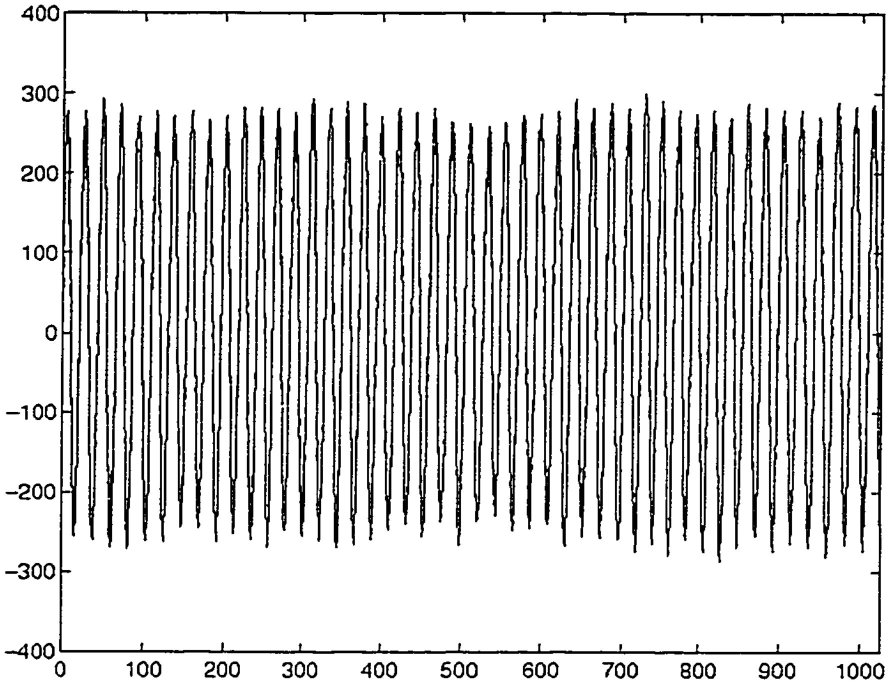


Figure 6.4 Real (solid) and predicted (dashed) output for data from 1 to 1024 points

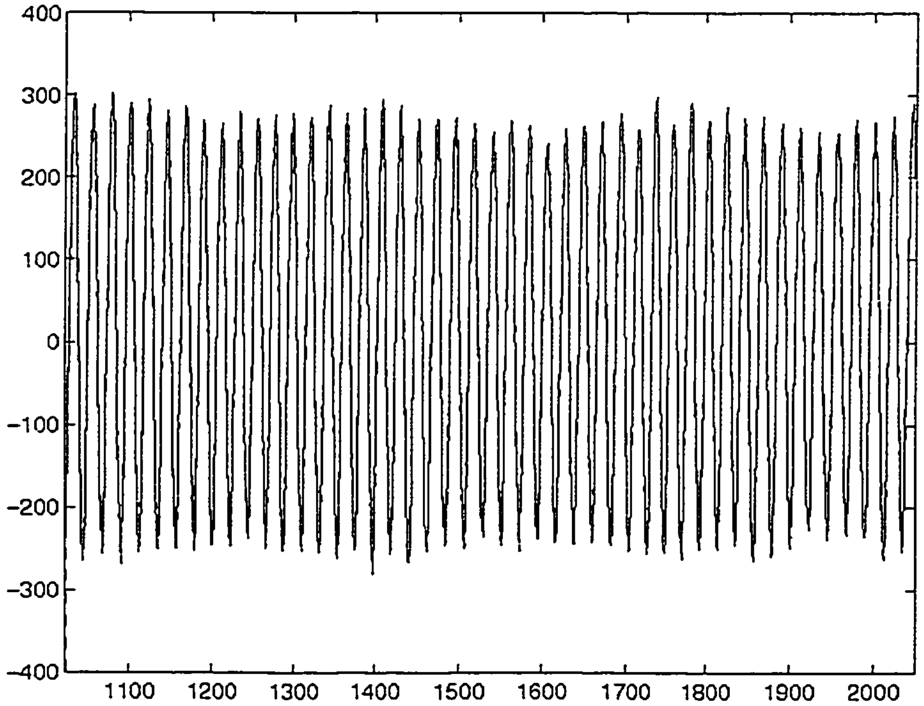


Figure 6.5 New test data (solid) and ARX predicted (dashed) output from 1025 to 2048 points

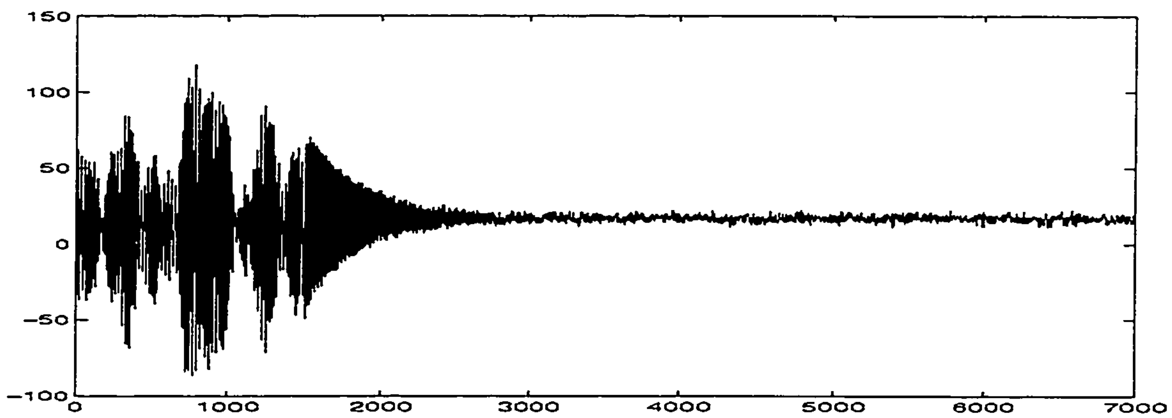


Figure 6.6 Output after adding the LQG controller (without constraint on actuator)

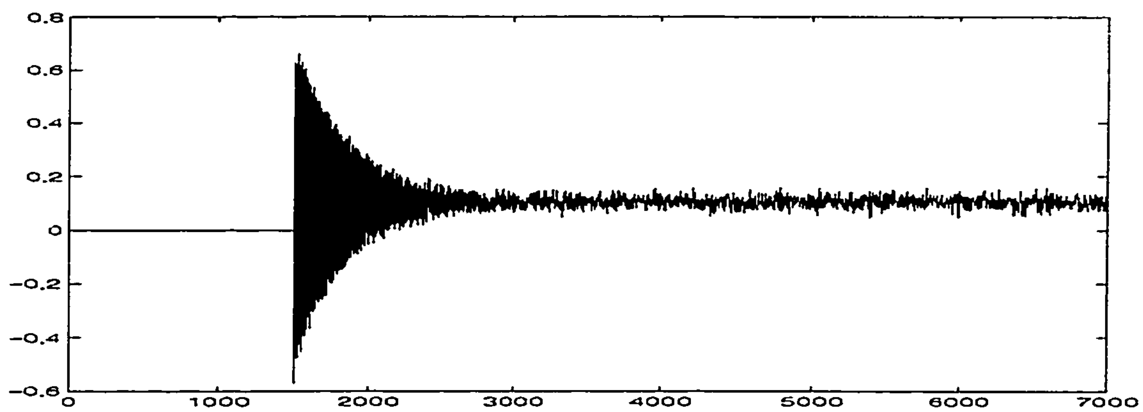


Figure 6.7 Dynamic feedback (without constraint on actuator)

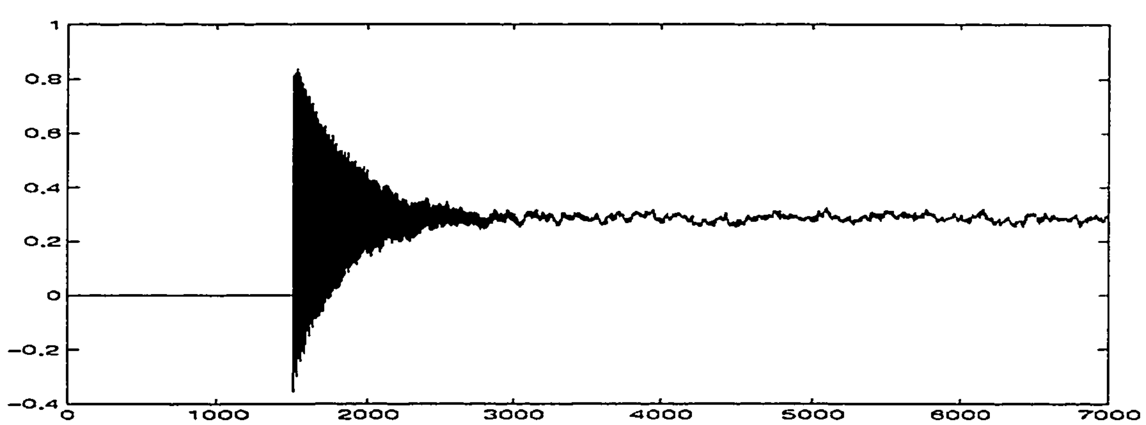


Figure 6.8 Plot for actuator (without constraint on actuator)

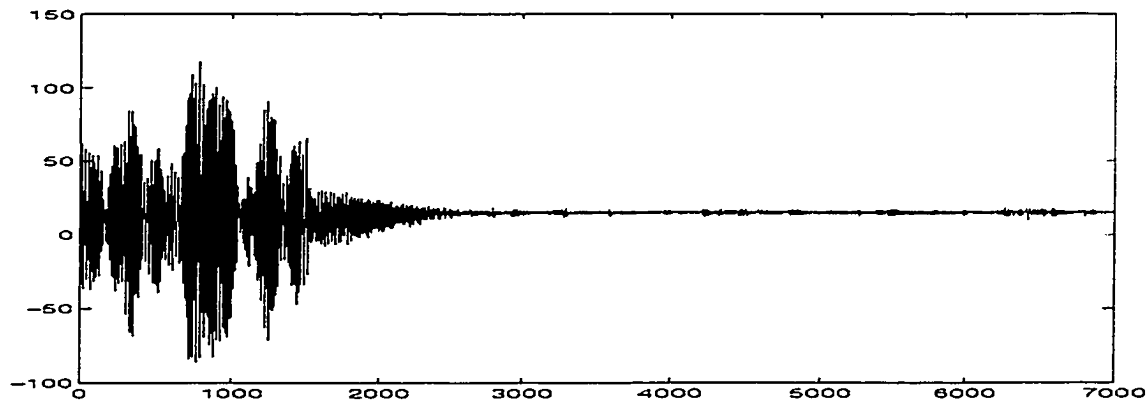


Figure 6.9 Output after adding the LQG controller (with constraint on actuator)

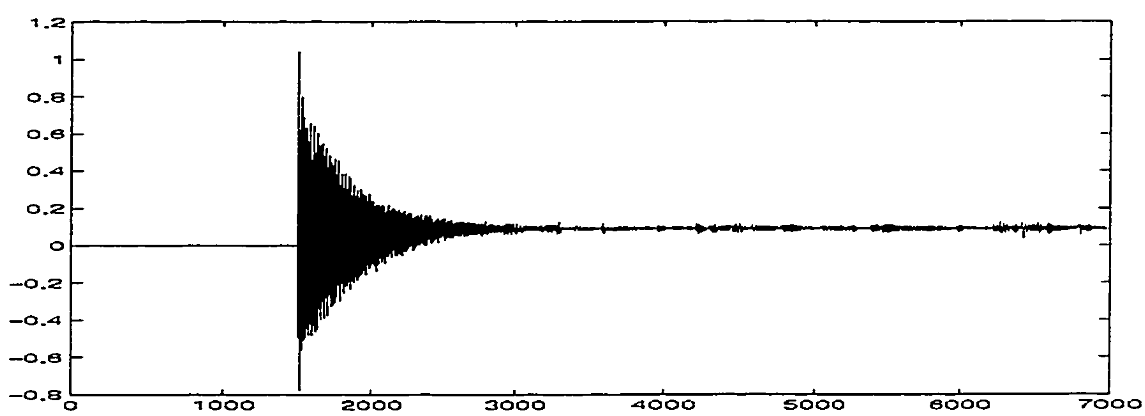


Figure 6.10 Dynamic feedback (with constraint on actuator)

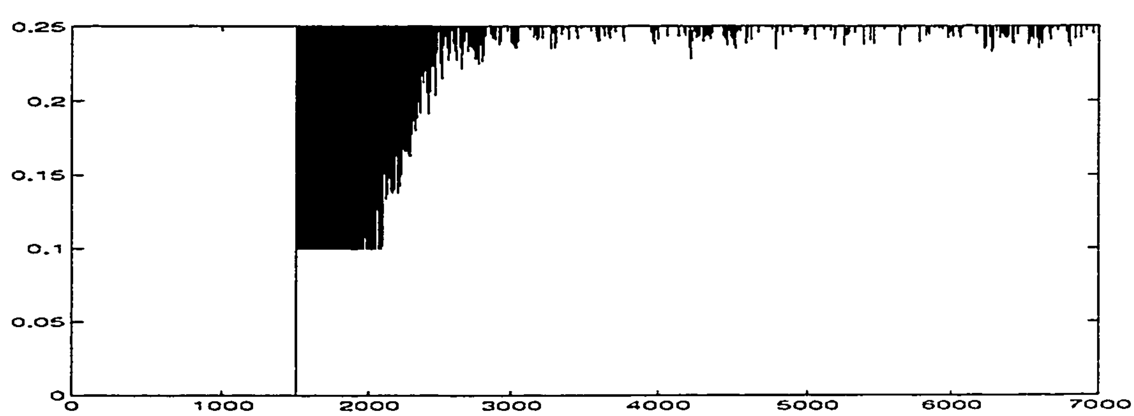


Figure 6.11 Plot for actuator (with constraint on actuator)

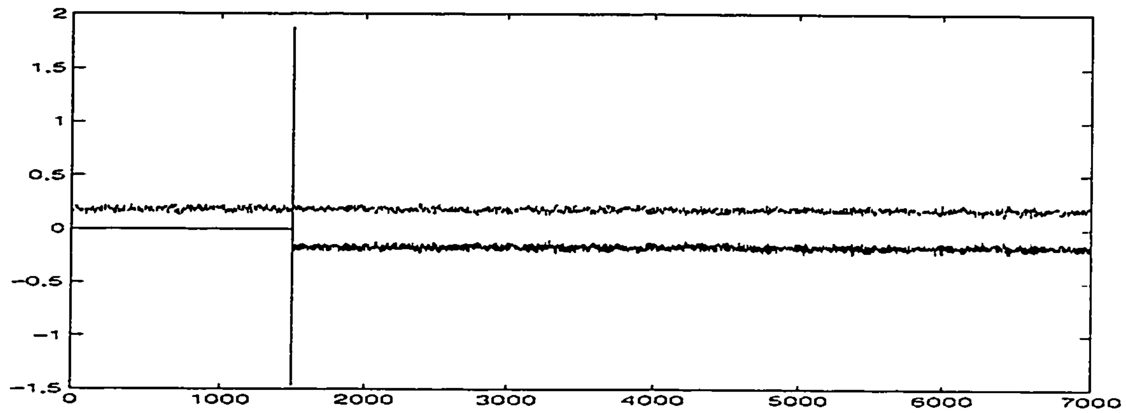


Figure 6.12 The original input (dashed) and the computed control force (solid), s=2.

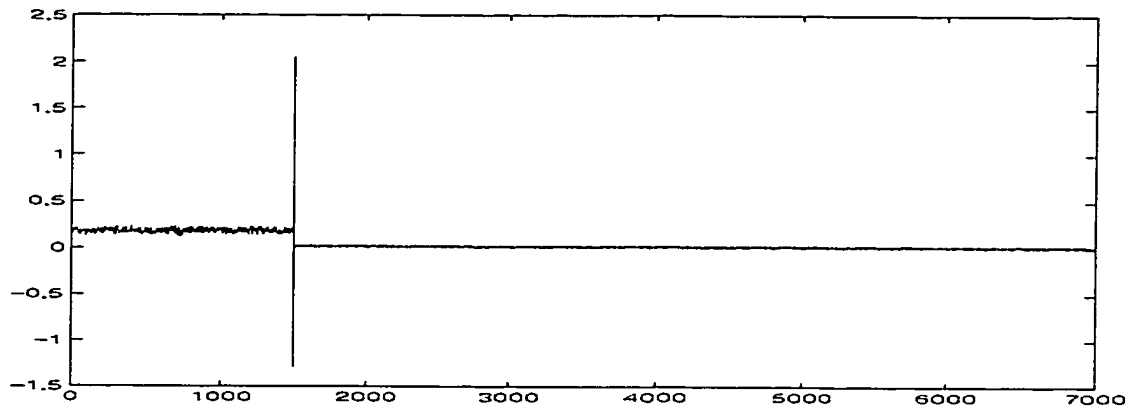


Figure 6.13 The overall new input (the original input + the control force), s=2.

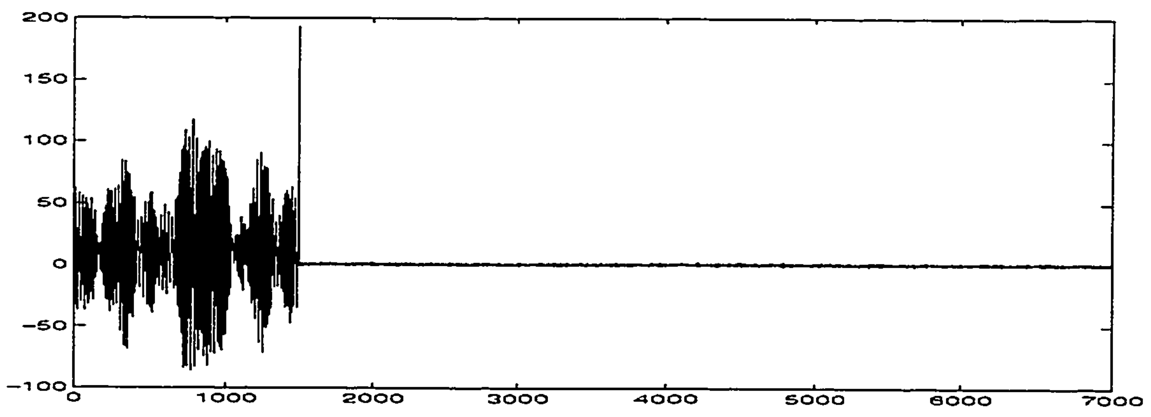


Figure 6.14 Output after adding the control force, s=2.

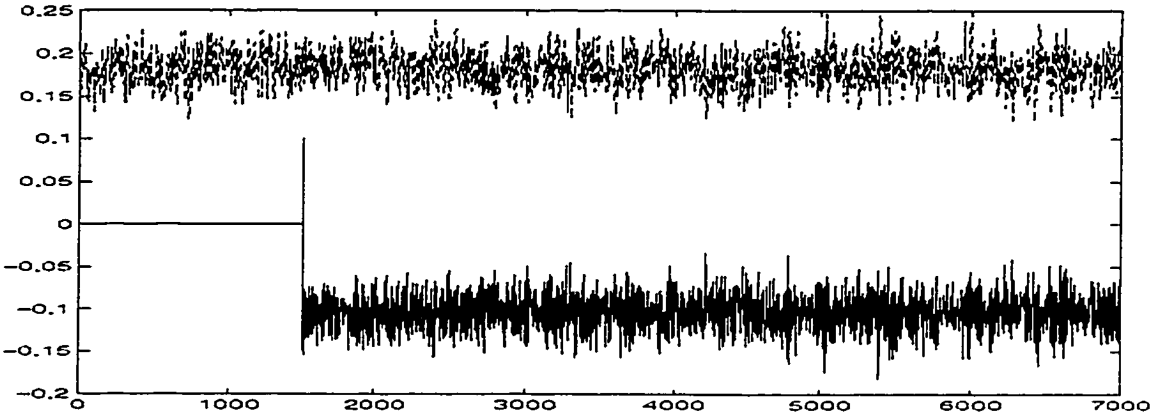


Figure 6.15 The original input (dashed) and the computed control force (solid), s=5.

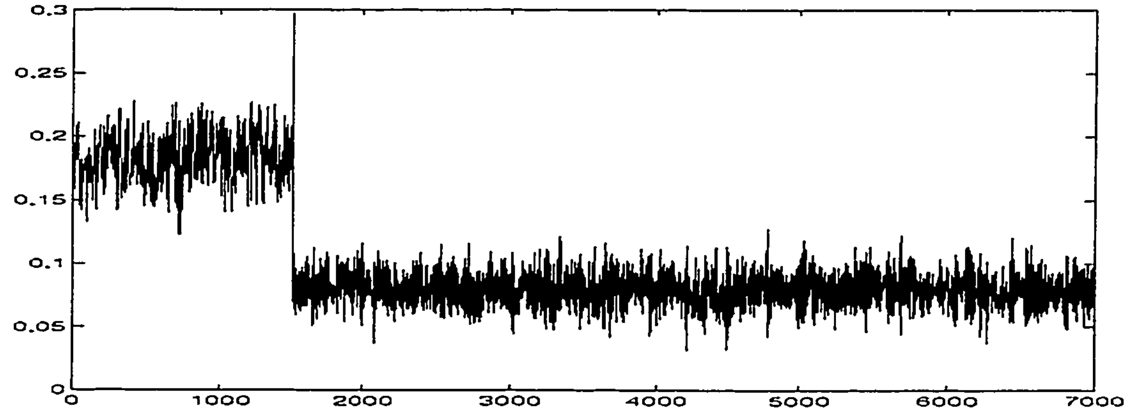


Figure 6.16 The overall new input (the original input + the control force), s=5.

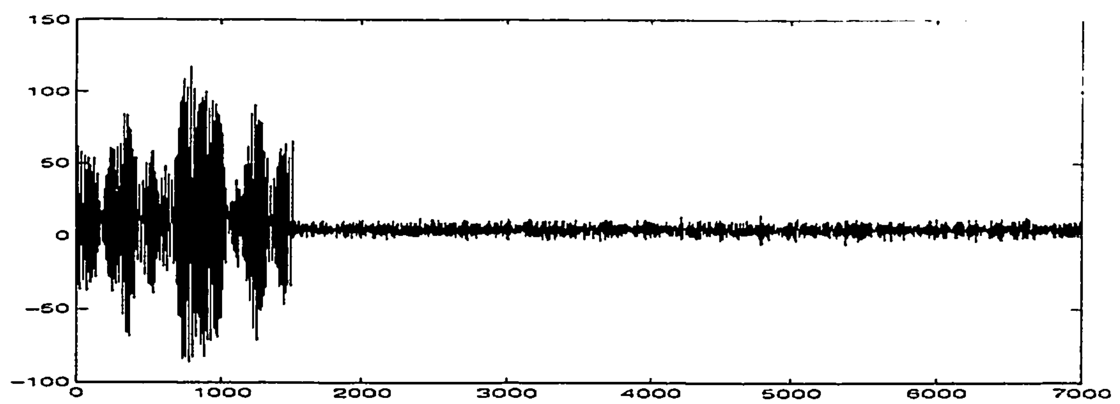


Figure 6.17 Output after adding the control force, s=5.

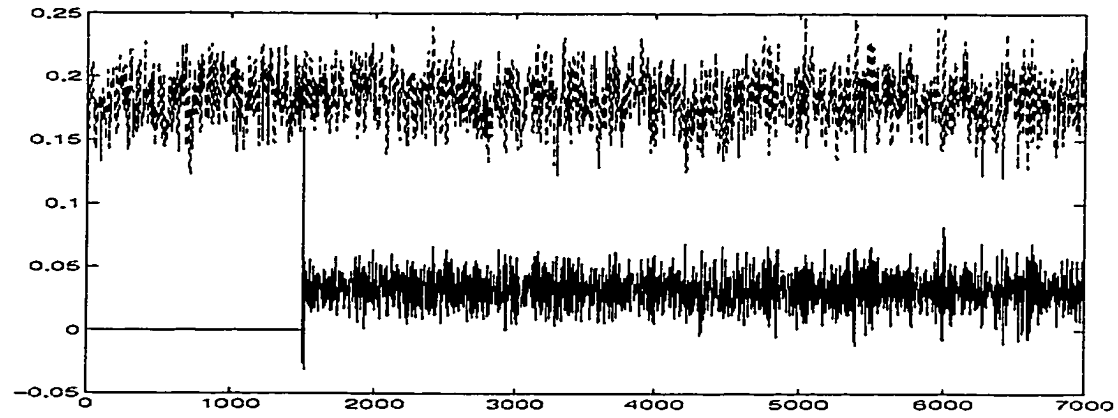


Figure 6.18 The original input (dashed) and the computed control force (solid), s=10.

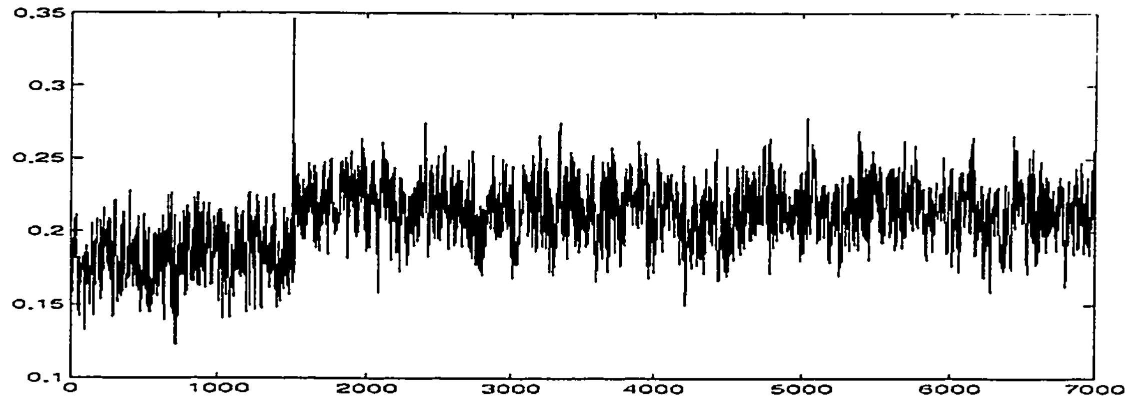


Figure 6.19 The overall new input (the original input + the control force), s=10.

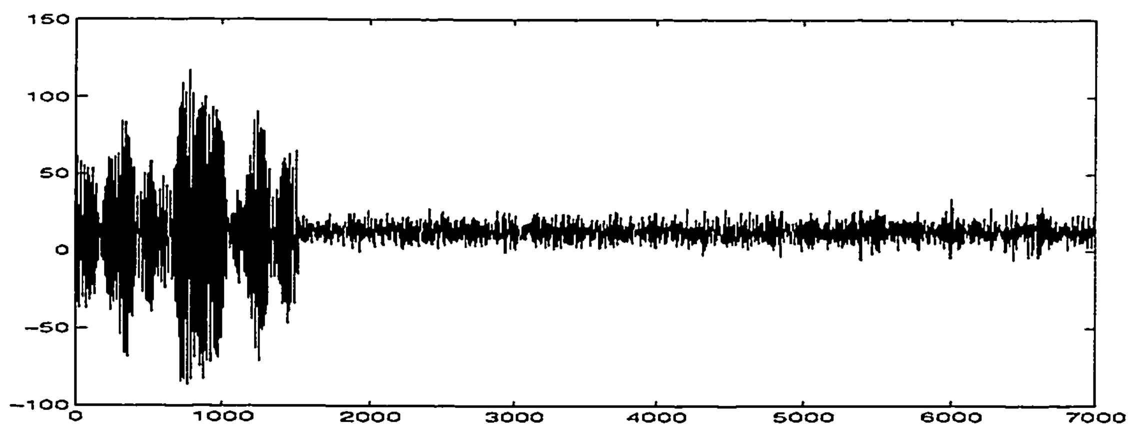


Figure 6.20 Output after adding the control force, s=10.

CHAPTER VII

CONCLUSIONS

7.1 Summary

An indirect closed-loop system identification through residual whitening has been proposed. The cavity noise system is identified and the reduction of noise is achieved with two active controllers, namely, LQG controller and deadbeat controller. The deadbeat controller includes indirect and direct algorithms.

When one identifies a system from available input and output data in time domain, the data length and the ARX model order are assumed to be long enough and sufficiently large, respectively. The identified observer tends to be optimal Kalman filter gain in the presence of process and measurement noises. In practice, however, one can't do it in this way. Hence, there are certain errors existing in the identification process. Actually, the major error is introduced by the truncation of the infinite ARX model series into a finite model.[53] In the residual whitening method, the ARMAX model which includes dynamics of noise is used instead of ARX model. Through residual whitening, the optimal properties of the Kalman filter could be enforced for a finite set of data, and the residual is minimized, orthogonal to the time-shifted versions of itself and to the given set of input-output data. The requirement of the model order could also be relaxed to reduce the computation burden, especially for several inputs and outputs systems.

When high speed air flows over a cavity, it produces a complex oscillatory flowfield and induces pressure oscillations within the cavity, and undesirable and loud noise generated. This may have an adverse effect on the objects. Hence, it's an important issue to work on noise reduction.

In the existing LQG control design, the controllers are designed by solving two separate, but dual problems: the state feedback design (LQR) and state estimation design (Kalman filter). Through system identification, one can identify the system matrices and steady state Kalman filter gain. Then the estimated states through the Kalman filter are used for state feedback. It needs trial-and-error approach to choose suitable weighting matrices for state feedback design. Hence, basically the LQG uses the output information to accomplish state feedback through state estimation.

In the deadbeat controller design, one can use either the indirect method or recursive least-squares method to design the controller through multi-step ahead output prediction. The multi-step ahead output prediction algorithms allow one to predict the future output several steps away by using the ARX coefficient parameters and set up different control steps to design a deadbeat controller which will bring the output to zero.

For the indirect method, one needs to calculate the estimated ARX coefficient parameters first, then use the deadbeat control algorithms to find out the feedback control parameters. For the classical recursive least-squares method, one can perform the adaptive control for on-line operation. Here, the system identification and deadbeat control are built into one formulation. Hence, one can simultaneously identify the system and perform deadbeat controller design. In this recursive algorithm, the control parameters are updated in every sampling period and no matrix inverse is involved.

7.2 Further Extension of the Research

One natural extension for the indirect closed-loop system identification through residual whitening is to investigate its application for recursive algorithms. In the presence of high process and measurement noises, however, the residual whitening method could not reach a significant result. Since the output is highly corrupted by noise, one can't expect only to whiten the residual to have a better result. The estimated coefficient parameters are matching with the corrupted output instead of the 'true' output. Hence, one may need to investigate another approach to ignore the influence of high noise, such as new input design.[54]

In the deadbeat control design, one may use fast versions of the recursive least-squares method, such as fast transversal filter and lattice filter,[18,48] besides the classical recursive least-squares. As the order of ARX model increases, the computation of classical recursive least-squares increases on the order of n^2 for each recursion, where n is the order of the model. However, the computation of the other two filters will only increase linearly with the filter order n. Hence, they will be more suitable for on-line operation.

REFERENCES

1 Baysal, O., Yen, G.W., and Fouladi, K., "Navier-Stokes Computations of Cavity Aeroacoustics with Suppression Devices," *Journal of Vibration and Acoustics*, Vol. 116, January 1994, pp.105-111.

2 Kim, I., and Chokani, N., "Navier-Stokes Study of Supersonic Cavity Flowfield with Passive Control," *Journal of Aircraft*, Vol. 29, No. 2, March-April 1992, pp. 217-223.

3 Tam, C.J., Orkwis, P.D., and Disimile, P.J., "Algebraic Turbulence Model Simulations of Supersonic Open-Cavity Flow Physics," *AIAA Journal*, Vol. 34, No. 11, November 1996, pp. 2255-2260.

4 Sarohia, V. and Massier, P.F., "Control of Cavity Noise," *Journal of Aircraft*, Vol. 14, No. 9, September 1977, pp. 833-837.

5 Sarno, R.L., and Franke, M.E., "Suppression of Flow-Induced Pressure Oscillations in Cavities," *Journal of Aircraft*, Vol. 31, No. 1, Jan.-Feb. 1994, pp. 90-96.

6 Jeng, Y.N., and Payne, U.J., "Numerical Study of a Supersonic Open Cavity Flow and Pressure Oscillation Control," *Journal of Aircraft*, Vol. 32, No. 2, March-April 1995, pp. 363-369.

7 Phan, M., Horta, L.G., Juang, J.-N.,and Longman, R.W., "Improvement of Observer/ Kalman Filter Identification (OKID) by Residual Whitening," *Journal of Vibrations and Acoustics*, Vol. 117, April 1995, pp. 232-239.

8 Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problem," *Trans. ASME Journal of Basic Engineering*, Vol. 82D, March, 1960, pp. 35-45.

9 Kalman, R.E., and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory," *Trans. ASME Journal of Basic Engineering*, Vol. 83, 1961, pp. 95-108.

10 Goodwin, G.C., and Sin, K.S., *Adaptive Filtering, Prediction and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1984.

11 Landau, Y.D., *Adaptive Control*, Marcel Dekker Inc., New York, 1980.

12 De Keyser, R.M.C. and Van Cauwenberghe, A.R., "A self-Tuning Multi-Step Predictor Application," *Automatica*, Vol. 17, No. 1, 1979, pp. 167-174.

13 Peterka, V., "Predictor-Based Self-Tuning Control," *Automatica*, Vol. 20, No. 1, 1984, pp. 39-50.

14 Clarke, D.W., Mohtadi, C., and Tuffs, P.S., "Generalized Predictive Control-Part I. The Basic Algorithm," *Automatica*, Vol. 23, No. 2, 1987, pp. 137-148.

15 Clarke, D.W., Mohtadi, C., and Tuffs, P.S., "Generalized Predictive Control-Part II. Extensions and Interpretations," *Automatica*, Vol. 23, No. 2, 1987, pp. 149-160.

16 Mosca, E., *Optimal, Predictive, and Adaptive Control*, Prentice-Hall, New Jersey, 1995.

17 Juang, J.-N. and Phan, M.G., "Recursive Deadbeat Controller Design," *NASA Technical Memorandum.*

18 Juang, J.-N., *Applied System Identification*, PRT Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1994.

19 Suge, A.P., and Melsa, J.L., *System Identification*, Academic Press, Inc., 1971

20 Ljung, L., *System Identification - Theory for the User*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

21 Sorenson, Harold W., *Parameter Estimation*, Marcel Dekker Inc., New York, 1980

22 Hsia, T.C., *System Identification: Least-Squares Methods*, Lexington Books, 1977.

23 Strejc, V., *Trends and Progress in System Identification*, edited by P. Eykhoff, Pergamon, New York, 1981, pp. 103-143.

24 Chen, C.W., Lee, G., and Juang, J.-N., "Several Recursive Techniques for Observer/ Kalman Filter System Identification From Data," *Paper No. 92-4386, Proceedings of AIAA Guidance, Navigation and Control Conference*, Hilton Head, SC, August 1992.

25 Juang, J.-N., and Phan, M., "Linear System Identification via Backward Observer Models," *NASA Technical Memorandum* TM-107632, May 1992.

26 Huang, J.-K., Juang, J.-N. and Chen, C.W., "Single-mode Projection Filters for Modal Parameter Identification for Flexible Structures," *AIAA Journal of Guidance, Control and Dynamics*, Vol. 12, No. 4, July-August 1989, pp. 568-576.

27 Chen, C.W., Juang, J.-N., and Huang, J.-K., "Adaptive Linear Identification and State Estimation," *Control and Dynamics Systems: Advances in Theory and Applications*, Vol. 57, Multidisciplinary Engineering Systems: Design and Optimization Techniques and Their Application, edited by C.T. Leondes. New: Academic Press, Inc., 1993, pp. 331-368.

28 Gustavsson, I., Ljung., and Soderstrom, T., "Survey Paper: Identification of Processes in Closed-loop-Identifiability and Accuracy Aspects," *Automatic*, Vol. 13, 1977, pp. 59-75.

29 Gustavsson, I., Ljung., and Soderstrom, T., "Choice and Effect of Different Feedback Configurations," *Trends and Progress in System Identification*, edited by P. Eykhoff, Pergamon, New York, 1981, pp. 367-388.

30 Juang, J.-N., and Phan, M., "Identification of System, Observer, and Controller from Closed-loop Experimental Data," *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 1, 1994, pp. 91-96.

31 Huang, J.-K., Hsiao, M.-H., and Cox, D.E., "Indirect Identification of Linear Stochastic Systems with Known Feedback Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, July-August 1996, pp. 836-841.

32 Phadke, M.S., and Wu, S.M., "Identification of Multi-Input-Multi-Output Transfer Function and Noise Model of a Blast Furnace from Closed-loop Data," *IEEE Transactions*, Vol. AC-19, 1974, pp. 944-951.

33 Chen, C.W., Huang, J.-K., Phan, M., and Juang, J.-N., "Integrated System Identification and Modal State Estimation for Control of Flexible Space Structures," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, 1992, pp. 88-95.

34 Phan, M., Horta, L.G., Juang, J.-N., and Longman, R.W., "Linear System Identification via an Asymptotically Stable Observer," *Journal of Optimization Theory and Application*, Vol. 79, No. 1, 1993, pp. 59-86.

35 Chen, C.W., Huang, J.-K., and Juang, J.-N., "Identification of Linear Stochastic Systems Through Projection Filters," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 4, July-August 1995, pp. 767-772.

36 Hsia, T.C., "On Least Squares Algorithms for System Parameter Identification," *IEEE Transactions on Automatic Control*, February 1976, pp. 104-108.

37 Groom, N.J., Britcher, C.P., "A Description of a Laboratory Model Magnetic Suspension Test Fixture with Large Angular Capability," *Proceedings of the First IEEE Conference on Control Applications*, Vol. 1, 1992, pp. 454.

38 Groom, N.J., "Analytical Model of a Five Degree of Freedom Magnetic Suspension Systems using Electromagnets in a Planar Array," NASA CP-10066, Vol. 1, March 1991, pp. 355-376.

39 Groom, N.J., Britcher, C.P., "Open-loop Characteristics of Magnetic Suspension Systems Using Electromagnets Mounted in a Planar Array," NASA-TP 3229, November 1992.

40 Ghofrani, M., "Approaches to Control of the Large Angle Magnetic Suspension Test Fixture," Master thesis, Old Dominion University, Norfolk, Virginia, Dec., 1992 and NASA CR-191890, Dec., 1992.

41 Dolye, J.C. and Stein, G., "Robustness with Observers," *IEEE Trans. on Automatic Control*, Vol.24, 1979, pp. 607-611.

42 Dolye, J.C. and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/ Modern Synthesis," *IEEE Trans. on Automatic Control*, Vol. 26, 1981, pp. 4-16.

43 Stein, G. and Athans, M., "The LQG/LQR Procedure for Multivariable Feedback Control Design," *IEEE Trans. on Automatic Control*, Vol. 32, 1987, pp. 105-114.

44 Lewis, F.L. and Syrmos V.L., *Optimal Control*, Second edition, John Wiley & Son, Inc., New York, 1995.

45 Kwakernaak, H. and Sivan, R., *Linear Optimal Control Systems*, John Wiley & Son, Inc., New York, 1972.

46 Franklin, G.F., Powell, J.D. and Workman, M.L., *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

47 Gelb, A., *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, Massachusetts, and London, England, 1974.

48 Haykin, S., *Adaptive Filter Theory*, Second edition, Prentice-Hall, Englewood Cliffs, New Jersey 07632, 1991.

49 Hsiao, M.H., "Explicit and Iterative LQG Controller Design," Ph.D. Dissertation, Old Dominion University, Norfolk, VA, May 1994.

50 Monopoli, R.V., "Model Reference Adaptive Control with an Augmented Error Signal," *IEEE Trans. on Automatic Control*, Vol. AC-19, 1974, pp. 474-482.

51 Astrom, K.J., U. Borisson, L. Ljung, and B. Wittenmark, "Theory and Application of Self-Tuning Regulator," *Automatica*, Vol. 13, No. 5, 1977, pp. 457-476.

52 Juang, J.-N. and Phan, M.G., "Deadbeat Predictive Controllers," *NASA Technical Memorandum*.

53 Kuo, C.H., Schoen, M.P., Chinvorarat, S., and Huang, J.-K., "Closed-Loop System Identification by Residual Whitening," To appear in the proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, August 10-12, 1998.

54 Schoen, M.P., Kuo, C.H., Chinvorarat, S., Huang, J.-K., and Lee, H.C., "Input Design For System Under Identification," To appear in the proceedings of the AIAA Guidance, Navigation, and Control Conference, Boston, MA, August 10-12, 1998.

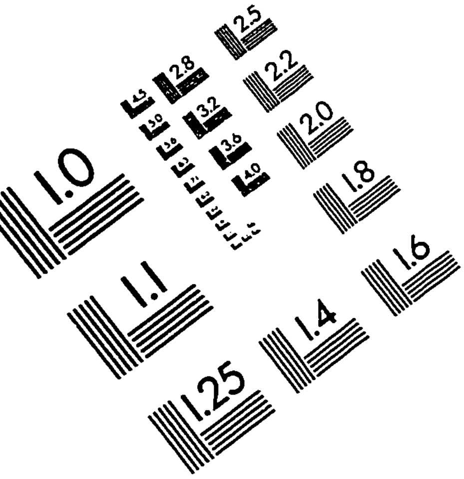
VITA

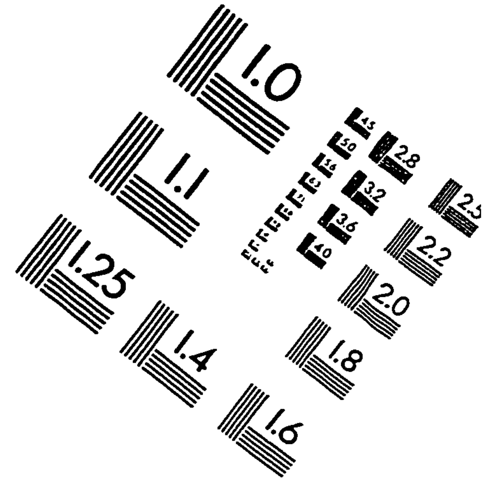
The educational background of Chien-Hsun Kuo is as follows: he completed his Doctoral degree in Mechanical Engineering at Old Dominion University in May 1998. Also, Mr. Kuo received his Master of Engineering degree from the Department of Mechanical Engineering at Old Dominion University in Dec. 1994, and Bachelor of Science degree in Mechanical Engineering from Feng-Chia University, Taichung, Taiwan in June 1990. After receiving B.S. degree, Mr. Kuo served in Taiwan's military for his two-year obligation. He worked in the industry and prepared for his overseas study for one year. Then, he went to United states in August 1993. His research interest includes system identification, controller design and noise reduction.

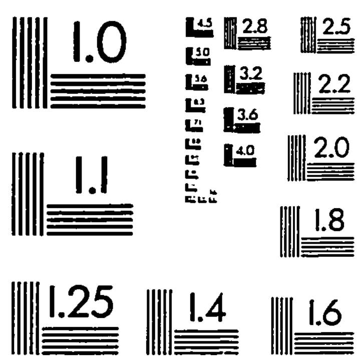
The address for the Department of Mechanical Engineering is Old Dominion University, Kaufman Hall Room 238, Norfolk, VA 23529-0247.

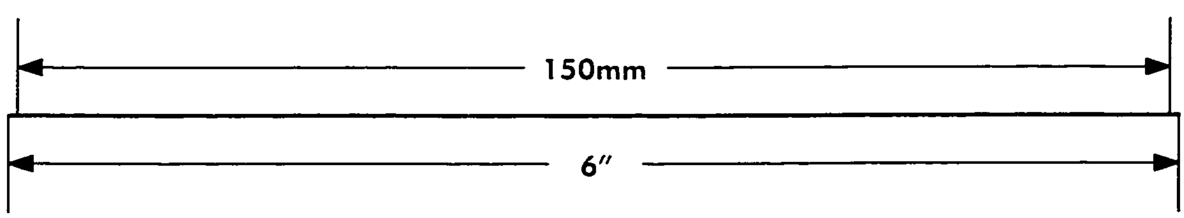
This dissertation was typed by the author using the FrameMaker for X-Windows version 5.0 software.

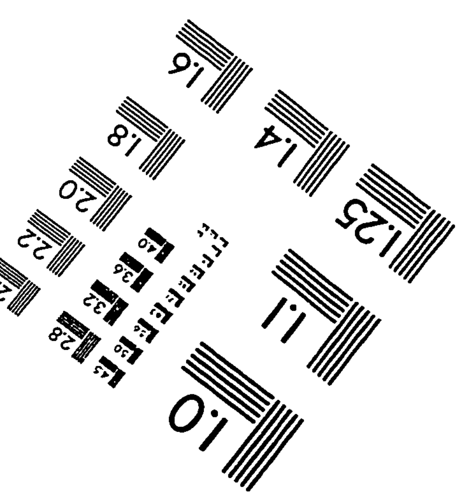
IMAGE EVALUATION TEST TARGET (QA-3)

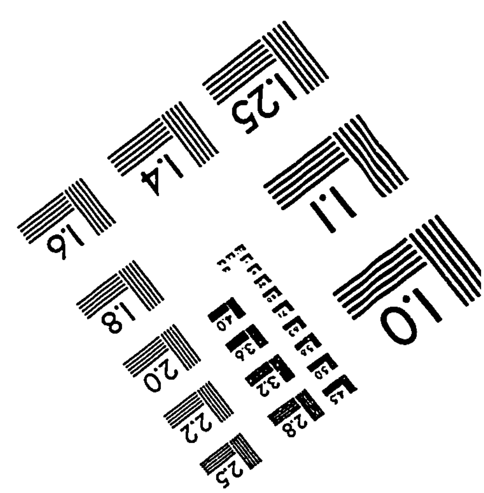












APPLIED IMAGE. Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989