Modeling the Impulse Response of an Office Room

Piya Kovintavewat

EX 055/1998
Department of Signals and Systems
Chalmers University of Technology
G?teborg, Sweden

Submitted to the School of Electrical and Computer Engineering, Chalmers University of Technology, in partial fulfillment of the requirement for the degree of Master of Science in Digital Communication Systems and Technology Program.

Supervisors: Assistant Prof. Jonas Sj?berg
Dr. Ulf Lindgren
Mr. Lester S. H. Ngia

Examiner: Assistant Prof. Jonas Sj?berg

Abstract

Repetition of signals due to reflection from ceiling, wall, floor and other objects in an enclosed environment can be perceived as the acoustic echo. It is resulted from the hands-free operation on the cellular, teleconferencing application and hearing aid system. Since the echo propagates in an enclosed environment, e.g. a room, the room acoustics acts as a filter. To remove the acoustic echo successfully, an estimate of the room impulse response is needed.

Given experimental data, a system identification technique is employed as a tool to build the estimated model of the room impulse response. Herein, its model parameters can be estimated by means of an off-line or batch method in the least squares sense. Several traditional linear model structures have been presented. However, they often lead to an approximation of very high order. In order to reduce the number of estimated parameters, alternative methods for modeling the room impulse response need to be investigated.

Approximation of the room impulse response by means of the so-called Laguerre and Kautz functions, which are the *z*-transform of a class of orthonormal exponentials, is then studied and examined. One of the most important parameters of these two functions is their dominating pole location. It can be shown that the number of estimated parameters is substantially reduced and the numerical accuracy is improved when the dominating pole is chosen properly. For a fixed model order, there exists the optimum choice of a dominating pole which gives the best approximation. Many methods to find a dominating pole are also given. Furthermore, since the typical echo response can be decomposed into two parts, i.e. the first part has rapid time variation and the second part is slowly decaying towards zero, the use of a two-stage echo canceller is then noteworthy to introduce and analyze.

As a preliminary step to investigate the possibility of reducing the number of estimated parameters by the proposed models, we shall therefore not consider the on-line (recursive or adaptive) method to find the values of the model parameters. Instead, the data used in this thesis project has been collected from the system (echo path) that is made as near stationary as possible in order to be able to use the off-line method to estimate the model parameters.

Acknowledgements

This report is the documentation of a project carried out at the Department of Signals and Systems at CTH in a cooperation with the Ericsson Mobile Communication AB at Lund during the second half of 1998. This thesis project is a compulsory part of the International Master's programs that the author attends, and it leads to the degree of Master of Science in Electrical Engineering.

First of all, I would like to give my sincere gratitude to my supervisors, Assistant Prof. Jonas Sj?berg at the Department of Signals and Systems at CTH and Dr. Ulf Lindgren at Ericsson Mobile Communication AB, who have provided me with a great chance to do this interesting thesis project and have given me invaluable expertise throughout the project.

In addition, I would also like to thank Mr. Lester S. H. Ngia, a Phd. student, at the Department of Signals and Systems at CTH for his enthusiastic support, guidance and invaluable discussions.

Finally, special thanks are given to the rest of the staff at the Department of Signals and Systems at CTH and at Ericsson Mobile Communications AB, who provided me with great supports along this project.

Contents

1. Introduction **1**
- 1.1 Acoustic Echo 1
- 1.2 Acoustic Feedback Coupling Mechanism 2
- 1.3 Echo Canceller 2
- 1.4 Problem Definition 3
- 1.5 Organization of the Thesis Report 3

2. Preliminary Theory and Terminology **5**
- 2.1 LTI system 5
  - 2.1.1 Linearity 5
  - 2.1.2 Time-invariance 5
  - 2.1.3 Causality 6
  - 2.1.4 Stability 6
- 2.2 Difference Equation 6
- 2.3 Z-Domain Analysis 7
  - 2.3.1 Stability 8
  - 2.3.2 Maximum, Minimum and Mixed phase systems 8
- 2.4 Linear Algebra 9
  - 2.4.1 Vectors 9
  - 2.4.2 Norm 9
  - 2.4.3 Inner product 10
  - 2.4.4 Projection 11
  - 2.4.5 Linear combination and linear independence 11
  - 2.4.6 Bases and coordinates 11
  - 2.4.7 The Gram-Schmidt procedure 12
  - 2.4.8 Matrices 12
  - 2.4.9 Linear equation 13

3. Fundamental of System Identification **16**
- 3.1 Dynamic System 16
- 3.2 Type of Models 16
- 3.3 System Identification Procedures 17
  - 3.3.1 Experiment design 17
  - 3.3.3 Structure and order estimation 17
  - 3.3.4 Parameter estimation 18
  - 3.3.5 Model validation 18
- 3.4 Summary 19

4. Least Squares Method **21**
- 4.1 Least Squares Method 21
- 4.2 Statistical Properties of the Ideal Estimator 24
  - 4.2.1 Unbiased and asymptotically unbiased estimator 24
  - 4.2.2 Consistent estimator 24

4.2.3 Efficient estimator 25
4.2.4 Robust estimator 25

5. Linear Model Structures 26
5.1 Model of the Acoustic Echo Generating System 26
5.2 Review of Linear Model Structure 27
4.2.1 The Finite Impulse Response model structure (FIR) 27
4.2.2 The Auto Regressive model structure with an eXternal input(ARX) 28
4.2.3 The Output Error model structure (OE) 29

6. Orthonormal Bases for System Identification 31
6.1 Motivation 31
6.2 System Representation 33
6.3 Discrete Laguerre Function 34
6.4 Discrete Kautz Function 35
6.5 Implementation of Orthonormal Basis Function 37
6.5.1 Orthonormal model with FIR structure 37
6.5.2 Orthonormal model with ARX structure 37
6.6 Two-Stage Echo Canceller 38

7. Optimum Choice of Dominating Pole 40
7.1 Based on a Given Set of Impulse Responses 40
7.2 Based on Minimizing the Loss Function 42
7.3 Based on the Derivative of the Squared Error Equation 44
7.3.1 Optimality condition of discrete Laguerre model 44
7.3.2 Optimality condition of discrete Kautz model 46
7.4 Suggestion on Model Reduction 47
7.4.1 Threshold technique 47
7.4.2 The ordering of the poles 48

8. Simulated Examples 49

9. Experimental Result on Real Acoustic Echo Data 59
9.1 Process of generating and collecting the data 59
9.2 Assumption 59
9.3 Experimental result 60

10. Conclusions 64

11. Suggestions for Future Works 65

12. References 66

Appendix A Table of the fits of error used in section 8 **68**

Appendix B Table of the fits of error used in section 9 **71**

Appendix C Matlab script files **74**

1. Introduction

In this section, the background of an acoustic echo is described. More details in an acoustic feedback coupling mechanism are introduced. The general concept of an echo canceller is described. Lastly, the problem definition of the thesis project and the organization of the thesis report are also given.

1.1 Acoustic Echo

Echo is a phenomenon in which a delayed and distorted replica of an original sound or electrical signal is reflected back to the source. Acoustic echo is caused by the reflection of the loudspeaker's sound from ceiling, wall, window, floor and other objects in an enclosed environment, e.g. in a room, back to the microphone as shown in Fig.1.1. In addition, acoustic echo may also result from a feedback path set up between the speaker and the microphone in teleconference and hearing aid systems. Note that, the near-end talker B can be viewed as combined signals between the pure speech signal from talker B and the acoustic echo generated by far-end talker A signal.

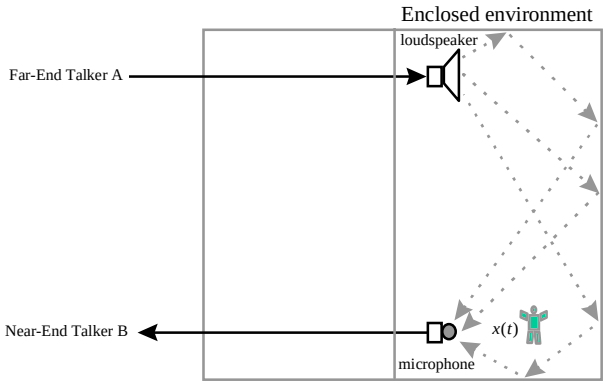

Figure 1.1: An acoustic echo generating system

The effects of an echo depend upon the time delay between the incident and the reflected waves, the strength of the reflected waves, and the number of echo paths through which the waves are reflected. In general, acoustic echo is reflected from many different surfaces and propagates through different paths. If the time delay is not too long, the acoustic echo can be viewed as a soft reverberation which enhances the artistic quality of a musical performance in a concert hall or a church hall. However, the acoustic echo arriving several tens of ms after the direct waves is undesirable and annoying. The echo cancellation system is therefore required when the effect of an echo exceeds an acceptable level.

Naturally, the problem of acoustic echo is quite complicated for a number of reasons:

- The time delay of acoustic echo is too long (up to a second). This implies that a large number of estimated parameters, probably in the order of a few thousands, are needed in the transversal FIR filter to obtain a reasonable approximation.

- The characteristic of an acoustic echo path is rather non-stationary owing to opening or closing of a door, or a moving person.
- The acoustic echo results from reflection of the loudspeaker's sound from many different paths, e.g. from walls, windows, floors and other objects in an enclosed environment.
- The echo path is not well approximated by a FIR and IIR linear filter because it has a mixture of linear and non-linear characteristics. The reflection of acoustic signals inside a room is almost linearly distorted, but the loudspeaker produces non-linearity.

1.2 Acoustic Feedback Coupling Mechanism

The problem in hand-free telephones, e.g. in teleconference system, and hearing aid system is the feedback coupling of the sound waves between the loudspeakers and the microphones. Acoustic feedback is also easily experienced in howling if a significant portion of the sound energy, transmitted to the loudspeaker, is received back to the microphone and circulated in the feedback loop. The total round-gain of the acoustic feedback loop relies on the frequency responses of the electrical and the acoustic signal paths. If the microphone-speaker-room system is excited at a frequency whose loop gain is greater than unity, then the far-end signal is amplified in a loop and results in a feedback howling.

Many methods for eliminating such an acoustic feedback are as follows:

- Install a frequency shifter or a phase shifter in the electrical path of the feedback loop. Each time the signal travels in the feedback loop it will be shifted by a few hertz before being retransmitted by the loudspeaker. This method can merely reduce the effect of howling but not the overall echo.
- Reduce the system gain at frequencies where acoustic oscillations emerge by using an adaptive notch filter. The drawback of this method is that there exist some distortion on the desired signal frequencies.
- Use the adaptive echo cancellation system which is known as the most effective method to eliminate the echo.

1.3 Echo Canceller (EC)

An effective way of removing the acoustic echo signal is to use an echo canceller. The EC first estimates the characteristics (or transfer function) of the true echo path, and then generates a replica of the true echo signal that is used to eliminate the true echo signal by subtraction. Since the transfer function of the echo path is generally unknown, an estimate of it must be determined. Several methods are employed to find the approximated transfer function which will be discussed in the sequels. The acoustic echo cancellation in an enclosed environment is depicted in Fig.1.2.

The synthetic echo, $\hat{r}(t)$, is generated by passing the far-end signal *u(t)* through the estimated model of the true echo path, *He* represents the transfer function of the true echo path, *r(t)* is the true echo signal produced from the true echo path, i.e. $r(t) = He * u(t)$ where $*$ denotes the convolution sum, and *x(t)* is the desired signal from the echo canceller. Therefore, the combined signal $y(t) = r(t) + x(t)$ is subtracted from the synthetic echo, $\hat{r}(t)$, to obtain the canceller error signal,

$$e(t) = (r(t) + x(t)) - \hat{r}(t) \tag{1.1}$$

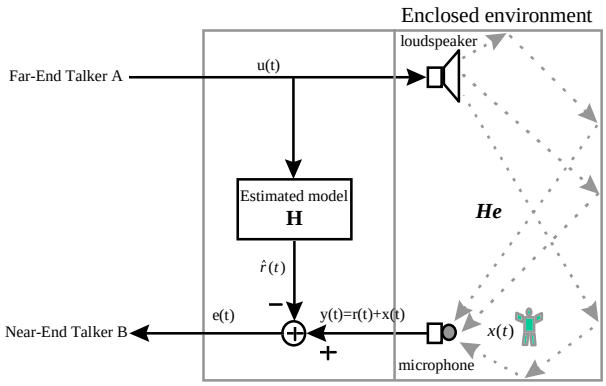

Figure 1.2 The acoustic echo cancellation in an enclosed environment

Typically, a good echo canceller yields *e(t) = x(t)*. Note that, *x(t)* can be perceived as the near-end speaker's speech signal and/or unknown disturbance. To model the echo path correctly, one must avoid to interpret *x(t)* as a part of the true echo signal. In this thesis report, *x(t)* is assumed for simplicity to be low-level disturbance and uncorrelated with the true echo signal.

In theory, the number of parameters employed in the filter depends upon the echo path delay and the length of the impulse response of an echo path. For the echo canceller to work suitably, the number of parameters should have a length greater than the combined effect of the true echo path's impulse response length and the echo path delay. Let T_s be the sampling period of the digitized speech signal, M be the number of parameters used in the filter, and τ be the combined effect to be considered. Consequently, one obtain [14]

$$MT_s > \tau \tag{1.2}$$

Since the typical value of T_s is 125 μs for a standard digital communication system, it is obvious that, if τ = 500 ms, M > 4000 parameters are required in the filter.

1.4 Problem Definition

In hands-free operation of the cellular, an undesired acoustical feedback of the loud-speaker occurs. To avoid transmitting the loudspeaker signal via the microphone, an acoustic echo cancellation algorithm must be employed. However, since the loudspeaker's signal propagates in an enclosed environment, e.g. a room, the room acoustics then acts as a filter. In order to eliminate the acoustic echo, an estimate of the room impulse response is needed.

The impulse response of an office room are often modeled as a Finite Impulse Response (FIR). A drawback with a FIR model is that the number of parameters to be estimated is considerably large, probably in the order of 4000. This will absolutely affect the complexity of any acoustic echo cancellation algorithms. To reduce the number of estimated parameters, alternative models are needed to investigate.

Therefore, the aim of the thesis project is to try alternative representations for room impulse response which require fewer parameters than a FIR model.

1.5 Organization of the Thesis Report

In the next section, some preliminary theories and terminology about LTI system, difference equation, *z*-domain analysis and linear algebra are given. The fundamental theory of system identification that is employed as a basic tool to construct the estimated model of the system of interest is described in section 3. The least squares method used to find the suitable value of estimated parameters by means of an off-line method is explained in section 4. Section 5 presents a few relevant linear model structures. The use of orthonormal basis function, which is employed to reduce the number of estimated parameters, is introduced in section 6. The concept of a two-stage echo cenceller, which has the first stage as a FIR model and the second stage as an orthonormal model, is also described. It has shown that the performance of using an orthonormal model depends primarily upon the model order and its dominating pole. For a fixed model order, there exists the optimal dominating pole that yields the best approximation. Several methods employed to find such an optimal dominating pole will then be given in section 7. Remarkable suggestion on the concept of model reduction is also introduced. Three examples based on the simulated data are shown in section 8. The process of generating and collecting the real acoustic echo data and its experimental result are explained in section 9. Eventually, the conclusions of the thesis project and the suggestions of future works are presented in section 10 and 11, respectively.

2. Preliminary Theory and Terminology

The aim of this section is to provide a brief overview of the fundamentals of LTI system, difference equation, *z*-domain analysis and linear algebra [17]. Some notations and terminology are introduced which will be used throughout this report.

2.1 LTI system

The system that is both linear and time-invariant is generally called as a *linear time-invariant (LTI)* system. In this thesis project, we shall mainly deal with the LTI system that is also causal and stable. Such properties can be described as follows:

2.1.1 Linearity

Let T[-] denote the system. A system is said to be *linear* if, for any two inputs $u_1(n)$ and $u_2(n)$ and for any two constants *a* and *b*, the following equation is satisfied

$$T\left[au_1(n) + bu_2(n)\right] = aT\left[u_1(n)\right] + bT\left[u_2(n)\right] \tag{2.1}$$

This property implies that if the input can be decomposed into the sum of weighted and shifted unit samples, such that

$$u(n) = \sum_{k=-\infty}^{\infty} u(k)\delta(n-k) \tag{2.2}$$

the output is then

$$y(n) = \sum_{k=-\infty}^{\infty} u(k)T\left[\delta(n-k)\right] = \sum_{k=-\infty}^{\infty} u(k)h_k(n) \tag{2.3}$$

where $h_k(n) = T\left[\delta(n-k)\right]$ is the response of the system to the delayed unit sample $\delta(n-k)$. Apparently, a linear system is completely specified once the signals $h_k(n)$ are known.

2.1.2 Time-invariance (or Shift-invariance)

A system is said to be *time-invariant* if a shift in the input signal by n_0 results in an equivalent shift in the output signal by n_0. That is, if $y(n)$ is the response of a time-invariant system to an input $x(n)$, thereby for any shift in the input, $x(n-n_0)$, the response of the system will be $y(n-n_0)$. In effect, time-invariance means that the properties and/or responses of the system do not change in time.

For a time-invariant system, if $h(n) = T\left[\delta(n)\right]$ is the response to a unit sample $\delta(n)$, the response to $\delta(n-k)$ is $h(n-k)$. As a consequence, the LTI system can be represented by

$$y(n) = x(n) * h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.4}$$

where $*$ denotes the convolution sum. Since the convolution sum allows one to evaluate the response of the LTI system to arbitrary input *x(n)*, the LTI system is uniquely characterized by its response, *h(n)*, to a unit sample. Therefore, *h(n)* is referred to as the *unit sample response* of the system.

2.1.3 Causality

Causality is an important property for dealing with all real-time systems. The system is *causal* if the output *y(n)* at an arbitrary time $n = n_0$ depends only upon the input for $n \leq n_0$. In other words, the output of a causal system at the present time n_0 relies only on the present and/or past input values, not on the future values. Equivalently, if a system is linear and time-invariant, hence it is a causal system if and only if $h(n) = 0$ for $n < 0$.

2.1.4 Stability

It is desirable to have the output that is bounded in amplitude whenever the input is bounded, i.e. if for any bounded system input *x(n)*,

$$| x(n) | \leq B_1 < \infty \tag{2.5}$$

the system output *y(n)* is also bounded,

$$| y(n) | \leq B_2 < \infty \tag{2.6}$$

where B_1 and B_2 are finite constants. A system with this property is said to be *stable* in *Bounded-Input Bounded-Output (BIBO)* sense. A LTI system is BIBO stable if and only if the unit sample response *h(n)* is absolutely summable

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty \tag{2.7}$$

2.2 Difference Equation

The important classes of LTI systems are those for which the input *x(n)* and output *y (n)* can be represented in terms of a difference equation with constant coefficients. Such a difference equation plays an important role for describing the input/output relationship in many fields of sciences, such as signal processing, communication, etc., because it is easy to interpret and analyze. The general difference equation can be expressed by

$$y(n) + \sum_{k=1}^{na} a_k y(n-k) = \sum_{k=0}^{nb} b_k x(n-k) \tag{2.8}$$

where *na* and *nb* determine the *order* of the system, and a_k and b_k are the *filter coefficients* that characterize the system. Obviously, the output *y(n)* is a linear combination of the past

output values, *y(n-k)* for *k=1,2,...,na*, coupled with the past and present input values, *x(n-k)*, for *k=0,1,2,...,nb*.

As an special case when *na = 0*, the difference equation becomes

$$y(n) = \sum_{k=0}^{nb} b_k x(n-k) \tag{2.9}$$

The output is simply a weighted sum of the past and present values. Consequently, the unit sample response is finite in length,

$$h(n) = \sum_{k=0}^{nb} b_k \delta(n-k) \tag{2.10}$$

and the system is then referred to as a *Finite length Impulse Response(FIR)* system. However, when $na \neq 0$, the unit sample response is infinite in length and the system is referred to as an *Infinite length Impulse Response (IIR)* system.

2.3 Z-Domain Analysis

The *z-transform* of a discrete-time signal *x(n)* is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{2.11}$$

It is only defined when the sum in Eq.(2.11) converges. The *z*-transform usually converges for only a certain range of complex-valued variable *z* known as the *region of convergence (ROC)*, the domain where *z*-transform of *x(n)* exists. Typically, this region must always be specified along with *X(z)* in order for *z*-transform to be complete.

The *inverse z-transform* is a method to find the corresponding signal *x(n)* from a given *X(z)*. Based on the Cauchy integral theorem which is given by

$$\frac{1}{2\pi j} \oint_{\Gamma} z^{k-1} dz = \delta(k) \tag{2.12}$$

where Γ is a closed path in a counterclockwise direction, including the origin, the inverse *z*-transform of *X(z)* is defined as

$$x(n) = \frac{1}{2\pi j} \oint_{\Gamma} X(z) z^{n-1} dz \tag{2.13}$$

where Γ is a contour integration in the ROC. It is also of importance to know when the *z*-transform exists, i.e. the *z*-transform of *x(n)* exists if and only if the sequence $\{x(n)r^{-n}\}$ is absolutely summable for some values of *r*, i.e.

$$\sum_{n=-\infty}^{\infty} \left| x(n) r^{-n} \right| < \infty \tag{2.14}$$

The *z*-transform of special importance in the design and analysis of LTI systems is the *system function*, which is the *z*-transform of the unit sample response

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n} \tag{2.15}$$

For a FIR system given in Eq.(2.9), the corresponding system function is a polynomial in z^{-1}

$$H(z) = \sum_{k=0}^{nb} b_k z^{-k} = b_0 \prod_{k=1}^{nb} (1 - z_k z^{-1}) \tag{2.16}$$

where the roots of this polynomial, z_k, are called *zeros* of the filter. This type of system function is normally referred to as an *all-zero* filter. For the case of an IIR system given in Eq. (2.8), the system function is a ratio of two polynomial in z^{-1}

$$H(z) = \frac{\sum_{k=0}^{nb} b_k z^{-k}}{1 + \sum_{k=1}^{na} a_k z^{-k}} = b_0 \frac{\prod_{k=1}^{nb} (1 - z_k z^{-1})}{\prod_{k=1}^{na} (1 - p_k z^{-k})} \tag{2.17}$$

where the roots of the numerator polynomial, z_k, are the zeros of *H(z)* and the roots of the denominator polynomial, p_k, are called the *poles*. A special case when *nb=0*, the system function in Eq.(2.17) is known as an *all-pole* filter, i.e.

$$H(z) = \frac{b_0}{1 + \sum_{k=1}^{na} a_k z^{-k}} = \frac{b_0}{\prod_{k=1}^{na} (1 - p_k z^{-k})} \tag{2.18}$$

2.3.1 Stability

A LTI system is BIBO stable if and only if the ROC of *H(z)* includes the unit circle, i.e. *H(z)* with ROC: $|z| \geq 1$. On the other hand, a causal LTI system is said to be BIBO stable if and only if all the poles of *H(z)* lie inside the unit circle, i.e. $|p_k| < 1$ for all *k*.

2.3.2 Maximum, Minimum and Mixed phase systems

The system is said to be a *minimum phase* system if all zeros are inside the unit circle, while it is a *maximum phase* system if all zeros lie outside the unit circle. If the zeros lie both inside and outside the unit circle, the system is known as a *mixed phase* system. For an IIR system, the inverse system is stable if it is a minimum phase system.

A mixed phase system, *H(z)*, can be decomposed to a minimum phase system, $H_{min}(z)$, and *all-pass* system, $H_{ap}(z)$, such that

$$H(z) = H_{\min}(z) H_{ap}(z) \tag{2.19}$$

Suppose that $H(z) = B(z) / A(z)$ and $B(z) = B_1(z)B_2(z)$, where $B_1(z)$ is a minimum phase system and $B_2(z)$ is a maximum phase system. Then the system function can be written as

$$H(z) = \frac{B_1(z)B_2(z)}{A(z)} = \left(\frac{B_1(z)B_2(z^{-1})}{A(z)} \right) \left(\frac{B_2(z)}{B_2(z^{-1})} \right) \tag{2.20}$$

where

$$H_{\min}(z) = \frac{B_1(z)B_2(z^{-1})}{A(z)} \quad \text{and} \quad H_{ap}(z) = \frac{B_2(z)}{B_2(z^{-1})} \tag{2.21}$$

Note that, an all-pass system, $H_{ap}(z)$, is always stable with maximum phase.

2.4 Linear Algebra

To deal with a problem in signal modeling and system identification, it is convenient to represent the signals in terms of the vector and matrix forms. Such a representation simplifies many mathematical expressions and allows us to draw upon many useful results from linear algebra to solve the problem. Herein, we shall present the fundamentals of linear algebra.

2.4.1 Vectors

A vector is an array of real or complex values for which it is usually used to represent the value of discrete time signals. We shall denote the vector by lowercase bold letters. A column vector **y** with *N* elements, usually referred to as a *N*-dimensional vector, is given by,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{2.22}$$

The transpose of a vector , $\mathbf{y}^T$, is a row vector

$$\mathbf{y}^T = [\, y_1, y_2, ..., y_N \,] \tag{2.23}$$

and the Hermitian transpose, $\mathbf{y}^H$, is defined as the complex conjugate of the transpose of **y**

$$\mathbf{y}^H = (\mathbf{y}^T)^* = \left[y_1^*, y_2^*, \ldots, y_N^* \right] \tag{2.24}$$

where $(\bullet)^*$ denotes the complex conjugate operation.

2.4.2 Norm

Several operations performed on a vector will primarily involve finding the magnitude of a vector. The norm or distance metric is usually used to find the magnitude of a vector. Given $\mathbf{y} = [\, y_1, y_2, ..., y_N \,]^T$, the *Euclidean* or L_2 *norm* is defined as

$$\|\mathbf{y}\|_2 = \left\{ \sum_{i=1}^{N} |y_i|^2 \right\}^{1/2} \tag{2.25}$$

Other useful norms are the *L_1 norm*

$$\|\mathbf{y}\|_1 = \sum_{i=1}^{N} |y_i| \tag{2.26}$$

and the *L_∞ norm*

$$\|\mathbf{y}\|_\infty = \max_i |y_i| \tag{2.27}$$

Since L_2 norm is normally preferable to cope with the problem in system identification, we shall denote L_2 norm by $\|\mathbf{y}\|$, unless indicated otherwise.

Assuming that the norm $\|\mathbf{y}\| \neq 0$, a vector may be *normalized* to have unit magnitude by dividing by its norm. Hence

$$\mathbf{v}_\mathbf{y} = \frac{\mathbf{y}}{\|\mathbf{y}\|} \tag{2.28}$$

is a unit norm that lies in the same direction as **y**. The squared norm of signals *y(n)* is equal to the signal energy, E_y, i.e.

$$\|\mathbf{y}\|^2 = \sum_{n=1}^{N} |y(n)|^2 = E_y \tag{2.29}$$

2.4.3 Inner product

Given two complex-valued vectors $\mathbf{a} = [a_1, a_2, ..., a_N]^T$ and $\mathbf{b} = [b_1, b_2, ..., b_N]^T$, the *inner product* is the scalar value determined by

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^H \mathbf{b} = \sum_{i=1}^{N} a_i^* b_i \tag{2.30}$$

The inner product can also be viewed as the geometrical relationship between two vectors given by

$$\langle \mathbf{a}, \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos\theta \tag{2.31}$$

where θ is the angle between two vectors. Two nonzero vectors, **a** and **b**, are said to be *orthogonal* if their inner product is equal to zero.

$$\langle \mathbf{a}, \mathbf{b} \rangle = 0 \tag{2.32}$$

Additionally, two vectors that are orthogonal and have unit norm are said to be *orthonormal*.

2.4.4 Projection

Projection of a vector **x** onto **y** is defined as

$$proj_{\mathbf{y}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{y} \rangle \frac{\mathbf{y}}{\langle \mathbf{y}, \mathbf{y} \rangle} = \langle \mathbf{x}, \mathbf{y} \rangle \frac{\mathbf{y}}{\|\mathbf{y}\|^2} \tag{2.33}$$

Since the vector $proj_{\mathbf{y}}(\mathbf{x})$ is parallel with **y**, then

$$\mathbf{x} - proj_{\mathbf{y}}(\mathbf{x}) \perp \mathbf{y} \tag{2.34}$$

2.4.5 Linear combination and linear independence

A sum of scaled vectors is called a *linear combination*, i.e. for any arbitrary complex constants of α_i for $i = 1,2,...,N$, the linear combination of vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ is given by

$$\sum_{i=1}^{N} \alpha_i \mathbf{y}_i \tag{2.35}$$

Moreover, a set of vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ is said to be *linear independent* if and only if

$$\sum_{i=1}^{N} \alpha_i \mathbf{y}_i = 0 \tag{2.36}$$

only when $\alpha_i = 0$ for all *i*. If the Eq.(2.36) is satisfied but $\alpha_i \neq 0$ for all *i*, then a set of vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ is said to be *linear dependent*.

2.4.6 Bases and coordinates

A *basis* for the subspace span $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ is defined as a set of linearly independent vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ such that

$$\text{span } \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\} = \text{ span } \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\} \tag{2.37}$$

That is, every vector **x** in span $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ can be written as a linear combination of vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$. If $\langle \mathbf{y_i}, \mathbf{y_j} \rangle = 0$ for i ≠ j, the basis is called an *orthogonal basis*. In addition, if $\langle \mathbf{y_i}, \mathbf{y_j} \rangle = 0$ for i ≠ j and $\|\mathbf{y_i}\| = \|\mathbf{y}_j\| = 1$, the basis is called an *orthonormal basis*.

If $\{\mathbf{y}_1, \mathbf{y_2}, ..., \mathbf{y}_N\}$ are orthonormal bases for span $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$, thus every $\mathbf{x} \in$ span $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ can be written as a linear combination of $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ in such a way that

$$\mathbf{x} = \sum_{i=1}^{N} \alpha_i \mathbf{y_i} \tag{2.38}$$

where α_j is called the *coordinate* of **x** with respect to $\mathbf{y}_j$ and can be computed by

$$\left\langle \mathbf{x}, \mathbf{y}_\mathbf{j} \right\rangle = \left\langle \sum_{i=1}^{N} \alpha_i \mathbf{y}_\mathbf{i}, \mathbf{y}_\mathbf{j} \right\rangle = \sum_{i=1}^{N} \alpha_i \left\langle \mathbf{y}_\mathbf{i}, \mathbf{y}_\mathbf{j} \right\rangle = \sum_{i=1}^{N} \alpha_i \delta_{ij} = \alpha_j \tag{2.39}$$

where δ_{ij} is the Kronecker delta function ($\delta_{ij} = 1$ only if $i = j$, and 0 elsewhere).

2.4.7 The Gram-Schmidt procedure

Given a set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$, we shall construct the orthonormal basis vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ to span $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$. The set of M vectors spans $N = M$ dimensional space only if the $\mathbf{x}_i$, for i = 1, ..., M, are linearly independent. If the $\mathbf{x}_i$ are not linearly independent, then $N < M$.

The number of dimensions, N, required to represent the set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ and a corresponding set of orthonormal basis vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ can be obtained by using the *Gram-Schmidt* procedure. Based on a concept of signal projection, i.e. $\mathbf{x} - proj_\mathbf{y}(\mathbf{x}) \perp \mathbf{y}$, a set of vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ is said to be orthonormal if and only if

$$\mathbf{x} - proj_{\mathbf{y}_1}(\mathbf{x}) - proj_{\mathbf{y}_2}(\mathbf{x}) - \ \ldots \ - proj_{\mathbf{y}_L}(\mathbf{x}) \perp \mathbf{y}_i, \quad \text{for } i = 1,2,...,L \tag{2.40}$$

The recursive procedure used to calculate the orthonormal basis vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$ from $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ can be done as follows

- Assume that $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ are linearly independent.
- Compute the first orthonormal basis vector from $\mathbf{y}_1 = \mathbf{x}_1 / \left\|\mathbf{x}_1\right\|$.
- For j = 2, 3, ..., M
  (a) Calculate the next basis function which is orthogonal to all $\mathbf{y}_i$, for i = 1, 2, ..., j-1

$$\tilde{\mathbf{y}}_j = \mathbf{x}_j - \sum_{l=1}^{j-1} proj_{\mathbf{y}_l}(\mathbf{x}_j) = \mathbf{x}_j - \sum_{l=1}^{j-1} \left\langle \mathbf{x}_j, \mathbf{y}_l \right\rangle \mathbf{y}_l \tag{2.41}$$

  (b) Normalize $\tilde{\mathbf{y}}_j$ by its norm to be orthonormal basis vector $\mathbf{y}_j$.

$$\mathbf{y}_j = \frac{\tilde{\mathbf{y}}_j}{\left\|\tilde{\mathbf{y}}_j\right\|} \tag{2.42}$$

2.4.8 Matrices

An $n \times m$ matrix is an array of real or complex values having n rows and m columns,

$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \tag{2.43}$$

The *transpose* of an $n \times m$ matrix, A^T, is an $m \times n$ matrix that is formed by interchanging the rows and columns of A. Hence, the (i,j)th element becomes the (j,i)th element. The *Hermitian transpose* denoted by A^H is the complex conjugate of the transpose of A, i.e.

$$A^H = (A^*)^T = (A^T)^* \tag{2.44}$$

Let A be an $n \times m$ matrix partitioned into m column vectors

$$A = [\ \mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m\ ] \tag{2.45}$$

where $\mathbf{c}_i = [\ a_{1i}, a_{2i}, ..., a_{ni}\ ]^T$. The *rank* of A, $\rho(A)$, is defined as the number of linearly independent columns in A. The property of the rank of a matrix is that the rank of A is equal to the rank of AA^H and A^HA,

$$\rho(A) = \rho(AA^H) = \rho(A^HA) \tag{2.46}$$

Since the rank of a matrix is equal to the number of linearly independent rows and the number of linearly independent columns, it follows that if A is an $n \times m$ matrix then

$$\rho(A) \leq \min(n,m) \tag{2.47}$$

If A is an $n \times m$ matrix and $\rho(A) = \min(n,m)$, A is said to be of *full rank*. Theoretically, if A is a square matrix of full rank, say $n \times n$ matrix, there exists a unique inverse matrix of A, A^{-1}, such that

$$A^{-1}A = AA^{-1} = \mathbf{I} \tag{2.48}$$

where

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{2.49}$$

is the identity matrix which has ones along the main diagonal and zeros elsewhere. In this case, a matrix A is said to be *invertible* or *nonsingular*. If, however, A is not the full rank, $\rho(A) < n$, the inverse matrix of A does not exist and it is said to be *noninvertible* or *singular*.

2.4.9 Linear equation

Typically, it is useful to represent the relationship between input and output signals in terms of linear equation because its solution will then be simplified and can be solved easily. In order to solve the linear equation, it is necessary to characterize the form of the solution in terms of existence and uniqueness.

Consider a set of linear equation with m unknown parameters, g_i for $i=1,2,...,m$, to be estimated.

$$y_1 = a_{11}g_1 + a_{12}g_2 + \ldots + a_{1m}g_m$$

$$y_2 = a_{21}g_1 + a_{22}g_2 + \ldots + a_{2m}g_m$$
$$\vdots$$
$$y_n = a_{n1}g_1 + a_{n2}g_2 + \ldots + a_{nm}g_m$$

These linear equations can also be written in matrix form as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} \tag{2.50}$$

or, equivalently,

$$Y = A\theta \tag{2.51}$$

where $Y = [\, y_1, y_2, \ldots, y_n \,]^T$, $\theta = [\, g_1, g_2, \ldots, g_m \,]^T$ and A is given in Eq.(2.43). The solution of finding θ depends upon the size of $n \times m$ and the rank of matrix A.

- **Square matrix** ($m = n$)

If A is a square $n \times n$ matrix, the solution to the linear equations depends on whether or not A is singular. If A is nonsingular, then the inverse matrix A^{-1} exists and the solution is uniquely defined by

$$\theta = A^{-1}Y \tag{2.52}$$

Nevertheless, if A is singular, there may be either no solution or many solutions.

- **Rectangular matrix** (for $n < m$)

When $n < m$, there are fewer linear equations than unknown parameters. If the equations are not *inconsistent*, there exists many vectors satisfying the equations, i.e. the solution is *underdetermined* or *incompletely specified*. The approach to obtain a unique solution is to find the vector satisfying the equations that have a minimum norm, i.e.

$$\min\|\theta\| \quad \text{such that} \quad Y = A\theta \tag{2.53}$$

If the rank of A is n, the $n \times n$ matrix AA^H is invertible and the *minimum norm solution* is

$$\theta = A^H (AA^H)^{-1} Y = \Theta Y \tag{2.54}$$

where

$$\Theta = A^H (AA^H)^{-1} \tag{2.55}$$

is called the *pseudo-inverse* of the matrix A for the underdetermined problem.

- **Rectangular matrix** (for $n > m$)

When $n > m$, there are more linear equations than unknown parameters and, in general, no solution exists. The equations are *inconsistent* and the solution is said to be *overdetermined*. However, the parameter vector θ can be solved by means of the *least squares method* which will be discussed in section 4. The aim is just to find the optimal parameter θ leading to the best approximation to Y.

3. Fundamental of System Identification

System identification deals with the problem of constructing the mathematical model of a dynamic system based on a given set of experimental data. Several fundamental concepts about system identification are introduced in this section. The procedures of building the estimated model by means of system identification technique are also given.

3.1 Dynamic System

The dynamic system can be conceptually illustrated as in Fig.3.1. The system is commonly stimulated by external stimuli. External signals that can be controlled by users are called input *u(t)*, while the other which cannot be manipulated is called disturbance *v(t)*. The observable signals we are interested in are called output *y(t)*. Typically, for any *dynamic systems*, it implies that the current output value depends not only on the current external stimuli but also on their past values.

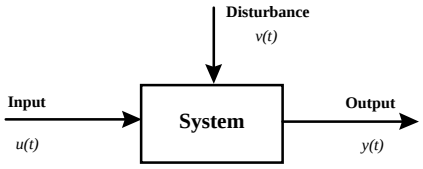

Figure 3.1: A dynamic system.

3.2 Type of Models

The purpose of a model is to describe how the various variables of the system relate to each other. The relationship among these variables is called a *model* of the system. Modeling the system of interest is considerably useful in many areas of science as an aid to properly describe the system's behavior. A good model must reflect all properties of such an unknown system. There are several kinds of models which can be classified as follows:

- Mental models which do not involve a mathematical formula. For instance, when driving a car, one is required the knowledge of turning the wheel, pushing the brake, etc.
- Graphical models which are also suitable for some certain systems using numerical tables and/or plots. For example, any linear systems can be uniquely described by their impulse, step, or frequency response which can be obviously represented in graphical forms.
- Mathematical models which describe how the variables relate to each other in terms of mathematical expressions such as differential and difference equations.

The mathematical models are useful in practice because they provide a description of the system's behavior by mathematical expressions which are simply to examine and analyze. Mathematical models can be derived in two ways, i.e.

- *Modeling* refers to derivation of models from basic laws in physics, economics, etc., to describe the dynamic behavior of a phenomenon or a process. It is done by subdividing an unknown system into subsystems, whose properties are well known from the basic laws, and thus combining these subsystems mathematically to obtain the whole system.
- *Identification* refers to the determination of dynamic models from experimental data. It consists of the establishment of identification experiment, the determination of a suitable form of a model coupled with its parameters, and a validation of the model.

3.3 System Identification Procedures

The system identification problem is to build a mathematical model from a given set of observed input and output data. These data can be obtained from the experiments which are carefully designed and performed on the system. A model is then fitted to the recorded data by assigning suitable values to its parameters. In many applications, the unknown system is so complicated that it is not possible to obtain a reasonable model by using only physical insight. This usually happens when a model based on physical insight contains a number of unknown parameters even if such a model is derived from physical laws. As a result, identification techniques must be applied to estimate its unknown parameters.

Generally, the models obtained by identification techniques have limited validity; for example, they are valid for a certain experiment, a certain input, etc. Furthermore the resulting estimated parameters can be solely used as tools to provide a good representation of the system's overall behavior. However, it is still an attractive method to build the model due to simplicity and flexibility. The system identification procedures can be categorized as the following steps:

3.3.1 Experiment design

The experiment is performed by exciting the system with some sort of chosen input signal, and thus observing and recording the output signals over a certain time. The aim of this experiment is to obtain the input and output data as maximally informative as possible in the presence of unknown disturbances.

3.3.2 Structure and order estimation

Practically, the stable LTI system can be sufficiently modeled by a linear black box model structure which can be written as

$$A(q) * y_0(t) = \frac{B(q)}{F(q)} * u(t - nk) + \frac{C(q)}{D(q)} * w(t) \qquad (3.1)$$

where $y_0(t)$, $u(t)$ and nk are the output, input and delay respectively. The noise $w(t)$ is white sequences with zero mean and unit variance. The time shift is then represented by q such that $q^{-1}u(t) = u(t-1)$. Each polynomial transfer function in the shift operator q is given by

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_{na} q^{-na}$$
$$C(q) = 1 + c_1 q^{-1} + c_2 q^{-2} + \ldots + c_{nc} q^{-nc}$$

$$D(q) = 1 + d_1 q^{-1} + d_2 q^{-2} + \ldots + d_{nd} q^{-nd}$$
$$F(q) = 1 + f_1 q^{-1} + f_2 q^{-2} + \ldots + f_{nf} q^{-nf}$$
$$B(q) = b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-nb}$$

where *na*, *nc*, *nd*, *nf* and *nb* are known as the order of each polynomial. Several model structures can be derived [13] from Eq.(3.1). For example, FIR model is derived when *A(q) = C(q) = D(q) = F(q) = 1*, ARX model corresponds to *C(q) = D(q) = F(q) =1*, OE model is obtained when *A(q) = C(q) = D(q) =1*, etc. FIR, ARX, and OE models will be more described in section 5.

3.3.3 Parameter estimation

Having selected the model structure, the next step is to estimate its parameters in order to completely characterize the system of interest. This is known as a key step in system identification. In the literature, many methods aimed at minimizing the influence of noise on the estimates can be found. In most of these the identification and estimation steps are linked together, because it is almost impossible to verify the quality of a model without estimating its parameters, and it is essential to define a model structure before its parameters can be estimated. The approach that is used to estimate the model parameters based only on a given set of observed data is often called *off-line* or *batch*. In section 4, we shall present the least squares method which is usually used to find the optimal values of the model parameters.

3.3.4 Model validation

After the model structure and its parameters have been suitably chosen according to a certain criteria, such a model is needed to verify whether it can well characterize the system or not. There exists many tools for model validation for which we shall address only three principle methods, namely residual check, model fit and cross validation.

- **Residual Check**

Due to a presence of noise, there will be some differences between the observed output and the estimated output. These differences are known as *residuals.* By investigating them, it is possible to infer whether or not there still exist errors from the estimated model. If so, this result will be fed back to the model building process. This check can be done by using the command “resid” in Matlab program in which the auto-correlation function of the error and the cross-correlation between error and the input data are computed and displayed within 99% confidence interval. The model could be acceptable if the residue plot lies inside this region, but it would be rejected if the residue plot goes outside this interval. A significant peak at lag *k* shows that the effect from input *u(t-k)* on *y(t)* is not appropriately described. A rule of thumb [11] is that a slowly varying cross-correlation function outside the confidence region is an indication of too few poles, while the shape peaks indicate too few zeros or wrong delays.

- **Model Fit** (Fit of Error)

Model fit is used to verify whether or not the signal generated from an estimated model differs from the true signal. It can be viewed as the difference, or “*fit of error*”, between the true signal and the estimated signal which is given by

$$V(\theta) = E\left[(y(t) - \hat{y}(t,\theta))^2\right] \tag{3.2}$$

where E[•] is the expectation operation.

Since the measured N data points are random variables, the estimated parameter $\hat{\theta}_N$ is also a random variable. To evaluate the fit of error, we shall take the expectation of $V(\hat{\theta}_N)$ with respect to the estimation data, i.e.

$$F_N = E\left[V(\hat{\theta}_N)\right] \tag{3.3}$$

This fit of error will typically rely on the used model structure as well as its model order, the number of data points N, and the properties of the data, such as input spectra, possible feedback, etc., employed for estimating θ and calculating the fit of error. Basically, if the fit of error is small according to our purpose, we have then achieved in modeling the system.

- **Cross Validation**

To guarantee the quality of an estimated model, the another method to confirm such a model should be performed. This is referred to as the *cross validation* method. It is done by splitting up the data into two parts, the estimation data $Z_{est}^{N_1}$ that are used to estimate the model parameters

$$\hat{\theta}_{N_1} = \arg\min_{\theta} V_{N_1}(\theta, Z_{est}^{N_1}) \tag{3.4}$$

and the validation data $Z_{val}^{N_2}$ for which the criterion is evaluated such that

$$\hat{F}_{N_1} = V_{N_2}(\hat{\theta}_{N_1}, Z_{val}^{N_2}) \tag{3.5}$$

where V_N corresponds to the criterion in Eq.(3.2) and $\hat{F}_N$ will be an unbiased estimate of the measure F_N defined by Eq.(3.3). The procedure would then be to try out a number of model structures and choose the one that minimizes $\hat{F}_{N_1}$.

The cross validation approach is somewhat attractive because if the result from Eq. (3.5) is minimized, one has some confidences in the proposed model, irrespective of any probabilistic framework that might be imposed. It is therefore the most normally used method to verify the estimated model. Consequently, we shall primarily evaluate our proposed models based on the cross validation approach.

3.4 Summary

All procedures in system identification technique discussed earlier can be summarized in the flowchart as depicted in Fig.3.2. It is most likely that the first estimated model will not pass the model validation test due to some following deficiencies.

- The selected model structure was not appropriate to characterize the system.
- The data set was not informative enough to provide guidance in choosing good models.
- The numerical procedure failed to find the best model according to criterion used.

- The employed criterion was not good.

One must go back and revise the various steps in the procedures. As a consequence, in order to obtain a good estimated model, one must perform each step in system identification procedures carefully because the result of the current step will affect that of the next step. Clearly, every step in modeling the system is closely related with each other.

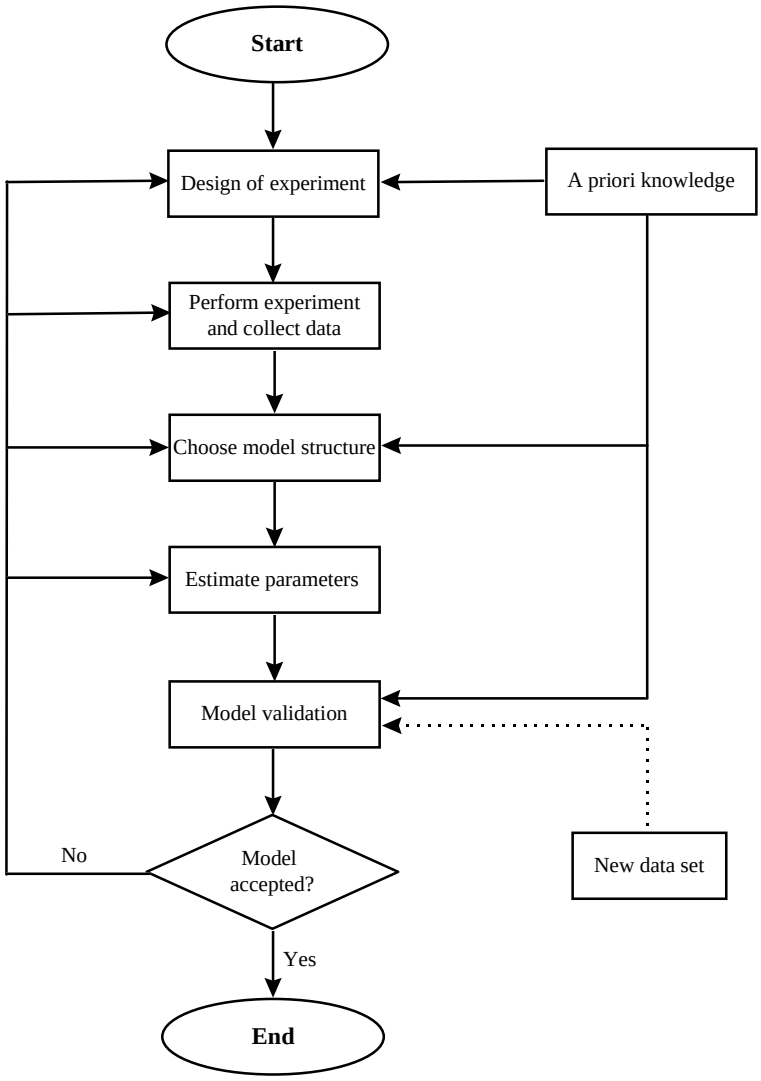

Figure 3.2: System identification procedures [20].

4. Least Squares Method

Given a set of experimental data, the problem of modeling an unknown system is to find a good estimated model and its parameters which can characterize the system's behavior as accurately as possible. Having chosen the best estimated model, the next task is to find the values of its parameters. There are many estimators, which possess nearly all properties of an ideal estimator, leading to good estimated parameters, such as Bayes estimator, Maximum Likelihood estimator, Markov estimator and Least Squares estimator [10]. Bayes estimator is the most powerful estimator, but it, however, requires the most a priori information, e.g. a probability density function (p.d.f.) of the unknown parameters and the p.d.f. of the noise on the measurements. On the contrary, the Least squares estimator can still be used even if there is no a priori information available. In this section, we shall then introduce a least squares method. The statistic properties of an ideal estimator are also given.

4.1 Least Squares Method

The Least Squares (LS) method is normally used to determine the optimal values of the estimated parameters based only on a given set of experimental data. The framework of using LS method [20] can be described as follows.

Assuming that the input-output relation of a stable LTI system is given by

$$y(t) = G^0(z)u(t-nk) + H^0(z)w(t) \tag{4.1}$$

where $G^0(z) = \sum_{k=1}^{\infty} g_k^0 z^{-k}$ is a rational stable transfer function, g_k^0 are the true parameters, z^{-1} is the delay operator, *y(t)* and *u(t)* are known output and input signals, *nk* is the delay, and *w(t)* is white sequences with zero mean and unit variance. We shall denote $v(t) = H^0(z)w(t)$ as unknown noise disturbance.

The estimated output $\hat{y}(t,\theta)$ can be considered as the output of the linear transversal filters of order n with tap weights g_k which can be expressed as

$$\hat{y}(t,\theta) = \sum_{k=1}^{n} g_k \beta_k(z) u(t-nk) \tag{4.2}$$

where g_k is the parameters to be estimated and $\beta_k(z)$ stands for a basis function used in the estimated model; for example, when $\beta_k(z) = z^{-k}$ the Eq.(4.2) corresponds to a FIR model. Typically, a proper choice of basis functions can increase the rate of convergence and reduce the bias and variance of the estimated model. This leads to less number of parameters to be estimated but still guarantee a useful model. Regardless of any choices of basis functions used, the optimal values of estimated parameters can be obtained by the least squares method

Given *N* samples of the observed output and input data, Eq.(4.2) can then be rewritten in the matrix form

$$\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} = \begin{bmatrix} \beta_1(z)u(1-nk) & \beta_2(z)u(1-nk) & \ldots. & \beta_n(z)u(1-nk) \\ \beta_1(z)u(2-nk) & \beta_2(z)u(2-nk) & \ldots. & \beta_n(z)u(2-nk) \\ \vdots & \vdots & \ldots. & \vdots \\ \beta_1(z)u(N-nk) & \beta_2(z)u(N-nk) & \ldots. & \beta_n(z)u(N-nk) \end{bmatrix} * \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

or, equivalently, in the linear regression form

$$\hat{y}(t) = \varphi^T(t)\theta \tag{4.3}$$

where $\varphi(t) = [\ \beta_1(z)u(t\text{-}nk)\ \ \beta_2(z)u(t\text{-}nk)\ \ \ldots\ \ \beta_n(z)u(t\text{-}nk)\ ]^T$, for $t = 1, 2, \ldots, N$, and $\theta = [\ g_1\ g_2\ \ldots g_n\ ]^T$. In general, $\hat{y}(t)$ is called *regressed variable*, the elements in the vector $\varphi(t)$ are known as *regression variables*, and θ is called the *parameter vector* to be estimated. One can also write Eq.(4.3) in matrix notation as

$$\hat{Y} = \Phi\theta$$

where $\hat{Y} = [\ \hat{y}(1)\ \hat{y}(2) \ldots\ \hat{y}(N)\ ]^T$ and $\Phi = [\ \varphi^T(1)\ \ \varphi^T(2)\ \ \ldots \varphi^T(N)\ ]^T$.

A simple way to find θ is to choose the number of measurement, N, equal to n in order to make Φ being a square matrix. So, if the matrix is nonsingular, i.e. its inverse matrix exists and the solution is unique, the linear equation of Eq.(4.3) could easily be solved for θ. Nevertheless, it is a good idea to use a number of observed data, N, greater than n as a consequence of the fact that the estimation should be improved with the additional data. Unfortunately, when $N > n$, Φ becomes overdetermined and the exact solution will not exist. The least squares method is therefore employed to solve this problem. The purpose is to find the estimate of θ which gives the best approximation of Y. Let us consider the one-step ahead prediction error, or *error equation*, which is given by

$$\varepsilon(t,\theta) = y(t) - \sum_{k=1}^{n} g_k \beta_k(z)u(t-nk) \tag{4.4}$$

or in the matrix form

$$E = Y - \Phi\theta \tag{4.5}$$

where the observed output vector $Y = [y(1)\ y(2)\ \ldots\ y(N)]^T$ and the error vector $E = [\varepsilon(1,\theta)\ \varepsilon(2,\theta)\ \ldots\ \varepsilon(N,\theta)\ ]^T$ which is often called *residual*. The *least squares solution* of θ is defined as the vector $\hat{\theta}$ that minimizes the norm of the error (loss function or squared error),

$$V(\theta) = \|E\|^2 = \|Y - \Phi\theta\|^2 \tag{4.6}$$

The geometrical representation of least squares solution is depicted in Fig.4.1. As shown in Fig.4.1, the least squares solution has the property that the error vector, E, is orthogonal to each of the vectors that are used in the approximation for Y, i.e. the column vectors of Φ which is given by $\beta_i(z)\mathbf{u}(t\text{-}nk)$ where $i = 1, 2, \ldots, n$, and $t = 1, 2, \ldots, N$. This orthogonality implies that

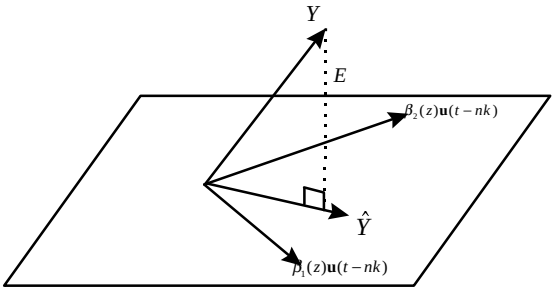

Figure 4.1: Geometrical illustration of the least squares solution to an overdetermined set of linear equation. The best approximation of *Y* is formed when *E* is orthogonal to the vectors $\beta_1(z)\mathbf{u}$(t-nk) and $\beta_2(z)\mathbf{u}$(t-nk).

$$\Phi^H E = 0 \tag{4.7}$$

or, equally,

$$\Phi^H \Phi\theta = \Phi^H Y \tag{4.8}$$

where $(\bullet)^H$ denotes the Hermitian transpose which is defined as $\Phi^H = (\Phi^*)^T = (\Phi^T)^*$. If the column of Φ are linearly independent, i.e. Φ has full rank, then the matrix $\Phi^H\Phi$ is invertible. The least squares solution is then

$$\hat{\theta} = (\Phi^H \Phi)^{-1} \Phi^H Y = \Theta Y \tag{4.9}$$

where the matrix

$$\Theta = (\Phi^H \Phi)^{-1} \Phi^H \tag{4.10}$$

is called the *pseudo-inverse* of the matrix Φ used for the overdetermined problem. Therefore, the best approximation of $\hat{Y}$ to *Y* is given by the projection of the vector *Y* onto the subspace spanned by the vectors $\beta_i\mathbf{u}$(t-nk) where *i = 1, 2, ..., n*, i.e.

$$\hat{Y} = \Phi\hat{\theta} = \Phi(\Phi^H \Phi)^{-1} \Phi^H Y \tag{4.11}$$

Therefore, the minimum value of $V(\theta)$ can be obtained by expanding the square in Eq.(4.6) and using the orthogonality condition given in Eq.(4.7), i.e.

$$V_{\min}(\hat{\theta}) = Y^H E = Y^H Y - Y^H \Phi\hat{\theta} \tag{4.12}$$

The identification of the unknown parameters g_k (for $k=1,2,...,n$) through least squares minimization of $\varepsilon(t,\theta)$ over the time interval is an *identification method* which has some favorable properties. Firstly, it is a linear regression scheme leading to a simple analytical solution. Secondly, it is the type of output error method, which has the advantage that the system $G^0(z)$ can be estimated consistently whenever the unknown noise disturbance *v(t)* is assumed to be uncorrelated with the input signal [12].

4.2 Statistical Properties of the Ideal Estimator

Because any estimates are a function of the observed data, which are random variables, the estimates themselves will then be random variables. As a result, it is essential to examine the performance of an estimator by investigating its statistical properties which can be considered in terms of bias, consistency, efficiency and robustness.

4.2.1 Unbiased and asymptotically unbiased estimator

Since the estimation problem deals with finding a true parameter, θ, from a given set of *N* random variables, the estimate will be a function of *N* random variables. Let $\hat{\theta}_N$ denote an estimate. In general, one would like to have the estimate to be equal, on the average, to the true value. The difference between the true value, θ, and the expected value of the estimate is called *bias* given by

$$B = \theta - E\left\{\hat{\theta}_N\right\} \tag{4.13}$$

where *B* is the bias and $E\{\bullet\}$ denotes the expectation operation. If $B = 0$, the expected value of the estimate will be equal to the true value,

$$E\left\{\hat{\theta}_N\right\} = \theta \tag{4.14}$$

and the estimate is said to be *unbiased*. If $B \neq 0$, $\hat{\theta}_N$ is said to be *biased*. In practice, it could be difficult to fulfill this requirement in some cases. It is then somewhat sufficient to consider the bias in an asymptotic sense, i.e. if an estimate is biased but bias goes to zero as the number of observed data goes to infinity, the estimate is then called *asymptotically unbiased*,

$$\lim_{N\to\infty} E\left\{\hat{\theta}_N\right\} = \theta \tag{4.15}$$

Obviously, it is desired to have an estimator being either unbiased or asymptotically unbiased.

4.2.2 Consistent estimator

To ensure that the estimate will converge to the true value, it is also required that the variance of the estimate should go to zero as the number of observations goes to infinity

$$\lim_{N\to\infty} Var\left\{\hat{\theta}_N\right\} = \lim_{N\to\infty} E\left\{\left|\hat{\theta}_N - E\left\{\hat{\theta}_N\right\}\right|^2\right\} = 0 \tag{4.16}$$

If $\hat{\theta}_N$ is unbiased, $E\left\{\hat{\theta}_N\right\} = \theta$, it follows from Tchebycheff inequality [17] that, for any $\delta > 0$

$$P\left[\left|\hat{\theta}_N - \theta\right| \geq \delta\right] \leq \frac{Var\left\{\hat{\theta}_N\right\}}{\delta^2} \tag{4.17}$$

Consequently, if the variance goes to zero as $N \to \infty$, the probability that $\hat{\theta}_N$ differs by more than δ from the true value will go to zero. In this case, $\hat{\theta}_N$ is said to converge to θ with probability one. Another scheme of convergence is mean-square convergence, i.e. an estimate $\hat{\theta}_N$ is said to converge to θ in the mean-square sense if

$$\lim_{N\to\infty} E\left\{\left|\hat{\theta}_N - \theta\right|^2\right\} = 0 \tag{4.18}$$

No matter which definition is used, the estimator is said to be *consistent* if it is at least asymptotically unbiased and has a variance approaching to zero as N goes to infinity. Note that, a consistent estimator exists which is always either unbiased or asymptotically unbiased, but an unbiased estimator exists which is not necessary to be consistent.

4.2.3 Efficient estimator

Practically, it is of importance not only to have small errors, but also to have small uncertainties on the estimate. That is, a biased estimator with a small uncertainty is sometimes preferable to an unbiased estimator with a large uncertainty. The variations of an estimate resulting from the presence of noise in the system can be described by the covariance matrix. The variations of individual parameters appear on the diagonal, whereas the off-diagonal represents relation between parameter pairs.

Consider two unbiased estimators P_1 and P_2 with mean value P_e and covariance matrices C_{P1} and C_{P2} which is given by

$$C_{P1} = E\left\{(P_1 - P_e)(P_1 - P_e)^T\right\} \tag{4.19}$$

$$C_{P2} = E\left\{(P_2 - P_e)(P_2 - P_e)^T\right\} \tag{4.20}$$

Accordingly, an estimator P_1 is more efficient than an estimator P_2 if $C_{P1} \leq C_{P2}$.

4.2.4 Robust estimator

So far, the estimate has been made based on the assumption that the noise is a stationary white noise process. In some applications, this assumption may not be valid. However, if some properties of an estimate are still valid in such applications, we shall call it a robust estimator. This is crucial in practice because most estimators are used in situations where some usual assumptions are not true. As a result, if it can be proved that an estimator robust with respect to the consistency, then one can guarantee that it will converge to the true value as the number of measurements increases.

5. Linear Model Structures

In this section, a model of the acoustic echo generating system is described. Several traditional linear model structures are introduced.

5.1 Model of the Acoustic Echo Generating System

The echo generating system can be approximately described by the linear model structure. To deal with the problem of estimating the acoustic echo path, we first assume that the echo path is linear and stationary for a short time interval. Although there exist some non-linearity characteristics presented in the system, the assumption of linearity still gives rise to satisfied results depending on an amount of non-linearity of the system. It is difficult to gain a deep insight about the echo generating system for each specific purpose. Consequently, it is suitable to model the system by using a black box model approach which is mainly based on observed input signal *u(t)* and output signal *y(t)*. The task is to estimate the impulse response of the unknown system (echo path), since if the system is linear, its impulse response will then completely represent it.

Supposing that the measured output signal is given by the echo *r(t)* corrupted by the near-end disturbance *x(t)*, i.e.

$$y(t) = r(t) + x(t), \quad \text{where } r(t) = H_e * u(t) \tag{5.1}$$

where $*$ denotes the convolution operation, and H_e is the transfer function of the true echo path. The echo generating system is shown in Fig.5.1.

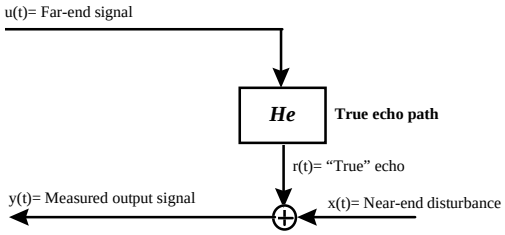

Figure 5.1: The echo generating system.

Since one cannot know the true echo signal, the purpose of modeling is to find out the model that can properly reproduce the true echo signal, given the input signal. It is not possible to exactly obtain the model of true system H_e. Hence, it is appropriate to introduce modeling error, $\varepsilon_m(t)$, in an estimated echo signal, $\hat{r}(t)$, to represent the true echo signal as shown in Fig.5.2. This modeling error can be used to determine how fit the estimated model, *H*, could be. If $\varepsilon_m(t)$ is equal to zero, then we have succeeded in modeling the true echo path.

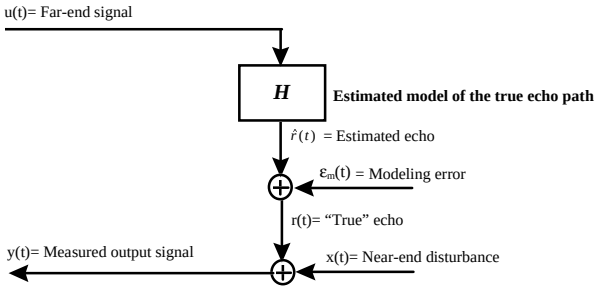

Figure 5.2: An estimated model *H* with modeling error.

5.2 Review of Linear Model Structure

Again, the stable LTI system can be adequately modeled by a linear black box model structure as given in Eq.(3.1). Its graphical representation associated with the echo generating system is depicted in Fig.5.3.

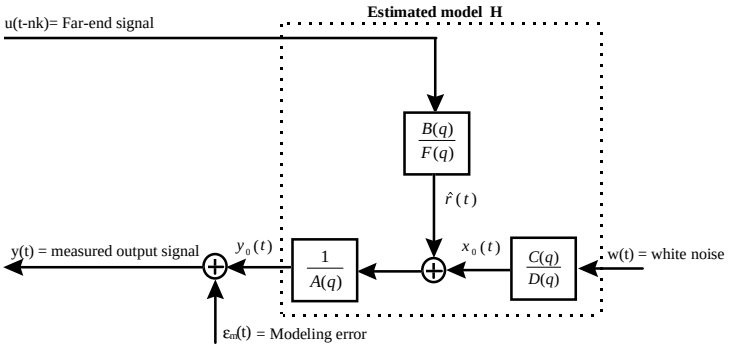

Figure 5.3: A general linear model structure.

where $y_0(t)$ is considered to be the optimum estimated output that can be obtained within a specific model class and a fixed model size. Note that, $A(q)$ represents the poles that are common between the input and the noise. Similarly, $F(q)$ corresponds to the poles that are unique for the input, whereas $D(q)$ determines the poles that are unique for the noise.

The advantage of this model is that the effect of disturbance is also taken into consideration to obtain a more accurate result. That is, since the actual near-end disturbance $x(t)$ is unknown, it is often more convenient to model $x(t)$ as being obtained by filtering a white noise source $w(t)$ through a linear filter $[C(q)/(D(q)A(q))]$. The modeling error $\varepsilon_m(t)$ represents the inability of the proposed model structure to correctly describe the relationship between the input and the output. Herein, three basic linear model structures–FIR, ARX and OE– are presented.

5.2.1 The Finite Impulse Response model structure (FIR)

FIR model is a widely used model structure to describe the system of interest due to its simplicity and stability. It is a special case of Eq.(3.1) when $A(q) = C(q) = D(q) = F(q) = 1$ which can be expressed as

$$
\begin{aligned}
y_0(t) &= B(q)u(t-nk) + w(t) \\
&= \sum_{k=1}^{nb} b_k q^{-k} u(t-nk) + w(t) \\
&= (b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-nb}) u(t-nk) + w(t) \\
&= b_1 u(t-nk-1) + b_2 u(t-nk-2) + \ldots + b_{nb} u(t-nk-nb) + w(t)
\end{aligned}
\tag{5.2}
$$

where the parameters, b_k for $k = 1,2,...,nb$, represent the magnitude of truncated system impulse response, and the signal *w(t)* is assumed to be uncorrelated with the input signal *u(t)*. Fig.5.4 illustrates the FIR model. Observe that, the signal *w(t)* can be perceived as the near-end speaker's speech signal and/or unknown disturbance.

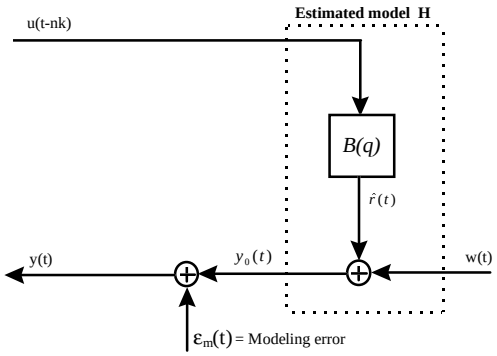

Figure 5.4: The FIR model structure.

In the view of the prediction error, the predicted signals are mainly based on the old inputs. They are usually called the regressors of the model and can be collected in a regression vector, $\varphi(t) = [u(t\text{-}nk\text{-}1)\ u(t\text{-}nk\text{-}2)\ \ldots\ u(t\text{-}nk\text{-}nb)]^T$. The parameter vector to be estimated can be written as $\theta = [b_1\ b_2\ \ldots\ b_{nb}]^T$.

Obviously, FIR model is suitable to represent the echo path because it does not model the signal *w(t)* which is considered as the desired signal in this thesis project. However, it leads to a large number of estimated parameters in order to obtain an accurate approximation.

5.2.2 The Auto Regressive model structure with an eXternal input (ARX)

This model structure can be derived from Eq.(3.1) by setting $C(q) = D(q) = F(q) = 1$. In this case, the poles represented by *A(q)* can be possibly used to model the system with less number of parameters. The output signal consists of an **A**uto **R**egressive part, $A(q)y_0(t)$, and an e**X**ternal input, $B(q)u(t\text{-}nk)$, which is normally called the e**X**ogenerous variable. The mathematical expression can be given by

$$
\begin{aligned}
&A(q) * y_0(t) = B(q)u(t-nk) + w(t) \\
&\sum_{k=0}^{na} a_k y_0(t-k) = \sum_{k=1}^{nb} b_k u(t-nk-k) + w(t) \\
&y_0(t) = -\sum_{k=1}^{na} a_k y_0(t-k) + \sum_{k=1}^{nb} b_k u(t-nk-k) + w(t)
\end{aligned}
\tag{5.3}
$$

Manifestly, the output is a function of the old input and old output. The regression vector is then given by $\varphi(t) = [-y_0(t-1)\ -y_0(t-2)\ ...\ -y_0(t-na)\ u(t-nk-1)\ u(t-nk-2)\ ...\ u(t-nk-nb)]^T$, and the parameter vector is presented by $\theta = [a_1\ a_2\ ...\ a_{na}\ \ b_1\ b_2\ ...\ b_{nb}]^T$. The ARX model is shown in Fig.5.5.

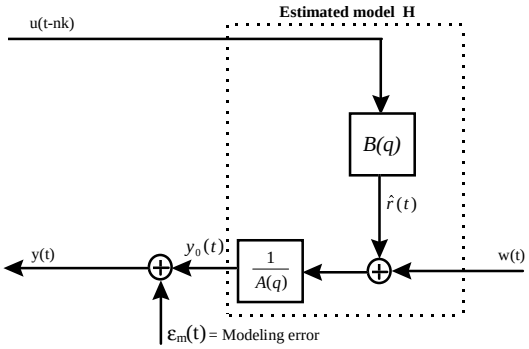

Figure 5.5: The ARX model structure.

Even though one may expect that the number of estimated parameters can be reduced by using ARX model, it is not appropriate for modeling the echo path because the signal *w(t)* will also be modeled by the poles from the polynomial *A(q)*. As a consequence, ARX model will not be considered in this thesis project.

5.2.3 The Output Error model structure (OE)

If the disturbance is assumed to be white sequences in the sense that it is uncorrelated with the input signal, it is therefore sufficient to use the OE model to represent this unknown system. This is because one can parameterize the transfer function independently, and the number of estimated parameters can be reduced when compared with FIR model. The OE model is depicted in Fig.5.6

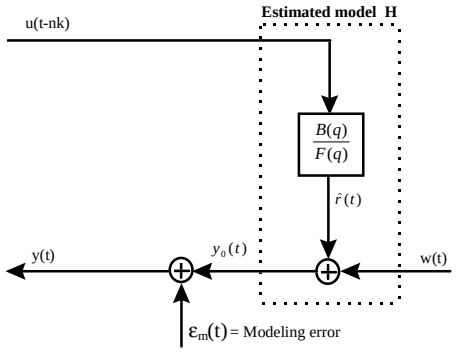

Figure 5.6: The OE model structure.

This model can be realized from Eq.(3.1) when *A(q) = C(q) = D(q) = 1*, which can be expressed as

$$y_0(t) = \hat{r}(t) + w(t) \tag{5.4}$$
$$y_0(t) = \frac{B(q)}{F(q)} * u(t-nk) + w(t)$$

or, equally,

$$\sum_{k=0}^{nf} f_k y_0(t-k) = \sum_{k=1}^{nb} b_k u(t-nk-k) + \sum_{k=0}^{nf} f_k w(t-k) \qquad (5.5)$$

Since $f_0 = 1$, one obtains

$$\left[y_0(t) - w(t)\right] = \sum_{k=1}^{nb} b_k u(t-nk-k) - \sum_{k=1}^{nf} f_k \left[y_0(t-k) - w(t-k)\right] \qquad (5.6)$$

Since $\hat{r}(t) = y_0(t) - w(t)$, the Eq.(5.6) can be rewritten as a regression of measured inputs and estimated outputs,

$$\hat{r}(t) = \sum_{k=1}^{nb} b_k u(t-nk-k) - \sum_{k=1}^{nf} f_k \hat{r}(t-k) \qquad (5.7)$$

Therefore, $\varphi(t)$ = [$u(t\text{-}nk\text{-}1)$ $u(t\text{-}nk\text{-}2)$ ... $u(t\text{-}nk\text{-}nb)$ $-\hat{r}(t-1)$ $-\hat{r}(t-2)$ ... $-\hat{r}(t-nf)$]^T is a regression vector and the parameter vector is defined by θ = [b_1 b_2 ...b_{nb} f_1 f_2 ... f_{nf}]^T.

Clearly, OE model can be an efficient model for modeling the echo path because it can reduce the number of estimated parameters, and it does not model the signal *w(t)*. Nonetheless, one may undergo some difficulties in using OE model for few reasons as follows. Firstly, one cannot absolutely guarantee the stability of the system function in OE model. Secondly, the presence of old predicted output in Eq.(5.7) prevents the use of a simple linear solution to find the parameters b_k and f_k. As a result, we shall not consider the OE model in this thesis project.

6. Orthonormal Bases for System Identification

The traditional approaches, as discussed earlier, for modeling the system of interest lead to the approximation of very high order in case of rapid sampling and/or dispersion in time constant, which is closely related to the dominating pole of a true system. Such a high model order cannot be acceptable in practice due to some difficulties in terms of performance and hardware complexity. By exploiting a priori information about the dominating pole of the system, more appropriate series expansions related to the use of orthonormal basis functions are proposed [4]. These orthonormal functions are constructed by orthonormalizing a given set of exponential functions. They are orthogonal in $L_2(0,\infty)$, and form a complete set in $L_2(0,\infty)$ and $L_1(0,\infty)$. Laguerre and two-parameter Kautz (or, more popularly, just "Kautz") functions are all special cases of orthonormal bases. Laguerre function is suitable for the system with dominant first-order dynamics, whereas Kautz function is appropriate for the system with dominant second-order resonant dynamics. They have indicated that the model order can be substantially reduced when the dominating pole is chosen suitably. In this section, Laguerre and Kautz functions are presented. The Laguerre and Kautz models can be easily implemented with FIR and ARX structures by just replacing a traditional delay operator of these structures with Laguerre and Kautz functions. We shall also introduce the use of the two-stage echo canceller which consists of two stages, i.e. the first stage is a conventional FIR model and the second one is a Laguerre or Kautz model.

6.1 Motivation

Consider a stable LTI system which is given by

$$y(t) = G(z,\theta)u(t-nk) + H(z,\theta)w(t) \tag{6.1}$$

where *y(t)*, *u(t)*, and *nk* are the output, input, and delay respectively. The noise *w(t)* is assumed to be a stationary process with zero mean and unit variance. $G(z,\theta)$ and $H(z,\theta)$ are the set of transfer function parameterized by the parameter vector θ for the input and the noise, where *z* is the unit step delay ($z^{-1}u(t) = u(t-1)$). Given *N* samples of observed data, by *t* we means the time at the sampling instants $t = kT$, for $k = 1,2,...,N$, and *T* is assumed to be equal to 1 for simplicity.

By assuming that $H(z,\theta)$ is minimum phase to guarantee that $H(z,\theta)^{-1}$ is stable, and normalized ($H(\infty,\theta) = 1$), the optimal one-step ahead predictor of *y(t)* is thus defined as

$$\hat{y}(t,\theta) = \left[H(z,\theta)^{-1}G(z,\theta)\right]u(t-nk) + \left[1 - H(z,\theta)^{-1}\right]y(t) \tag{6.2}$$

We shall assume that $H(z,\theta)$ and $G(z,\theta)$ have the same unstable poles. It then follows that

$$H(z,\theta)^{-1}G(z,\theta) = \sum_{k=1}^{\infty} b_k \beta(z) \tag{6.3}$$

$$1 - H(z,\theta)^{-1} = \sum_{k=1}^{\infty} a_k \beta(z) \tag{6.4}$$

where the basis function $\beta_k(z) = z^{-k}$, $|z| \geq 1$, known as the delay operator. For a stable LTI system, it is required that

$$\sum_{k=1}^{\infty} |b_k| < \infty \quad \text{and} \quad \sum_{k=1}^{\infty} |a_k| < \infty \tag{6.5}$$

Practically, since both b_k and a_k tend to zero as $k \to \infty$, we may truncate these expansions at $k = n$ to adequately approximate the system with the finite number of estimated parameters. Hence, Eq.(6.2) can be rewritten in a truncated series expansion as

$$\hat{y}(t,\theta) = \sum_{k=1}^{n} b_k \beta_k(z) u(t-nk) + \sum_{k=1}^{n} a_k \beta_k(z) y(t) \tag{6.6}$$

By taking the parameter vector $\theta = [a_1\ a_2\ \ldots\ a_n\ b_1\ b_2\ \ldots\ b_n]^T$, the well-known ARX model structure is derived. Setting $a_k = 0$, for all k, leads to FIR model structure, or by taking $b_k = 0$, for all k, AR model structure is realized. In theory, the usefulness of the estimate is limited by how fast the sums in Eq.(6.3) and Eq.(6.4) converge, i.e., the rate of the error terms

$$\sum_{k=n+1}^{\infty} |b_k| \quad \text{and} \quad \sum_{k=n+1}^{\infty} |a_k| \tag{6.7}$$

tend to zero as $n \to \infty$. The rate of convergence is basically determined by the location of the poles of $G(z,\theta)$ and the zeros of $H(z,\theta)$. Poles and zeros close to unit circle imply a slow rate of convergence.

If $G(z,\theta)$ and $H(z,\theta)$ are obtained by sampling a continuous time system, using a sampling interval T, the continuous time poles and zeros, λ, are approximately mapped to the discrete time poles and zeros at $\{e^{\lambda T}\}$ for small T. Since most digital applications require a high sampling rate, i.e. $T \to 0$, one could get a serious problem of estimating this system due to a very slow rate of convergence because the discrete time poles and zeros approach one. Additionally, because the variance of an estimated model is proportional to the number of estimated parameters, it is advantageous to use as few parameters as possible but still guarantee an useful model. These problems have motivated to use an alternative operator which is less sensitive to the location of poles and zeros. This is a consequence of the fact that the delay operator has too short memory (only one sampling step). By introducing the operator with longer memory, the number of estimated parameters necessary to obtain an accurate approximation can then be reduced. This operator can be chosen as

$$\beta_k(z) = \frac{1}{z - \xi_k} = z^{-1} + \xi_k z^{-2} + \xi_k^2 z^{-3} + \xi_k^3 z^{-4} + \ldots \tag{6.8}$$

where the poles ξ_k are chosen according to a priori knowledge of the true system. It is desirable to construct the orthonormal basis function based on this operator. The procedure of a unifying construction of orthonormal bases can be seen in [8] and the result is given by

$$\beta_k(z) = \left(\frac{\sqrt{1-|\xi_k|^2}}{z-\xi_k}\right)\prod_{j=1}^{k-1}\left(\frac{1-\overline{\xi_j}z}{z-\xi_j}\right), \quad \text{for } k \geq 1 \tag{6.9}$$

where a variety of poles at $\{\xi_1, \xi_2, ..., \xi_k\}$ are incorporated. These basis functions are orthonormal with respect to the following inner product [8]

$$\left\langle \beta_i(z), \beta_j(z) \right\rangle = \frac{1}{2\pi j}\oint_T \beta_i(z)\overline{\beta_j(z)}\frac{dz}{z} = 0, \quad \text{for } i \neq j \tag{6.10}$$

and have unit normalization

$$\left\|\beta_k(z)\right\|^2 = 1, \quad \text{for } \forall k \tag{6.11}$$

Several advantages [3] are attained from employing the orthonormal basis functions in system identification problem. Firstly, they have transforms that are rational functions with a very simple repetitive form. This allows their practical realization with concatenated blocks as illustrated in Fig.6.1. Secondly, the solution of transfer function estimation problems leads to the normal equation having a diagonal structure. However, it holds only if the input is white sequences. Furthermore, if the normal equation has a Toeplitz structure, much is gained in terms of numerical algorithm, sensitivity, etc. Finally, as seen in Eq.(6.9), the orthonormal basis function consists of a first order low-pass term and *k-1* all-pass terms. Such all-pass filters are favorable in terms of numerical sensitivity, and they are often recommended to use in filter design.

FIR, Laguerre and Kautz functions are all special cases of orthonormal basis functions. Manifestly from Eq.(6.9), FIR function is realized by choosing $\xi_k = 0$ for all *k*. Laguerre function is obtained when $\xi_k = \xi$, where ξ is real-valued and $|\xi| < 1$, for all *k*, while Kautz function is generalized when the pole is chosen such that $\xi_k = \xi$, where ξ is complex-valued and $|\xi| < 1$, for all *k*.

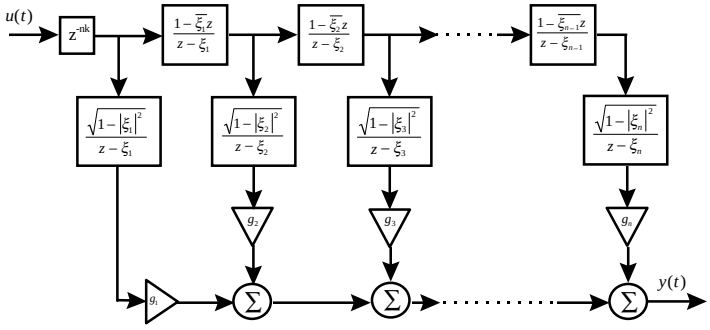

Figure 6.1: Discrete-time Orthonormal networks of order *n*

6.2 System Representation

The problem of modeling an unknown system from a given set of observed input and output is of importance in many fields of science. Suppose that the input-output relation of a stable LTI system is given by

$$y(t) = G^0(z)u(t-nk) + v(t), \qquad G^0(z) = \sum_{k=1}^{\infty} g_k^0 z^{-k} \tag{6.12}$$

where *y(t)*, *u(t)*, and *nk*, are the output, input, and delay respectively. The time shifts are represented by the delay operator $z^{-1}u(t) = u(t-1)$. The complex-valued function $G^0(z)$, where $z \in \mathbf{C}$, is the system transfer function. For stable systems $\sum_{k=1}^{\infty} \left| g_k^0 \right| < \infty$, it implies that $G^0(z)$ is analytic in $|z| > 1$ and continuous in $|z| \geq 1$. The noise *v(t)* is assumed to be a stationary process with zero mean and unit variance.

System identification deals with the problem of finding the estimate of $G^0(z)$ from the experimental data of $\{y(t), u(t)\}$, where $t = 1, 2, ..., N$. Theoretically, $G^0(z)$ can be represented by an infinite number of given basis functions. In practice, a truncated series expansion, say the n^{th} order, is used to estimate $G^0(z)$ with the result that there is a truncation error. This error can be minimized by a proper choice of basis function. In addition, the identification problem can be simplified to a linear regression estimation problem if the model structure is a priori linear in the parameters, i.e., if the model can be expressed as

$$G(z) = \sum_{k=1}^{n} g_k \beta_k(z) \tag{6.13}$$

where $\beta_k(z)$ is a set of given basis functions and g_k are the parameters to be estimated. The least squares estimation method is then applied to find the optimal values of the model parameters of lower order terms. Since the proper choice of basis function, $\beta_k(z)$, will give rise to a considerable increase in the rate of convergence and a significant decrease in the asymptotic variance of the estimated transfer function, this leads to accurate approximation with less number of estimated parameters. Herein, we shall investigate only two special cases of orthonormal basis functions, namely Laguerre and (two-parameter) Kautz functions.

6.3 Discrete Laguerre Function

The Laguerre function has been extensively studied in [4]. This function can be obtained by the Gram-Schmidt orthogonalization process of the following sequences

$$f_k(t,\xi) = t^k \xi^t, \quad \text{for } t=0, 1, 2, 3, ... \text{ and } k=1, 2, 3, ... \tag{6.14}$$

where $|\xi| < 1$ is called the (real-valued) Laguerre pole. Denote $l_k(t,\xi)$ the k^{th} discrete Laguerre function, thus

$$l_k(t,\xi) = (1-\xi^2)^{1/2} \sum_{l=0}^{k-1} \binom{k-1}{l} \binom{l+t-1}{k-1} (-1)^l \xi^{2l+t-k}, \quad j,k \geq 1 \tag{6.15}$$

This set of function is orthonormal, i.e.

$$\sum_{t=1}^{\infty} l_i(t,\xi) l_j(t,\xi) = \delta_{ij} \tag{6.16}$$

where δ_{ij} is the Kronecker delta function. The discrete Laguerre functions have *z*-transform

$$L_k(z,\xi) = \frac{\sqrt{1-\xi^2}}{z-\xi}\left(\frac{1-\xi z}{z-\xi}\right)^{k-1}, \quad \text{for } k \geq 1 \tag{6.17}$$

and form a complete orthonormal set in L_2 norm. Additionally, it can also be obtained from Eq.(6.9) by setting ξ_k = (real-valued)ξ for all k. Owing to the completeness of the Laguerre functions, any real sequences belonging to L_2 norm can be put in the form

$$G(z) = \sum_{k=1}^{n} g_k L_k(z,\xi), \quad \text{for } |z| \geq 1 \tag{6.18}$$

where the Laguerre parameters can be obtained by

$$g_k = \langle G(z), L_k(z,\xi) \rangle = \frac{1}{2\pi j}\oint_{\Gamma} G(z) L_k(z^{-1},\xi)\frac{dz}{z} \tag{6.19}$$

and Γ is a circle of radius higher than 1 and lower than $1/|\xi|$. The estimated output can then be written as

$$\hat{y}(t) = \left(\sum_{k=1}^{n} g_k L_k(z,\xi)\right) u(t-nk) \tag{6.20}$$

The realization of using discrete Laguerre functions has a structure sketched in Fig.6.2.

Even if the performance of Laguerre model is much more better than that of FIR, it still has some drawbacks in case of scattered dominating poles and resonant poles. Therefore, the Laguerre model is extended and generalized to Kautz model which can cope with several different possible complex poles.

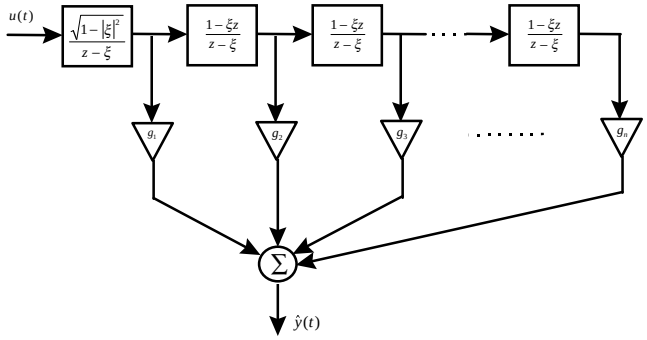

Figure 6.2: Discrete-time Laguerre network of order *n*. Note that the first block is a low-pass filter and all other blocks are all-pass filters.

6.4 Discrete Kautz Function

The problem in system identification using Kautz function has been studied in [5]. Since the system in real applications is much more complicated than modeling it by using one real-valued pole, it is necessary to use a complex-valued pole to characterize the resonant part of the system. To incorporate a complex pole, it is essential to recast the formulation in Eq.

(6.9), since as soon as one pole, say ξ_j is chosen as a complex pole, then the impulse responses for the $\beta_k(z)$ for $k > j$ become complex, and this is physically unacceptable in a system identification problem. To avoid this problem, one needs to include its complex-conjugated pole automatically whenever one complex-valued pole is chosen in Kautz basis function. The detail of deriving Kautz basis function can be seen in [8].

Given the n^{th} model order, the truncated Kautz series expansion can be written as

$$G(z) = \sum_{k=1}^{n/2} \left\{ g_{2k-1} \Psi_{2k-1}(z) + g_{2k} \Psi_{2k}(z) \right\} \tag{6.21}$$

where the Kautz basis function is given by

$$\Psi_{2k-1}(z) = K_k^{(1)} \prod_{i=1}^{k-1} \Lambda_i(z) \tag{6.22}$$

$$\Psi_{2k}(z) = K_k^{(2)} \prod_{i=1}^{k-1} \Lambda_i(z) \tag{6.23}$$

where $K_k^{(1)} = \dfrac{\sqrt{(1-\gamma_k^2)}(z-\alpha_k)}{z^2 - \alpha_k(\gamma_k+1)z + \gamma_k}$ and $K_k^{(2)} = \dfrac{\sqrt{(1-\gamma_k^2)(1-\alpha_k^2)}}{z^2 - \alpha_k(\gamma_k+1)z + \gamma_k}$, (6.24)

$$\Lambda_i(z) = \left(\frac{\gamma_i z^2 - \alpha_i(\gamma_i+1)z + 1}{z^2 - \alpha_i(\gamma_i+1)z + \gamma_i} \right), \text{ for } i = 1, 2, ..., k\text{-}1, \tag{6.25}$$

$$\alpha_k = \frac{\xi_k + \bar{\xi}_k}{1 + |\xi_k|^2} \text{ and } \gamma_k = |\xi_k|^2, \text{ for which } |\alpha_k| < 1 \text{ and } |\gamma_k| < 1, \tag{6.26}$$

and ξ_k is a complex-valued pole. In this thesis project, we shall consider only two-parameter Kautz functions where ξ_k is chosen equal to (complex-valued) ξ, for all k. On account of an orthonormal property, the Kautz parameters $\{g_{2k-1}, g_{2k}\}$ can be determined by the projection on the basis functions, i.e. $g_{2k-1} = \langle G(z), \Psi_{2k-1}(z) \rangle$ and $g_{2k} = \langle G(z), \Psi_{2k}(z) \rangle$. The discrete Kautz network can be simply represented in concatenated blocks as illustrated in Fig.6.3. For higher order properties such as convergence of generalized covariance functions, error bound, consistency, asymptotic variance, etc., one can read in [6].

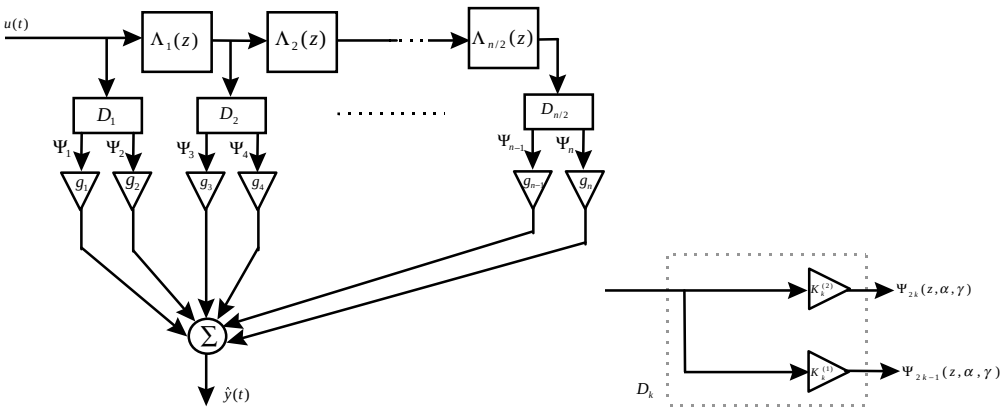

Figure 6.3: Discrete Kautz Network of order n. The section D_k are sketched separately in the right picture.

6.5 Implementation of Orthonormal Basis function

It is simply to construct the orthonormal model with FIR and ARX structure. The idea is just to replace the traditional delay operator with the corresponding orthonormal basis function. The construction can be shown as follows.

6.5.1 Orthonormal model with FIR structure

FIR structure is a widely used model structure to represent the system because it is easy to implement and inherently stable. The general form of FIR model is

$$y(t) = G(z) * u(t - nk) + v(t) \tag{6.27}$$

where $G(z) = \sum_{k=1}^{nb} b_k z^{-k}$ and the noise *v(t)* is assumed to be white sequences. By replacing *G (z)* by $G(z,\theta) = \sum_{k=1}^{nb} g_k \beta_k(z)$, the Laguerre model is realized if $\beta_k(z)$ corresponds to Eq. (6.17), and the Kautz model is derived if $\beta_k(z)$ is equal to Eq.(6.22) - Eq.(6.23). The estimated output can then be written as

$$\hat{y}(t) = \left(\sum_{k=1}^{nb} g_k \beta_k(z) \right) u(t - nk) \tag{6.28}$$

Given *N* samples of the observed output and input data, Eq.(6.28) can be rewritten in matrix form

$$\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} = \begin{bmatrix} \beta_1(z)u(1-nk) & \beta_2(z)u(1-nk) & \ldots. & \beta_{nb}(z)u(1-nk) \\ \beta_1(z)u(2-nk) & \beta_2(z)u(2-nk) & \ldots. & \beta_{nb}(z)u(2-nk) \\ \vdots & \vdots & \ldots. & \vdots \\ \beta_1(z)u(N-nk) & \beta_2(z)u(N-nk) & \ldots. & \beta_{nb}(z)u(N-nk) \end{bmatrix} * \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_{nb} \end{bmatrix}$$

Thereby, the regression vector is given by $\varphi(t) = [\ \beta_1(z)u(t\text{-}nk)\ \beta_2(z)u(t\text{-}nk)\ \ldots\ \beta_{nb}(z)u(t\text{-}nk)\ ]^T$, for $t = 1, 2, \ldots, N$, and the parameter vector is represented by $\theta = [g_1\ g_2\ \ldots\ g_{nb}]^T$. The parameter vector can be solved by means of an off-line method in the least squares sense as discussed earlier.

6.5.2 Orthonormal model with ARX structure

Although the ARX model is not considered in this thesis project, it is noteworthy to point out that the realization of orthonormal model with ARX structure is similar to that with FIR structure except having some terms contributed from the output signal. Let *na* and *nb* be a number of output and input parameters to be estimated respectively, thereby the estimated output is given by

$$\hat{y}(t) = -\sum_{k=1}^{na} a_k \beta_k(z) y(t) + \sum_{k=1}^{nb} b_k \beta_k(z) u(t - nk) \tag{6.29}$$

which can be written in matrix form

$$\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} = \begin{bmatrix} -\beta_1(z)y(1) & \ldots. -\beta_{na}(z)y(1) & \beta_1(z)u(1-nk) & \ldots. & \beta_{nb}(z)u(1-nk) \\ -\beta_1(z)y(2) & \ldots. -\beta_{na}(z)y(2) & \beta_1(z)u(2-nk) & \ldots. & \beta_{nb}(z)u(2-nk) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\beta_1(z)y(N) & \ldots. -\beta_{na}(z)y(N) & \beta_1(z)u(N-nk) & \ldots. & \beta_{nb}(z)u(N-nk) \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ b_{nb} \end{bmatrix}$$

In this case, the regression vector is determined by $\varphi(t) = [-\beta_1(z)y(t) \ -\beta_2(z)y(t) \ldots -\beta_{na}(z)y(t) \ \beta_1(z)u(t-nk) \ \beta_2(z)u(t-nk) \ldots \beta_{nb}(z)u(t-nk)]^T$ and the parameter vector is defined by $\theta = [a_1 \ a_2 \ldots a_{na} \ b_1 \ b_2 \ldots \ b_{nb}]^T$. The solution to find the parameters, a_i ($i=1,2,...,na$) and b_j ($j=1,2,...nb$) is therefore identical to that of orthonormal model with FIR structure.

6.6 Two-Stage Echo Canceller

The typical echo impulse response can be decomposed into 2 parts, namely the first part which normally has a rapid time variation, and the second part known as the tail of the impulse response which is slowly decaying towards zero. By exploiting the characteristics of such an echo impulse response, an approach of using a two-stage echo canceller is introduced [9]. Its purpose is to reduce the number of estimated parameters as well as the computational complexity of an echo canceller algorithm, particularly in case of the echo response with a long tail. This can be done by dividing the structure of an echo canceller into two stages. The first stage is a conventional transversal FIR filter that spans for the first few parameters of the echo impulse response, while the second stage approximates the remainder or tail of the response by a linear combination of orthonormal functions. In other words, the tail may be well approximated by combining a few parameters, whereas a conventional transversal FIR filter representation will involve far more parameters. As a result, if the number of estimated parameters in an orthonormal model is not large, a substantial reduction in the model order and the computational complexity can be achieved.

Fig. 6.4 shows the two-stage echo canceller which is divided into two parts. The first is a conventional transversal FIR filter with *nb* tap parameters, b_k for $k = 1,2,...,nb$, which approximate the first *nb* of the first part of the echo impulse response. The second is an orthonormal model with *na* parameters, g_k for $k = 1,2,...,na$, which are used to approximate the tail portion of the echo response. Note that, the orthonormal model can be either Laguerre model or Kautz model.

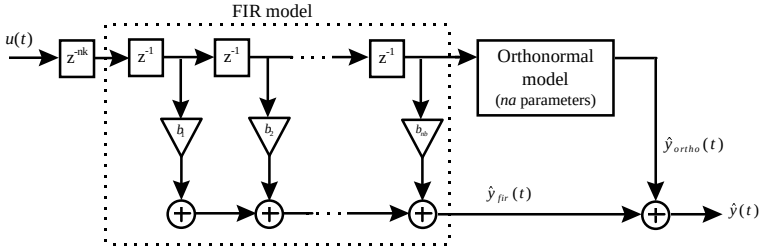

Figure 6.4: The two-stage echo canceller

The estimated output is therefore given by

$$\hat{y}(t) = \underbrace{\left(\sum_{k=1}^{nb} b_k z^{-k}\right) u(t-nk)}_{\hat{y}_{fir}(t)} + \underbrace{\left(\sum_{k=1}^{na} g_k \beta_k(z)\right) u(t-nk-nb)}_{\hat{y}_{ortho}(t)} \qquad (6.30)$$

where $\beta_k(z)$ are the orthonormal basis functions, for *k = 1, 2, ..., na*, corresponding to Eq. (6.17) for Laguerre function and to Eq.(6.22)-Eq.(6.23) for Kautz function. The Eq.(6.30) can also be expressed in matrix form,

$$\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix} = \begin{bmatrix} u(1-nk)z^{-1} & \cdots & u(1-nk)z^{-nb} & \beta_1(z)u(1-nk-nb) & \cdots & \beta_{na}(z)u(1-nk-nb) \\ u(2-nk)z^{-1} & \cdots & u(2-nk)z^{-nb} & \beta_1(z)u(2-nk-nb) & \cdots & \beta_{na}(z)u(2-nk-nb) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u(N-nk)z^{-1} & \cdots & u(N-nk)z^{-nb} & \beta_1(z)u(N-nk-nb) & \cdots & \beta_{na}(z)u(N-nk-nb) \end{bmatrix} * \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ g_{na} \end{bmatrix}$$

Consequently, the regression vector is given by φ(t) = [*u(t-nk-1) ... u(t-nk-nb)* $\beta_1(z)u(t-nk-nb)$ *...* $\beta_{na}(z)u(t-nk-nb)$]^T, and the parameter vector is θ = [b_1 *...* b_{nb} g_1 *...* g_{na}]^T which can also be solved by the least squares method.

7. Optimum Choice of Dominating Pole

Any stable LTI system can be modeled by an infinite series of orthonormal functions which involve a free parameter, closely related to the dominating pole. Theoretically, when infinitely many parameters are employed in the expansion, the choice of a dominating pole is somewhat arbitrary. In practice, however, a truncated series expansion is used and results in the truncation error. This error is basically a function of the model order and its dominating pole. For a fixed model order, there exists an optimal dominating pole that minimizes the truncation error. Hence, the choice of a dominating pole plays an important role in the quality of an approximation of the system by Laguerre and Kautz models. The selection of a good dominating pole is then a crucial problem. If this parameter is chosen appropriately, the Laguerre and Kautz models can efficiently approximate a large class of linear system with less number of estimated parameters when compared with FIR model. Three approaches are presented here to determine the optimal dominating pole. Each approach has its own advantage in the sense of complexity and performance. Finally, a remarkable suggestion on model reduction by means of the orthonormal model is also given.

7.1 Based on a Given Set of Impulse Responses

This method was firstly introduced in [23] which requires the knowledge of the impulse responses of the system. It is used merely for Laguerre model where the dominating pole is restricted to be real-valued. The key idea can be summarized as follows.

In theory, any stable LTI system (which usually contains at least one unit delay) can be exactly represented by an infinite sum of discrete Laguerre function which is given by

$$G(z) = \sum_{k=1}^{\infty} g_k L_k(z, \xi) \tag{7.1}$$

where $L_k(z,\xi)$ is given in Eq.(6.17), g_k is the Laguerre parameters to be estimated, and ξ is called the Laguerre pole where $|\xi| < 1$. The corresponding time domain signal $l_k(t,\xi)$ is the inverse z-transform of $L_k(z,\xi)$. Since each Laguerre function is realizable as the unit impulse response of a stable LTI system, one can express such a unit impulse response, $h(t)$, in terms of Laguerre sequences as

$$h(t) = \sum_{k=1}^{\infty} g_k l_k(t, \xi) \tag{7.2}$$

Nevertheless, the truncated Laguerre series expansion is employed in practice to model a system, and the least squares estimation method will be able to obtain the suitable values of Laguerre parameters. Herein, we shall therefore introduce the optimal Laguerre pole, ξ, which minimizes the following *performance index*,

$$J = \sum_{k=1}^{\infty} k g_k^2 \tag{7.3}$$

This performance index is selected because it linearly increases the weight of each additional Laguerre parameter. To obtain a fast convergence rate in Eq.(7.3), the rate showing that the coefficients of higher order Laguerre functions go towards zero, the minimum of performance index must be achieved since $\sum_{k=1}^{\infty} g_k^2 = \sum_{t=0}^{\infty} h^2(t)$ is a constant for a given system.

Given the unit impulse response, *h(t)*, of a stable LTI system, we define

$$M_1 = \frac{1}{\|h\|^2} \sum_{t=0}^{\infty} t h^2(t) \quad \text{and} \quad M_2 = \frac{1}{\|h\|^2} \sum_{t=0}^{\infty} t\left[\Delta h(t)\right]^2 \tag{7.4}$$

where $$\|h\|^2 = \sum_{t=0}^{\infty} h^2(t) = \sum_{k=1}^{\infty} g_k^2 \tag{7.5}$$

$$\sum_{t=0}^{\infty} t\left[\Delta h(t)\right]^2 = \sum_{t=0}^{\infty} \left[\Delta h(t)\right]\left[t\Delta h(t)\right], \text{ for } \Delta h(t) = h(t+1) - h(t) \tag{7.6}$$

Here, M_1, M_2, and $\| h \|$ depend on a given system. Observe that, M_1 and M_2 essentially characterize the rate of decay and the smoothness of the impulse response of the system, respectively. Therefore, we choose the Laguerre pole ξ such that the performance index [23]

$$J = \sum_{k=1}^{\infty} k g_k^2 = \frac{\xi M_2 + (1-\xi)^2 M_1 + \xi - \xi^2}{1-\xi^2} \|h\|^2 \tag{7.7}$$

is minimized. Thus we set the derivative of Eq.(7.7) with respect to ξ equal to zero, to obtain

$$\xi^2 (M_1 + J - 1) + \xi (M_2 - 2M_1 + 1) + (M_1 - J) = 0 \tag{7.8}$$

For real-valued ξ, it follows that

$$(M_2 - 2M_1 + 1)^2 - 4(M_1 + J - 1)(M_1 - J) \geq 0 \tag{7.9}$$

The minimum performance index is

$$J_{\min} = \frac{1 + \sqrt{4M_1 M_2 - M_2^2 - 2M_2}}{2} \tag{7.10}$$

and the corresponding optimal Laguerre pole is given by

$$\xi_{opt} = \frac{2M_1 - 1 - M_2}{2M_1 - 1 + \sqrt{4M_1 M_2 - M_2^2 - 2M_2}} \tag{7.11}$$

For any stable LTI system, it can be shown that $(4M_1 - M_2 - 2)\|h\|^2 = \sum_{t=0}^{\infty} t[h(t)+h(t+1)]^2$ which can not be negative, accordingly, $4M_1M_2 - M_2^2 - 2M_2 \geq 0$ is always true. Therefore, the optimum Laguerre pole given in Eq.(7.11) is definitely a real value. See in [23] for proofs.

If there is a time delay in the continuous system, the sampled system will have more than one unit delay. The values of both M_1 and M_2 will then be affected by this delay. For the system containing the delay d ($d > 1$), we have to recast the solution in accordance with this delay. Let us denote M_1 and M_2 by $M_1^{(d)}$ and $M_2^{(d)}$, respectively, thus

$$M_1^{(d)} = M_1^{(1)} + (d-1) \tag{7.12}$$

$$M_2^{(d)} = M_2^{(1)} + 2(d-1)(1-D_3) \tag{7.13}$$

where D_3 is independent of the delay d and is defined by:

$$D_3 = \frac{\sum_{t=0}^{\infty} h(t)h(t+1)}{\|h\|^2} \tag{7.14}$$

When the delay is d, denote $D_1 = M_1^{(1)}$ and $D_2 = M_2^{(1)}$, the optimal Laguerre pole ζ_{opt} is also derived from Eq.(7.11) where

$$M_1 = D_1 + (d-1) \tag{7.15}$$

$$M_2 = D_2 + 2(d-1)(1-D_3) \tag{7.16}$$

Note that D_1, D_2 and D_3 depend only on the delay-free system.

Clearly, the optimal Laguerre pole depends on characteristics of the system impulse response, such as its rate of decay, its smoothness and the time delay. Note that, this method does not consider about how large the model order should be in order to obtain an acceptable approximation of the true system. This means that the optimal Laguerre pole deriving from this method is just an approximation of the true dominating pole in asymptotic sense because it is calculated from the whole set of given impulse responses. Then the optimal Laguerre pole is primarily dependent of how good the given impulse response is. Furthermore, since this optimum Laguerre pole will be used to evaluate the fit of error at every model order, one can expect that the fit of error of Laguerre model with this pole may be worse than that of FIR model at some model orders.

7.2 Based on Minimizing the Loss Function

Again, let the input-output relation of an estimated model of order n be represented by

$$\hat{y}(t) = \left(\sum_{k=1}^{n} g_k \beta_k(z) \right) u(t-nk) = \varphi^T(t)\theta \tag{7.17}$$

where $\beta_k(z)$ corresponds to a set of chosen basis function, i.e. Laguerre or Kautz function, $\varphi(t) = [\beta_1(z)u(t-nk) \ \beta_2(z)u(-nk) \ ... \ \beta_n(z)u(t-nk)]^T$ and $\theta = [g_1 \ g_2 \ ... \ g_n]^T$. Consequently, the least

squares solution of θ is defined as the parameter $\hat{\theta}$ that by minimizing the loss function as given in Eq.(4.6), i.e.

$$v(\theta) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \varphi^T(t)\theta\right)^2 \tag{7.18}$$

Now let us introduce the following notation:

$$\hat{d}_j(\xi) = \frac{1}{N}\sum\nolimits_{t=1}^{N} y(t)\beta_j u(t-nk), \text{for } 1 \le j \le n \tag{7.19}$$

$$\hat{\gamma}_n(\xi) = (\hat{d}_1(\xi), \hat{d}_2(\xi), \ldots\ldots, \hat{d}_n(\xi))^T \tag{7.20}$$

$$\hat{\Gamma}_n(\xi) = \frac{1}{N}\sum\nolimits_{t=1}^{N} \varphi(t)\varphi^T(t) \tag{7.21}$$

Given a stable LTI system, the loss function will be a function of the model order and its dominating pole. For a fixed model order, there exists an optimal dominating pole that minimizes the loss function. The optimal dominating pole is therefore chosen as the one that minimizes the Eq.(7.18), which is simply equivalent to [2]

$$\xi_{opt} = \arg\min_{\xi}\left[\frac{1}{N}\sum_{t=1}^{N} y^2(t) - \hat{\gamma}_n(\xi)^T \hat{\Gamma}_n(\xi)^{-1}\hat{\gamma}_n(\xi)\right] \tag{7.22}$$

or, equally,

$$\xi_{opt} = \arg\max_{\xi}\left[\hat{\gamma}_n(\xi)^T \hat{\Gamma}_n(\xi)^{-1}\hat{\gamma}_n(\xi)\right] \tag{7.23}$$

The solution in Eq.(7.23) can be either a real value or a complex value. Accordingly, this method will be applied to find the optimal dominating pole for both Laguerre and Kautz models. The resulting optimal dominating pole from this method is more effective than the first method given in section 7.1 because it is derived by minimizing the Eq.(7.18) at each model order.

Nevertheless, the drawback of this method is that we need to use the search method to find the optimal dominating pole at each model order. It is done by varying the value of ξ within unit circle in order to retain the stability of a model, and choosing the pole that yields the maximum value in Eq.(7.23). Since the matrix size of each element in Eq.(7.23) will be proportional to the number of estimated parameters in orthonormal model, if one is required to find the optimal dominating pole at very high order, it will take long time for solving Eq. (7.23). Therefore, such wasted time can be viewed as the computational complexity, i.e. the larger the matrix size, the longer the wasted time, and then the higher the computational complexity. Evidently, this method requires much higher computational complexity than the first method. As a result, to avoid the computational complexity of calculating the optimal dominating pole at very high order, one need to subdivide the whole interval of the pole into smaller subintervals of more manageable size.

7.3 Based on the Derivative of the Squared Error Equation

For a fixed model order, the squared error (SE) equation given in Eq.(4.4) will be a function of a dominating pole. Herein, we shall derive an optimal dominating pole for both Laguerre model and Kautz model by setting the derivative of their SE equation with respect to the dominating pole equal to zero. In general, the methods for deriving the optimal dominating pole for Laguerre and Kautz functions are in essence identical. More details about the optimality condition of Laguerre model can be read in [15], [16], [18], [21] and [22], while that of Kautz model can be seen in [1].

7.3.1 Optimality condition of discrete Laguerre model

Since the impulse response of any stable LTI system can be represented by a sum of exponential terms (time polynomials), e.g. $h^0(t) = \sum_{i=1}^{\infty} A_i \xi_i^t$ where $A_i>0$ and $0<\xi_i<1$ are real numbers, it seems reasonable to use basis functions of the same type to model this impulse response. The main problem of an approximation is a suitable choice of the exponent of the exponential, which is referred to as a dominating pole, because the squared error (SE) surface of an approximation is a function of this parameter.

Suppose the input-output relation of the true system is given by

$$y(t) = \sum_{m=1}^{\infty} h^0(m) u(t-m) \tag{7.24}$$

where $y(t)$ and $u(t)$ are output and input sequences, and $h^0(t)$ is the true unit sample response. Consider the case where the unit sample response $h^0(t)$ is represented by a set of orthonormal sequences $\{\phi_k(t)\}$ such that

$$h^0(t) = \sum_{k=1}^{\infty} g_k \phi_k(t) \tag{7.25}$$

where

$$g_k = \sum_{t=1}^{\infty} h^0(t) \phi_k(t) \tag{7.26}$$

is the k^{th} coefficient of the series expansion. Herein, we shall consider only the case of Laguerre model where the sequences $\{\phi_k(t)\}$ are replaced by the Laguerre sequences $\{l_k(t,\xi)\}$ as given in Eq.(6.15).

However, the truncated Laguerre series expansion, say the n^{th} order Laguerre model, is used in practice with a result that there is a truncation error. An approximation of the true unit sample response $h^0(t)$ can then be expressed as

$$h(t) = \sum_{k=1}^{n} g_k l_k(t,\xi) \approx h^0(t) \tag{7.27}$$

If this approximation turns into an equality, the function $h^0(t)$ is said to be within the model set. Due to the orthonormal property, the squared error (SE) can be defined as the squared

value of a discrepancy between the true impulse response $h^0(t)$ and the estimated impulse response $h(t)$ which can be written as

$$\varepsilon(t,\xi) = \sum_{t=1}^{\infty}\left[h^0(t) - h(t)\right]^2 \tag{7.28}$$

Since the Laguerre sequences $\{l_k(t,\xi)\}$ are orthonormal, it is straightforward to show that Eq. (7.28) is equivalent to

$$\varepsilon(t,\xi) = \sum_{t=1}^{\infty}\left[h^0(t)\right]^2 - \sum_{k=1}^{n} g_k^2 \tag{7.29}$$

For a given stable LTI system, the SE will be a function of the model order and its dominating pole. Hence, for a fixed model order, there exists the optimal Laguerre pole that minimizes $\varepsilon(t,\xi)$ or maximizes $\sum_{k=1}^{n} g_k^2$, i.e. the optimal Laguerre pole is defined as

$$\xi = \arg\min_{\xi}\left[\varepsilon(t,\xi)\right] \tag{7.30}$$

or, equally,

$$\xi = \arg\max_{\xi}\left[\sum_{k=1}^{n} g_k^2\right] \tag{7.31}$$

In order that Eq.(7.31) attains the maximum, we set the derivative of it with respect to ξ equal to zero, to obtain

$$\frac{d\left(\sum_{k=1}^{n} g_k^2\right)}{d\xi} = 2\sum_{k=1}^{n} g_k \frac{dg_k}{d\xi} = 0 \tag{7.32}$$

Based on the orthonormal property, it is easily to show that [15]

$$\frac{d\left(\sum_{k=1}^{n} g_k^2\right)}{d\xi} = \frac{n}{\xi} g_n g_{n+1} \tag{7.33}$$

The optimal Laguerre pole ξ that minimizes the squared error in Eq.(7.30) satisfies the following equation,

$$g_n g_{n+1} = 0 \tag{7.34}$$

which has the roots

$$g_n = 0 \quad \text{and/or} \quad g_{n+1} = 0 \tag{7.35}$$

Remarks:

- The problem of finding a minimum of $\varepsilon(t,\xi)$ with respect to ξ reduces to finding the zeros of either of the parameters g_n or g_{n+1} as a function of (real-valued) ξ, and then checking that the value of $g_n g_{n+1}$ changes sign from positive to negative or vice versa as ξ increases. If the sign of the product $g_n g_{n+1}$ changes at roots of Eq.(7.34), the stationary point is either a maximum or a minimum. Hence, the optimal value ξ can be obtained by determining $\sum_{k=1}^{n} g_k^2$ at all possible values of ξ and choosing the one that makes $\sum_{k=1}^{n} g_k^2$ largest.
- According to Eq.(7.35), it tells us that we are looking for the value of ξ, which corresponds to a zero of g_{n+1}, otherwise the model order can be reduced to n, i.e. $g_n=0$, without any change in model accuracy.

7.3.2 Optimality condition of discrete Kautz model

An approximation of the true unit sample response $h^0(t)$ by a truncated Kautz series expansion is given by

$$h(t) = \sum_{k=1}^{n/2} \{g_{2k-1}\psi_{2k-1}(t) + g_{2k}\psi_{2k}(t)\} \approx h^0(t) \tag{7.36}$$

where the Kautz sequences $\psi_{2k-1}(t)$ and $\psi_{2k}(t)$ are inverse z-transform of $\Psi_{2k-1}(z)$ and $\Psi_{2k-1}(z)$, given in Eq.(6.23) and Eq.(6.24) respectively. The squared error given in Eq.(7.28) can be simplified as

$$\varepsilon(t,\xi) = \sum_{t=1}^{\infty}\left[h^0(t)\right]^2 - \sum_{k=1}^{n/2}\left\{g_{2k-1}^2 + g_{2k}^2\right\} \tag{7.37}$$

The optimal Kautz pole is then defined as

$$\xi = \arg\min_{\xi}\left[\varepsilon(t,\xi)\right] \tag{7.38}$$

or, equivalently,

$$\xi = \arg\max_{\xi}\left[\sum_{k=1}^{n/2}\left\{g_{2k-1}^2 + g_{2k}^2\right\}\right] \tag{7.39}$$

where *arg min* and *arg max* are over the complex parameter $\xi = \alpha + j\beta$, where $|\xi| < 1$. In order that Eq.(7.39) attains the maximum, we set the derivative of it with respect to the real part, α, and to the imaginary part, β, of ξ equal to zero by using complex gradient operator, such that

$$\frac{\partial\varepsilon(t,\xi)}{\partial\alpha} = 2\Re\left[\frac{\partial\varepsilon(t,\xi)}{\partial\xi}\right] = 0 \tag{7.40}$$

$$\frac{\partial\varepsilon(t,\xi)}{\partial\beta} = -2\Im\left[\frac{\partial\varepsilon(t,\xi)}{\partial\xi}\right] = 0 \tag{7.41}$$

where $\Re$ and $\Im$ denote real and imaginary part, respectively. Eventually, the optimal solution satisfying the Eq.(7.40) - (7.41) is given by [1]

$$g_{n-1} = g_n = 0 \tag{7.42}$$

and/or

$$g_{n+1} = g_{n+2} = 0 \tag{7.43}$$

Similarly, given the n^{th} order Kautz model, the problem reduces to finding the zeros of the product of consecutive Kautz parameters either ($g_{n-1}g_n$) or ($g_{n+1}g_{n+2}$). Having found a set of all possible Kautz poles ξ, the optimal Kautz pole can then be obtained by determining $\sum_{k=1}^{n/2}\{g_{2k-1}^2 + g_{2k}^2\}$ at all possible values of ξ, and choosing the value that makes $\sum_{k=1}^{n/2}\{g_{2k-1}^2 + g_{2k}^2\}$ maximum.

7.4 Suggestion on Model Reduction

So far, we have dealt with the problem of finding an optimal dominating pole for each model order which leads to the minimum fit of error, regardless of how large the model order is. Herein, we shall "just" give a remarkable suggestion on how to further reduce the order of the proposed models given in section 6, but still retain an acceptable fit of error and hold sufficient energy of the system.

Most applications are required to have the estimated model with the lowest possible order according to some error criterion. This can be considered in terms of the energy of the system in such a way that the model containing the maximum energy with less number of estimated parameters will be mostly chosen, irrespective of how large the fit of error is. Clearly, this is a compromise between the model order and the fit of error that one should take into consideration for each specific purpose. The energy is usually contained in the impulse response of the system. Let *h(t)* be the system impulse response. The energy can then be calculated in the time-domain, in the frequency domain using Parseval theorem, or from the expansion coefficients in orthonormal function given in Eq.(7.26), i.e.

$$E = \sum_{t=1}^{\infty} h(t)h^*(t) = \sum_{k=1}^{n} |g_k|^2 \tag{7.44}$$

where $(\bullet)^*$ denotes the conjugate operation. Two schemes for model reduction, namely the threshold technique and the ordering of the poles, are presented here and their applications will be demonstrated in section 8.

7.4.1 Threshold technique

This scheme was proposed by P. Bodin [19]. The concept can be summarized as follows. The system which is represented by the sum of orthonormal bases will have nice interpretation properties. For example, the energy of the system is always preserved in the coefficient magnitudes of the basis functions, and the coefficient estimates remain the same although some basis functions are discarded. Since the signal is usually contaminated with unknown disturbances or noise of some kind, the basis function should be chosen as "far away" as possible from the disturbances to be able to recover the signal correctly. Given the suitable order of an orthonormal model which yields a minimum fit of error, one can take a

close look at each coefficient magnitude. The coefficient magnitudes that are small compared to the noise in each basis function can be set to zero or, in other words, *thresholded*. For instance, if the noise is assumed to be white in the sense that it is equally distributed over all possible basis functions, small coefficients can be considered to mostly consist of noise. Therefore, it is a good idea to choose a basis that collects the information in a few large coefficients and many small ones rather than the other way around. For a deep detail, one can see in [19].

By applying the threshold technique over an estimated model, one obtains the suitable model that has less number of estimated parameters but still holds adequate energy.

7.4.2 The ordering of the poles

Since the systems in reality normally have many poles and zeros, it is then a good idea to incorporate a variety of poles in orthonormal model. Albertus [2] had proposed the solution of model reduction by considering *the ordering of the poles* according to a criterion of the system energy in such a way that the last sections contribute least or the first sections contribute most to the overall energy of the system impulse response in quadratic sense. However, he required a given set of possible poles which was estimated from the Prony's method [17] in order not to optimize over the poles within unit circle. See in [2] for more details. To simplify our discussion here, we shall not require such a given set of possible poles. Instead, each dominating pole will be chosen within unit circle based on the concept of the ordering of the poles. This is of course a drawback of our modified method when searching the dominating pole at each model order, especially at very high model order.

The steps of choosing each dominating pole are as follows:

1. Choose the first order section with a real-valued pole if
   a. The absolute value of the weight associated with this pole is largest.
   b. The squared absolute value of this weight is larger than the sum of squared absolute values of the weights of two additional first order sections for complex-conjugated pole pair.
2. Select two additional first order sections either with a complex-conjugated pole pair or with two real-valued poles if there is no real pole satisfying the previous condition. This selection is as before based on a maximum additional energy.
3. Go to the step 1 again to find the next dominating pole until the increment of an additional energy is less than an acceptable level; for instance, an additional energy is lower than 0.1 percent of the preceding energy.

Note that, in this thesis project, we shall not concern on how to employ these two schemes in the real acoustic echo data to reduce the number of estimated parameters, because we are focusing on the model leading to the minimum fit of error, based on the cross validation approach.

8. Simulated Examples

In this section, three examples are given to demonstrate the performance of using the Laguerre and Kautz models, compared to FIR model. They are all implemented with the FIR structure. We have also introduced and examined the performance of using the two-stage echo canceller in the third example. Furthermore, we have illustrated the use of the concept of model reduction as given in section 7.4, where the threshold technique is applied in the first two examples and the ordering of the poles is described only in the third example.

Example 1: Given a continuous time system being sampled with a rapid time sampling, we shall investigate the performance of using Laguerre model, compared to FIR model.

Let the continuous time transfer function in *s*-domain be given by

$$G(s) = \frac{3}{s+1} \tag{8.1}$$

and be sampled with T = 0.1. The corresponding discrete time transfer function in *z*-domain then has a pole located at 0.9048 $\{e^{(-1)(0.1)}\}$, a statistic gain of 3 and the delay of 1. The system is excited by white sequences as the input.

Firstly, let us estimate the impulse response of this system using a FIR model of the order 100 as shown in Fig.8.1.

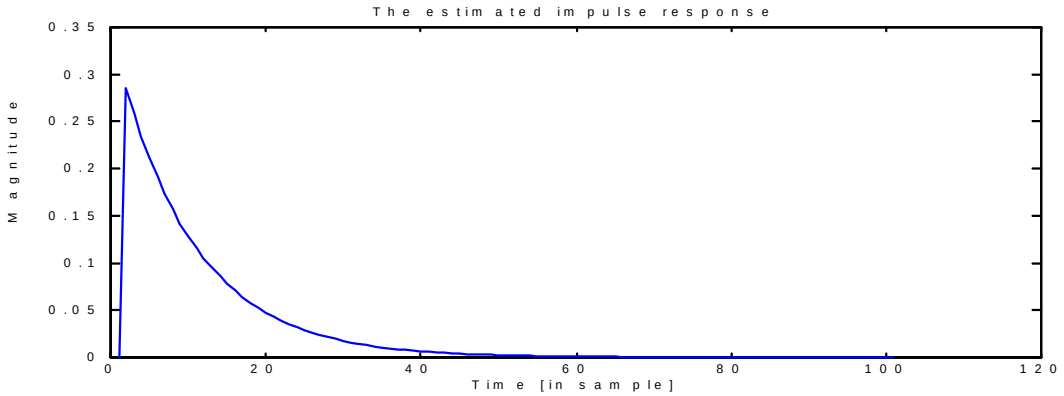

Figure 8.1: The estimated impulse response using a FIR model of the order 100.

By using the proposed method given in section 7.1, Eq.(7.11), to calculate the Laguerre pole ξ, one obtain it equal to 0.9048. Undoubtedly, the resulting Laguerre pole is exactly equal to the true discrete time pole because the system is the first order and has no disturbance.

Having found an optimal (Laguerre) pole, we shall compare the performance of using the Laguerre model with FIR model. The 1^{st} order Laguerre model with an optimal pole is then compared with the 6^{th} order FIR model. The bode plots of the true system, FIR model and Laguerre model are illustrated in Fig.8.2.

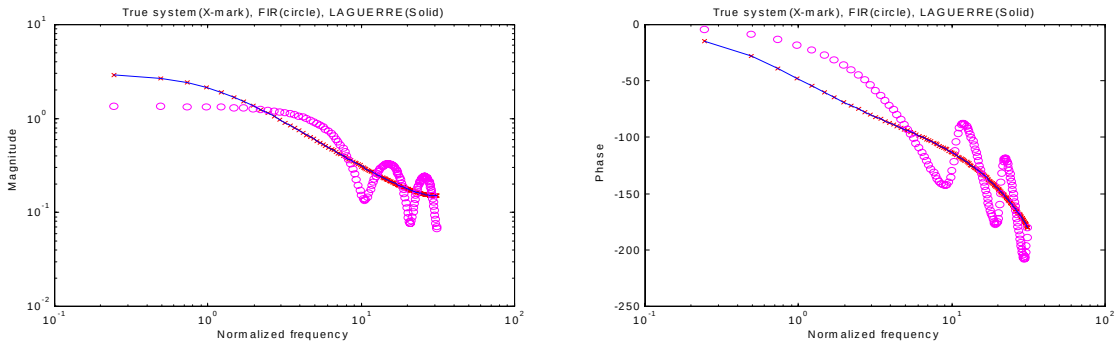

Figure 8.2: Bode plots (Magnitude and Phase) of transfer function approximations, where True system (X-mark), FIR (Circle) and Laguerre (Solid).

Obviously, only the 1^{st} order Laguerre model with an optimal pole is quite close to the true transfer function if compared to the 6^{th} order FIR model.

Next, to evaluate the fit of error, let us define **M12** model as the Laguerre model where the optimal Laguerre pole is calculated from Eq.(7.11) and is used to compute the fit of error at every model order. We shall compare the performance between M12 and FIR models by using the cross validation approach. This is done by splitting the data into 2 different sets which have 4000 samples each. The first one, ZE, is used to estimate the model parameters and the other one, ZV, is used to calculate the fit of error based on the resulting model parameters. The fit of error and the DB improvement of M12 model with a pole of 0.9048 are compared with those of FIR model as illustrated in Fig.8.3, when ZE≠ZV.

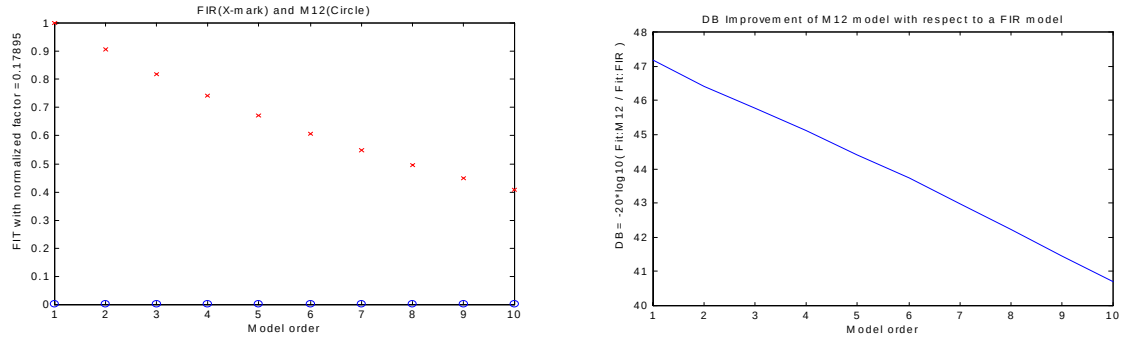

Figure 8.3: Left: Fits of error between FIR model (X-mark) and M12 model with a pole of 0.9048 (Circle). Right: The DB improvement of M12 model with respect to a FIR model at each model order.

Secondly, let us now demonstrate the other methods given in section 7.2 to find the optimal pole. The fits of error of FIR, M12 and Laguerre models are computed when ZE≠ZV and ZE=ZV, as given in appendices A1 and A2 respectively. The plots of DB improvement of the proposed models (M12 and Laguerre) with respect to a FIR model are depicted in Fig.8.4, where **DB** = -20*log10(Fit of error of the proposed model / Fit of error of a FIR model). Several conclusions can be drawn from these results. Firstly, the optimal pole obtained from the first method given in Eq.(7.11) is sufficient to use for modeling a stable LTI system, particularly when it is the first order system. Secondly, the second proposed method given in section 7.2 in which the optimal pole is computed from Eq.(7.23) leads to a better performance. This can be shown when the set of data ZV=ZE is used to calculate the fit of error as depicted in the Fig.8.4 (Right) because it gives an optimal pole at each model order. Accordingly, the second method given in section 7.2 is much more efficient to find the

optimal pole than the first method given in section 7.1 as a consequence of the fact that the optimal pole is dependent of the model order. Finally, as illustrated in the Fig.8.4 (Left), one gains less DB improvement when the model order gets higher. This is because the correct choice of an optimal pole is not crucial since the transfer function can be arbitrary well approximated by an orthonormal model for any choice of pole ξ ($|\xi| < 1$) by just increasing the model order substantially large enough [6]. However, since the variance of an estimated model is proportional to the number of estimated parameters, we shall therefore interest of using as few parameters as possible but still give the reasonable approximation.

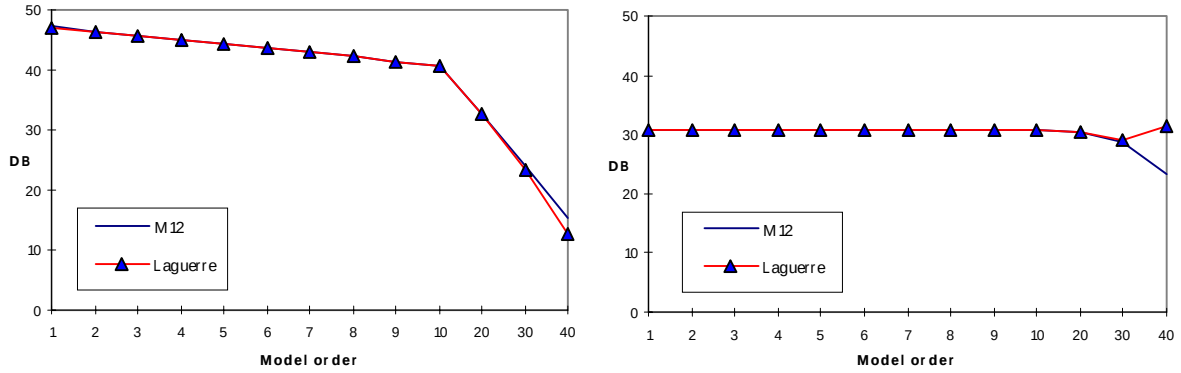

Figure 8.4: The DB improvement of the proposed models with respect to a FIR model at each model order, where the Left figure is for ZE≠ZV and the Right figure is for ZE=ZV.

Thirdly, we shall investigate the concept of model reduction based on the threshold technique as given in section 7.4.1. Obviously, it is sometimes "preferable" to have the model with as few estimated parameters as possible but still preserve the energy of the system sufficiently, regardless of how large the fit of error is. Since the energy can be considered as the magnitude of the estimated coefficients, let us plot the coefficient magnitudes of the 20^{th} order Laguerre model with a pole of 0.9763 (a Laguerre pole at the 20^{th} order given in an appendix A1) and with a pole of 0.9048 (a true pole), as shown in Fig.8.5.

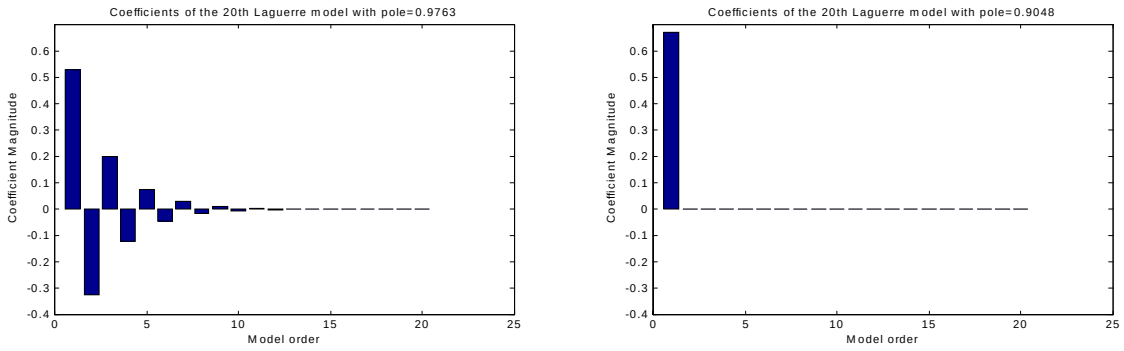

Figure 8.5: The coefficient magnitudes of the Laguerre model with pole 0.9763(Left) and 0.9048 (Right).

Based on the threshold technique, it is apparent that one may need about the 12^{th} order Laguerre model with a pole of 0.9763, whereas it is quite enough to use only the 1^{st} order Laguerre model with a pole of 0.9048. This is because the coefficient magnitudes that are very small can be negligible according to the concept of model reduction.

Eventually, let us now increase the number of estimation/validation data from 4000 samples to 8000 samples and then evaluate the fits of error of FIR, M12 and Laguerre models,

when ZE≠ZV. The fits of error are given in an appendix A3 and the plot of the corresponding DB improvement up to the order of 40 for the data length of 8000 and 4000 samples is depicted in Fig.8.6. Clearly, one obtains a better DB improvement when the data length increases. This is a basic property of using the least squares method to estimate the model parameters, i.e. the larger the data length used, the better the performance that we shall get from an approximation. Furthermore, the optimal pole at each model order given in an appendix A3 is slightly different from that given in an appendix A1 because the optimal dominating pole calculating from Eq.(7.23) is also dependent of the data length.

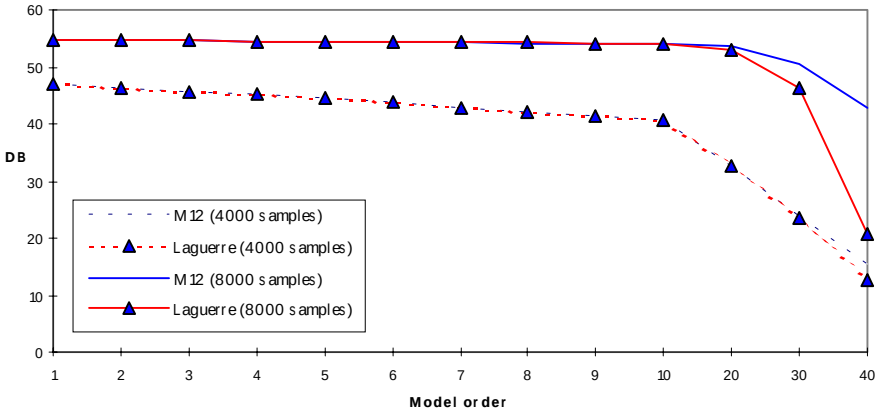

Figure 8.6: The DB improvement of the proposed models with respect to a FIR model at each model order, when ZE≠ZV with the data length of 8000 (Solid) and 4000 (Dashdot).

Example 2: Given the second order resonant system, we shall investigate the performance of using Kautz model, compared to FIR, M12 and Laguerre models.

Consider the discrete time transfer function in *z*-domain

$$G(z) = \frac{z^{-1}}{1 - 1.6z^{-1} + 0.89z^{-2}} \tag{8.2}$$

which has the system poles located at 0.8±0.5i within the unit circle. The system is excited by white sequences as the input.

The estimated impulse response of this system using a FIR model of the order 200 is given in Fig.8.7. By using the Eq.(7.11), one obtains the optimal Laguerre pole equal to 0.5562.

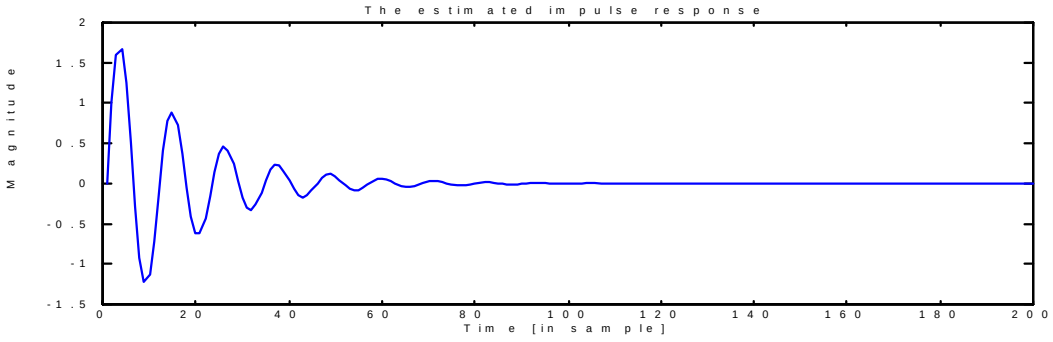

Figure 8.7: The estimated impulse response using a FIR model of the order 200.

The bode plots of the true system, the *10th* order FIR model, and the *2nd* order Laguerre model with a pole of 0.5562 are shown in Fig.8.8. Similarly, by using the cross validation approach (ZE≠ZV), the fit of error and the DB improvement of M12 model with a pole of 0.5562 is compared to those of a FIR model as depicted in Fig.8.9.

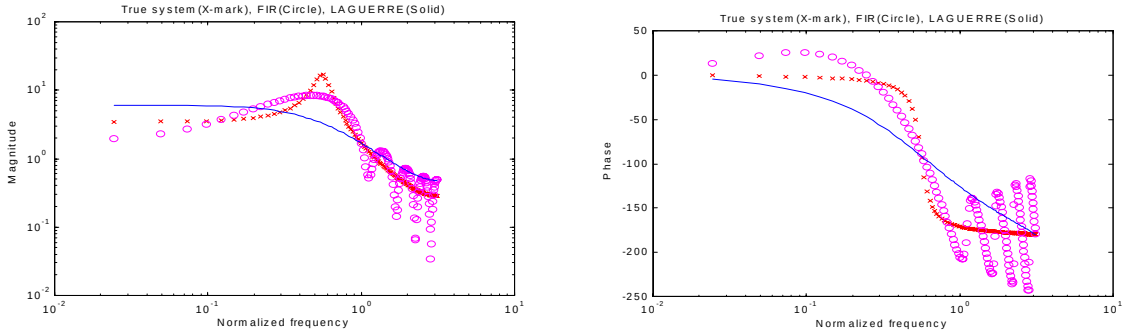

Figure 8.8: Bode plots (Magnitude and Phase) of transfer function approximations, where True system (X-mark), FIR (Circle) and Laguerre (Solid).

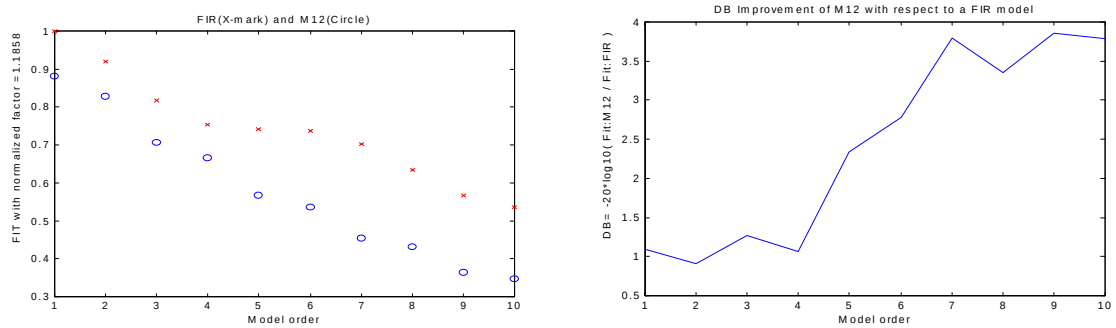

Figure 8.9: Left: Fits of error between FIR model (X-mark) and M12 model with a pole of 0.5562 (Circle). Right: The DB improvement of M12 model with respect to a FIR model at each model order.

Even though the M12 model gives a better performance than FIR model, one can still obtain higher improvement if the optimal pole is chosen as a complex value according to the true dynamics. In this case, the *2nd* order Kautz model with a pole of 0.8+0.5i is instead used. The bode plots of the true system, the *10th* order FIR model and the *2nd* order Kautz model are shown in Fig.8.10. It is evident that only the *2nd* order Kautz model with a pole of 0.8+0.5i is quite close to the true system. Fig.8.11 demonstrates the plots of the fit of error and the DB improvement between a Kautz model with a fixed pole of 0.8+0.5i for every model order and a FIR model, when ZE≠ZV.

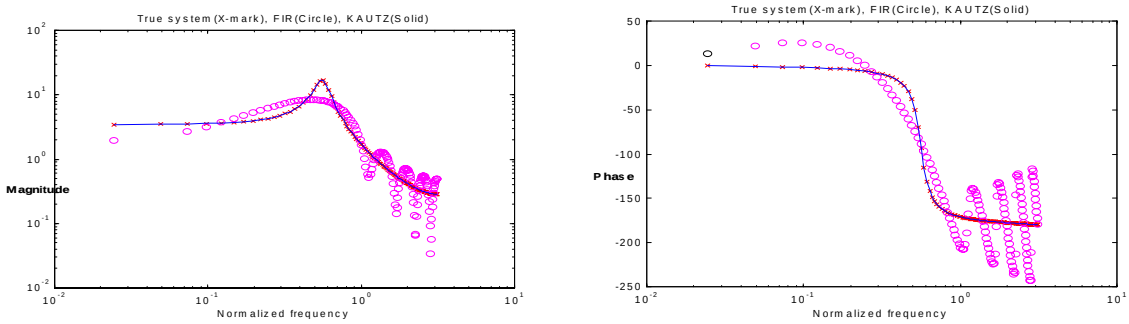

Figure 8.10 : Bode plots (Magnitude and Phase) of transfer function approximations, where True system (X-mark), FIR (Circle) and Kautz (Solid).

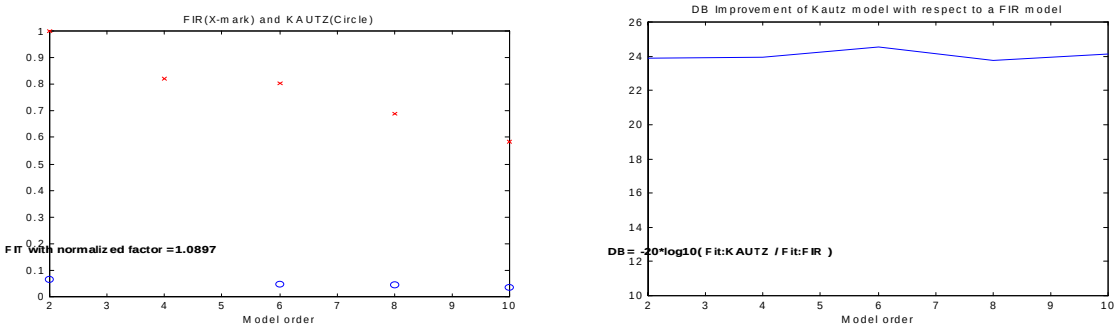

Figure 8.11: Left: Fits of error between FIR model (X-mark) and Kautz model (Circle).
Right: The DB improvement of Kautz model with respect to a FIR model at each model order.

Let us now evaluate the fits of error of FIR, M12, Laguerre and Kautz models as given in an appendix A4. Hence, the plot of the corresponding DB improvement of the proposed models with respect to a FIR model up to the order of 40 is plotted in Fig.8.12, when ZE≠ZV.

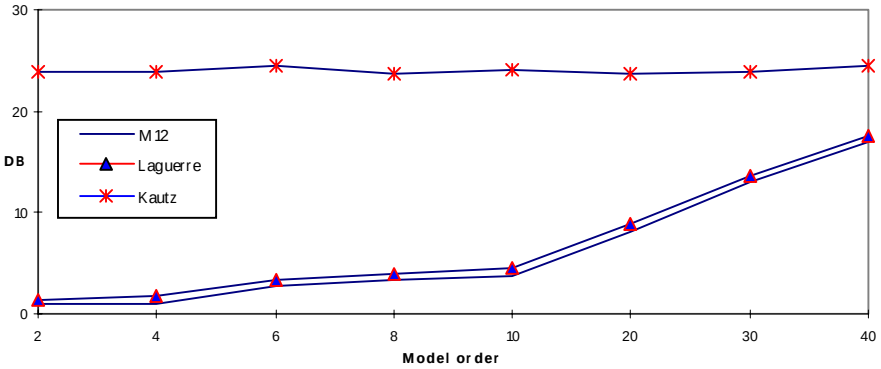

Figure 8.12: The DB improvement of the proposed models with respect to a FIR model at each model order.

Evidently, Kautz model give a better performance than M12 and Laguerre models, which are all better than FIR model. In general, since any stable LTI system in reality is likely to be a resonant system, Kautz model is preferable to model such a system because it requires less number of estimated parameters.

Next, we shall demonstrate the effect of existing zero over the optimal dominating pole. Let the system in Eq.(8.2) have the zeros located at 0.7±0.4i. The zeros will basically cause the optimal pole location slightly changed to other positions. Therefore, the estimated impulse response using a FIR model of the order 200 is again calculated, and it is used to compute the optimal Laguerre pole from Eq.(7.11) which is equivalent to 0.4697. The fits of error for FIR, M12, Laguerre and Kautz models according to this system are given in an appendix A5. The plot of the corresponding DB improvement of the proposed models with respect to a FIR model is sketched in Fig.8.13, when ZE≠ZV. Clearly, the proposed model is still better than a FIR model. This is because the proposed methods to find the optimal dominating pole as given in section 7 still give satisfied results.

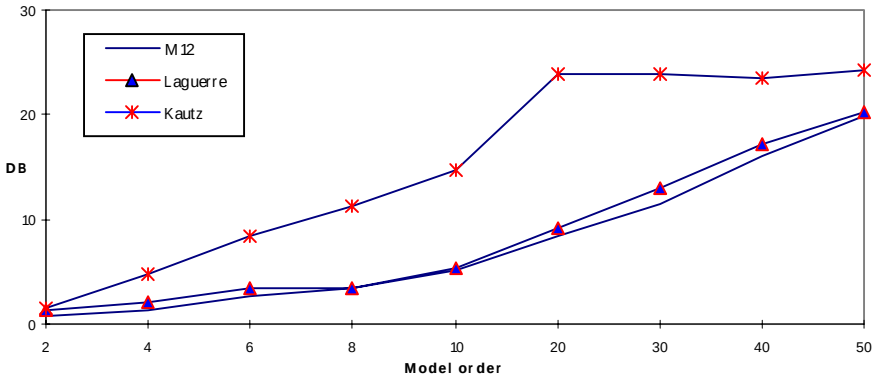

Figure 8.13: The DB improvement of the proposed models with respect to a FIR model at each model order.

Finally, we shall again examine the concept of model reduction based on the threshold technique as discussed in section 7.4.1. Let us consider the plot of coefficient magnitudes of the 40^{th} order Laguerre model with an optimal pole of 0.52 (given in an appendix A4), and the 40^{th} order Kautz model with an optimal pole of 0.67+0.42i (given in an appendix A4) and with a pole of 0.8+0.5i (a true pole), as depicted in Fig.8.14.

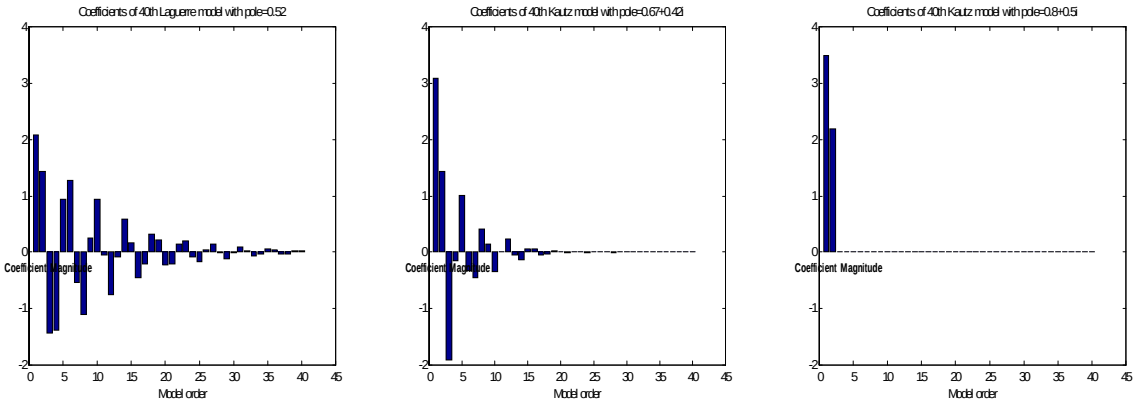

Figure 8.14: The coefficient magnitude of the Laguerre model with a pole of 0.52(Left), the Kautz model with a pole of 0.67+0.42i (Middle), and the Kautz model with a pole of 0.8+0.5i (Right).

Since the coefficient of Laguerre model at the order of 40 still contains some amount of the system energy, it would then be difficult to use this Laguerre model of the order less than 40. On the contrary, it seems that the whole system energy is kept in the coefficients up to the order of 20 and 2 for Kautz model with a pole of 0.67+0.42i and 0.8+0.5i, respectively. As a consequence, based on the threshold technique, only the 2^{nd} order Kautz model with a pole of 0.8+0.5i is quite enough to model this system because the coefficient magnitudes at higher orders are very small which can be neglected.

Example 3: We shall examine the performance of using the two-stage echo canceller, compared to FIR, Laguerre and Kautz models.

Consider the discrete time transfer function in *z*-domain given by

$$G(z) = \frac{z^{-1}(1+0.5z^{-1})(1-0.6z^{-1})^4(1-1.2z^{-1}+0.4z^{-2})^2}{(1-1.6z^{-1}+0.89z^{-2})^2(1+1.4z^{-1}+0.74z^{-2})(1+0.2z^{-1})(1-0.9z^{-1})^2} \qquad (8.3)$$

which has several poles and zeros within the unit circle. The system is excited by white sequences as the input. The estimated impulse response of this system using a FIR model of the order 200 is shown in Fig.8.15. By using the Eq.(7.11), one obtains the Laguerre pole equal to 0.0972.

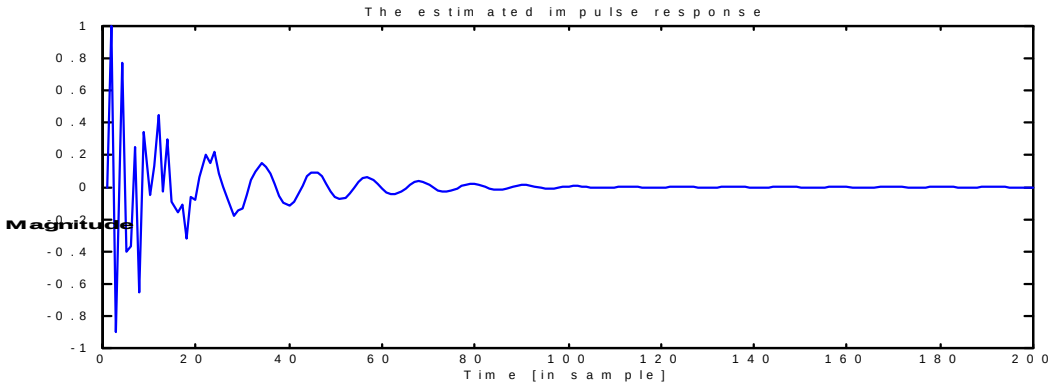

Figure 8.15: The estimated impulse response using a FIR model of the order 200.

As mentioned earlier, since the first part of this impulse response has a rapid time variation, while the remainder or the tail of this impulse response is slowly decaying towards zero, the model complexity can then be reduced if the tail can be modeled with less number of estimated parameters. The two-stage echo canceller is therefore introduced for the purpose of reducing the number of estimated parameters and the computational complexity. To evaluate the performance of using the two-stage echo canceller, the first stage, a FIR model with NB parameters, is then fixed at NB=30 and NB=50. The second stage is a Laguerre or Kautz filter with NA parameters. The fits of error of FIR, M12, Laguerre and Kautz models are given in an appendix A6, while the fit of error of a two-stage echo canceller is illustrated in an appendix A7. The plot of the corresponding DB improvement of the proposed models with respect to a FIR model is shown in Fig.8.16, when ZE≠ZV.

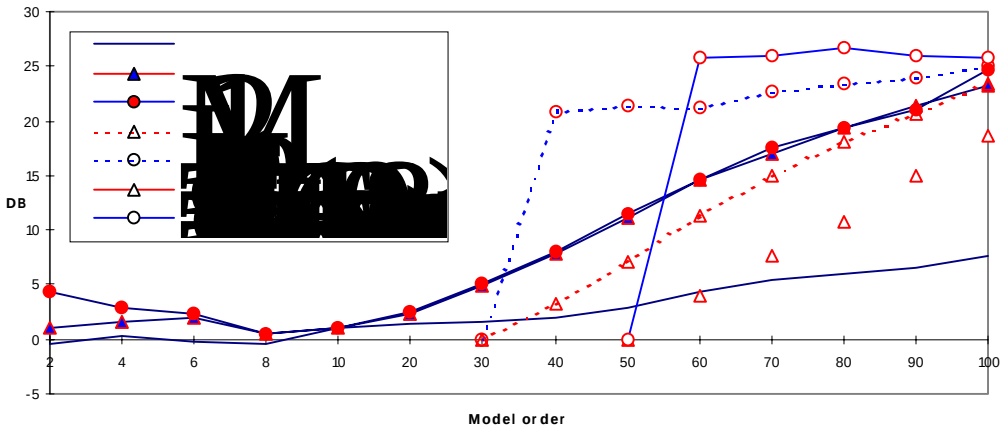

Figure 8.16: The DB improvement of the proposed models with respect to a FIR model at each model order

As shown in Fig.8.16, the performance of using the two-stage echo canceller has shown satisfied results and it has gained in terms of reducing the computational complexity, when compared with Laguerre and Kautz models. This is because the computational complexity can be viewed as the matrix size of orthonormal model used for calculating the optimal pole in Eq.(7.23). In addition, the two-stage echo canceller with Kautz filter as the second stage

yields a better performance than that with Laguerre filter. Consequently, the two-stage echo canceller is attractive method to model the impulse response of the system having a rapid time variation at the beginning part of the impulse response and a slow decay towards zero at the remainders.

Eventually, it is noteworthy to be pointed out here that, since the system given in Eq. (8.3) has several poles and zeros, it is then a good idea to incorporate a variety of poles in an orthonormal model. Let us now consider the concept of model reduction based on the ordering of the poles as given in section 7.4.2. By using this scheme, one obtains the useful orthonormal model with only the 11^{th} order with 3 complex-conjugated pole pair and 5 real-valued poles as given in table 8.1. Apparently, the fit of error at the 11^{th} order orthonormal model (*0.0135*) is close to that of the 50^{th} order Kautz model (*0.0140*), given in an appendix A6. When looking at the plot of coefficient magnitude of orthonormal model given in Fig.8.17 (Left), we shall clearly realize that only the 11^{th} order orthonormal model is possibly sufficient to use because the coefficient magnitudes at higher orders are very small which can be negligible.

In addition, it is also interesting to use the concept of model reduction on a two-stage echo canceller. Since the first part of the impulse response, say at the order of 4, in Fig.8.15 has rapid time variation, we shall use the FIR model of NB=4 to represent this part and the orthonormal model of the order that is resulted from the use of model reduction based on the ordering of the poles. Table 8.2 shows the fit of error up to the order of 12. Obviously, at the 11^{th} order, the fit of error in table 8.2 (*0.0134*) is close to that in table 8.1 (*0.0135*), but it gains in terms of less computational complexity. The plot of coefficient magnitude is also shown in Fig.8.17 (Right).

Table 8.1: Fits of error when ZE≠ZV with data length of 8000.

Model order	FIR	Model Reduction		
		Fit	Pole	**DB**
2	0.4431	0.2670	-0.70+0.50i	**4.40**
4	0.3664	0.1331	0.80+0.50i	**8.79**
6	0.3438	0.0355	0.80+0.50i	**19.72**
7	0.2856	0.0263	0.15	**20.72**
8	0.2696	0.0184	0.15	**23.32**
9	0.2692	0.0161	0.85	**24.46**
10	0.2663	0.0145	0.85	**25.28**
11	0.2311	***0.0135***	0.15	**24.67**

Remark:

- **FIR** column contains the fit of error of FIR model at each model order.
- **Model Reduction** column is the orthonormal model where a variety of poles are incorporated in the model. For example, at the 4^{th} order, it means that the ordering of poles is [-0.70+0.50i, 0.80+0.50i] and then use them to compute the fit of error.

Table 8.9: Fits of error when ZE≠ZV with data length of 8000 using

a two-stage echo canceller with NB=4.

Model order	FIR	Model Reduction		
		Fit	Pole	**DB**
1-4				
6	0.3438	0.2589	-0.70+0.50i	**2.46**
8	0.2696	0.1043	0.80+0.50i	**8.25**
10	0.2663	0.0177	0.80+0.50i	**23.55**
11	0.2311	***0.0134***	0.9	**24.73**
12	0.2311	0.0126	0.9	**25.27**

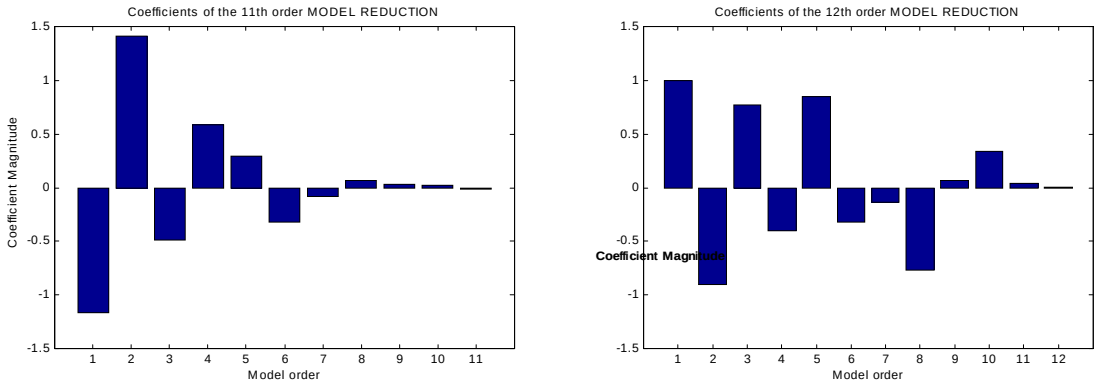

Figure 8.17: The coefficient magnitude of a orthonormal model (Left) and a two-stage echo canceller (Right)

Consequently, we can conclude the results from these three examples as follows. Firstly, Laguerre and Kautz models yield a better performance than FIR model for a given stable LTI system. Laguerre model is appropriate for the system with dominant first order dynamics, whereas Kautz model is suitable for the system with dominant second order resonant dynamics. Secondly, a proper choice of the optimal dominating pole will improve the performance of the proposed model. The methods to find the optimal pole given in section 7 have shown satisfied results. Thirdly, the performance of a two-stage echo canceller is comparable to that of Laguerre and Kautz models but it gains in terms of reducing the computational complexity. Finally, the concept of model reduction has illustrated that the model order of the proposed models can be further reduce but still hold sufficient energy of the system at the expense of a higher fit of error. Evidently, this is a compromise between the model order and the fit of error that one should take into account for each specific purpose.

9. Experimental Result on Real Acoustic Echo Data

In this section, the process of generating and collecting the real acoustic echo data is given. The performance of the proposed models given in section 6 is compared with that of a FIR model.

9.1 Process of generating and collecting the data

Since the real acoustic echo data employed in this thesis project has been collected from the system that is made as near stationary as possible in order to be able to use the off-line method to estimate the values of the model parameters, it is noteworthy to explain the process of generating and collecting the real acoustic echo data. Such a process can be represented in Fig.9.1. The microphone is located approximately 80 cm from the loudspeaker of high quality. The environment in which the recording is made is a hard acoustical room. The dimension of the room is approximately 3 m by 6 m and the height is 3 m.

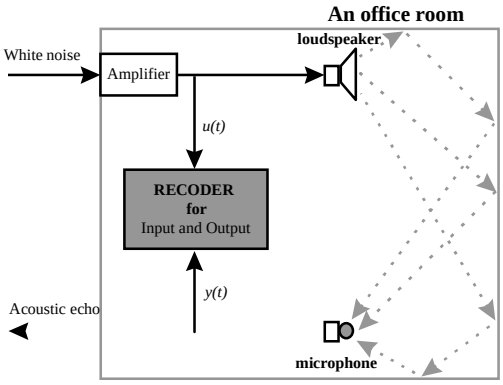

Figure 9.1: The process of generating an collecting the acoustic echo data

The signal (white noise) of 10 ms is fed to the amplifier before being collected at the recorder as the input signal, *u(t)*, and transmitted to the loudspeaker. The signal is propagated in the room and then reflected back to the microphone as acoustic echo which is collected by the same recorder as the output signal, *y(t)*. All signals collected at the recorder are sampled at 12 kHz. Nothing is moved in this room.

9.2 Assumption

To model the room impulse response based on a given set of observed input signal, *u (t)* and output signal *y(t)*, we shall firstly assume that the system (acoustic echo path) is linear and stationary for a short time interval. Although there might be some non-linearity characteristics presented in the system, the assumption of linearity still gives satisfied results depending on an amount of non-linearity of the system. It is difficult to gain a deep insight about the echo generating system for each specific purpose. As a result, it is suitable to model

the system by using a linear black box model structure which is primarily based only on a given set of experimental data. The task is then to estimate the impulse response of the system (acoustic echo path), since if the system is linear, its impulse response will completely represent it.

9.3 Experimental result

Let us estimate the impulse response of the acoustic echo path from the input-output data as explained in section 9.1. The estimated room impulse response using a FIR model of the order 3500 is shown in Fig.9.2. Clearly, the estimated impulse response is quite time variation. Therefore, one might expect the difficulty to model this impulse response using a FIR model because its model order is probably in the order of a few thousands.

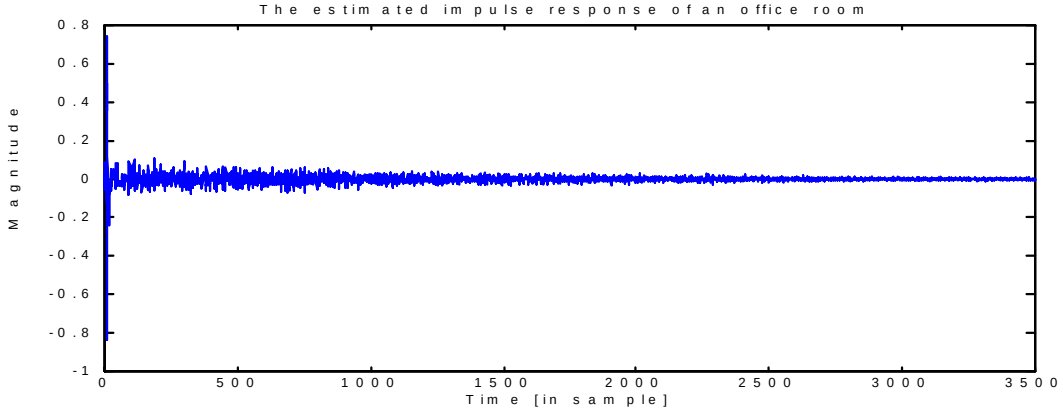

Figure 9.2: The estimated room impulse response using a FIR model of the order 3500.

Now we shall compare the performance of the proposed models as given in section 6 with that of a FIR model. The criterion used to evaluate the performance of each model is the cross validation approach where ZE≠ZV. Each set of estimation/validation data of 8000 samples normalized by 15991 are employed in this thesis project. Several aspects are worth to address here, such as how the segmented data used for estimation and validation affect the resulting optimal dominating pole, and how large the data size used in an approximation influences the fit of error and the optimal dominating pole.

Firstly we shall investigate how the segmented data used for estimation and validation affect the resulting optimal dominating pole. Three different segments of data are randomly chosen. We shall use the segment of estimation data at ZE=10000:17999 and validation data at ZV=20000:27999 as the reference set. The optimal dominating pole and the corresponding fit of error from this data set are given in an appendix B1. The plot of the DB improvement of the proposed models with respect to a FIR model is depicted in Fig.9.3. In addition, the resulting optimal dominating pole given in an appendix B1 is used to calculate the fits of error for the other two sets of segmented data which are given in an appendix B2 and the plots of their DB improvement are sketched in Fig.9.4. As shown in Figs.9.3-9.4, the performance of Laguerre and Kautz models is generally better than that of a FIR model. Furthermore, one can adequately use the optimal dominating pole calculated from one segmented data to evaluate the fit of error for the other segmented data.

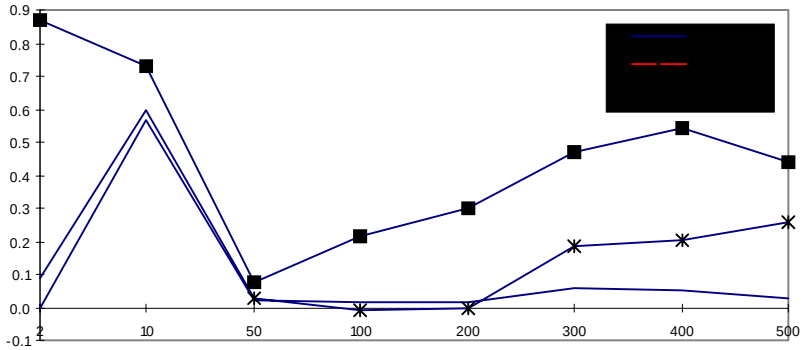

Figure 9.3: The DB improvement of the proposed models with respect to a FIR model at each model order as **the reference set** when ZE≠ZV.

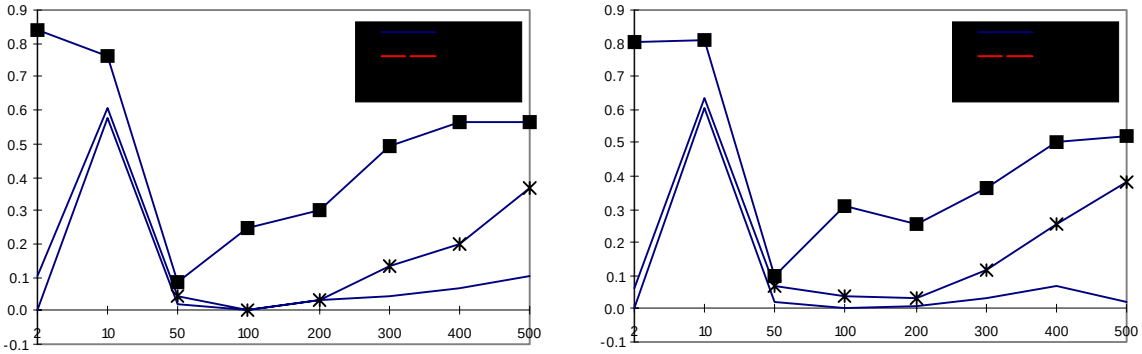

Figure 9.4: The DB improvement for the other **two sets of segmented data** using **the same dominating pole** as given in an appendix B1 to evaluate the DB improvement when ZE≠ZV.

Now let us recalculate the optimal dominating pole for each segmented data which is the same as given in an appendix B2. The resulting dominating pole and the corresponding fit of error are therefore illustrated in an appendix B3 and the plots of their DB improvement are shown in Fig.9.5.

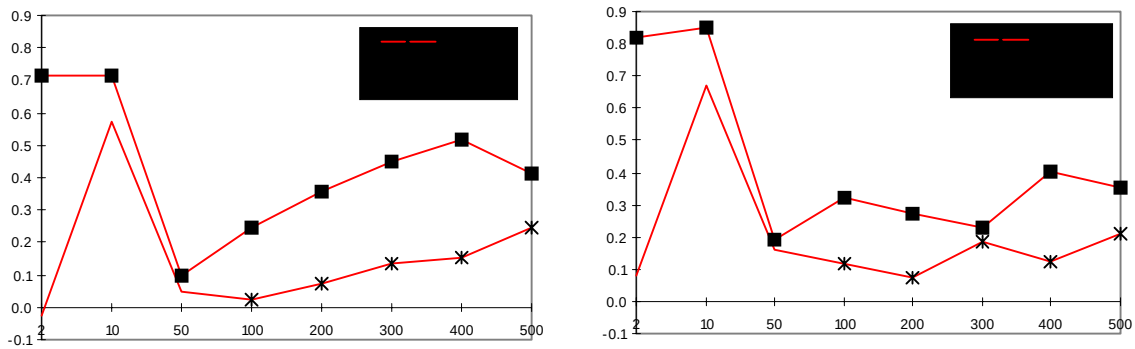

Figure 9.5: The DB improvement for **the two sets of segmented data** as given in an appendix B2 using **the new dominating pole** recalculated from Eq.(7.23) to evaluate the DB improvement when ZE≠ZV.

Evidently, the resulting dominating poles given in an appendix B3 are slightly different from those given in an appendix B1. This is because the optimal dominating pole calculated from Eq.(7.23) depends on the set of segmented data used in estimating the model parameters.

Next we shall examine how large the data size used in an approximation influences the fit of error and the optimal dominating pole. The fits of error of two sets of data size, 4000 and 12000 samples, are shown in an appendix B4 by using the same dominating pole given in an appendix B1 which was calculated from the data size of 8000 samples. The plots of DB improvement for these two sets of data size are depicted in Fig.9.6, when ZE≠ZV. Noticeably, the dominating poles computed from the data size of 8000 samples at each model order can still be used for the data sizes of 4000 and 12000 samples. Using the data size of 12000 samples in an approximation yields a better performance than using the data size of 4000 samples. As a result, the larger the data size, the better the fit of error we shall get.

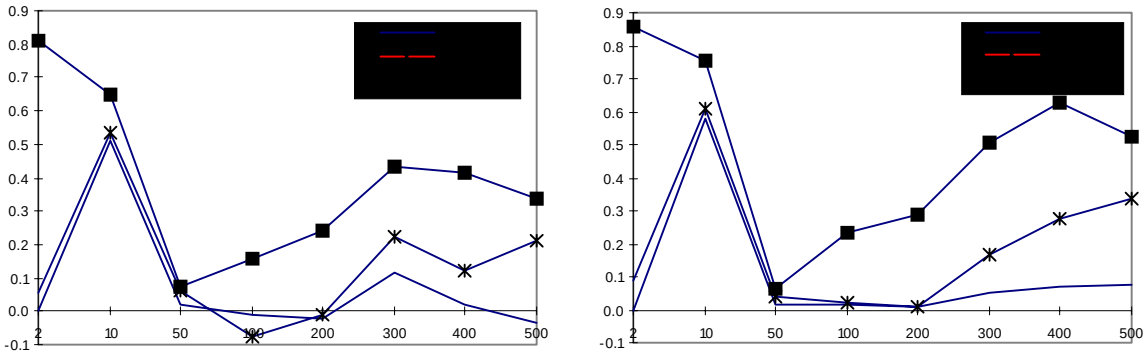

Figure 9.6: The DB improvement for **two data sizes**, 4000 samples (Left) and 12000 samples (Right), using **the same dominating pole** given in an appendix B1 to evaluate the DB improvement when ZE≠ZV.

Now let us recalculate the dominating pole and the corresponding fit of error for each sets of data size as given in an appendix B4. The results are shown in an appendix B5 and the plots of DB improvement are illustrated in Fig.9.7.

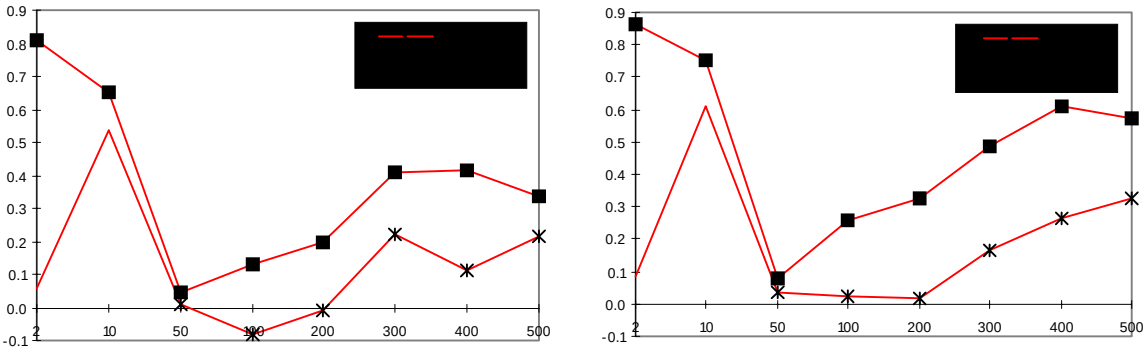

Figure 9.7: The DB improvement for **two data sizes**, 4000 samples (Left) and 12000 samples (Right), using **the new dominating pole** recalculated from Eq.(7.23) to evaluate the DB improvement when ZE≠ZV.

Similarly, the recalculated dominating poles are slightly different from those which are derived from the data size of 8000 samples as given in an appendix B1. This is because the optimal dominating pole calculated from Eq.(7.23) is also dependent of the data size used in estimating the model parameters.

Eventually, the performances of using Laguerre and Kautz models as well as the two-stage echo canceller up to order 2000 are given in an appendix B6. The plots of their corresponding DB improvement are depicted in Figs.9.8-9.9, when ZE≠ZV.

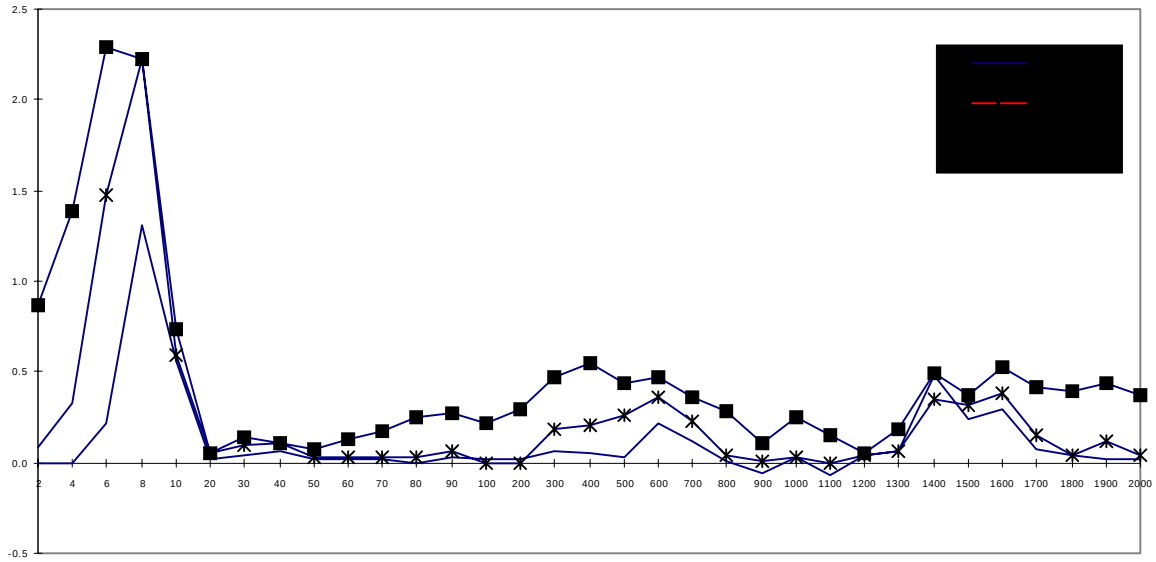

Figure 9.8: The DB improvement of M12, Laguerre and Kautz models with respect to a FIR model at each model order when ZE≠ZV and the data length of 8000 samples.

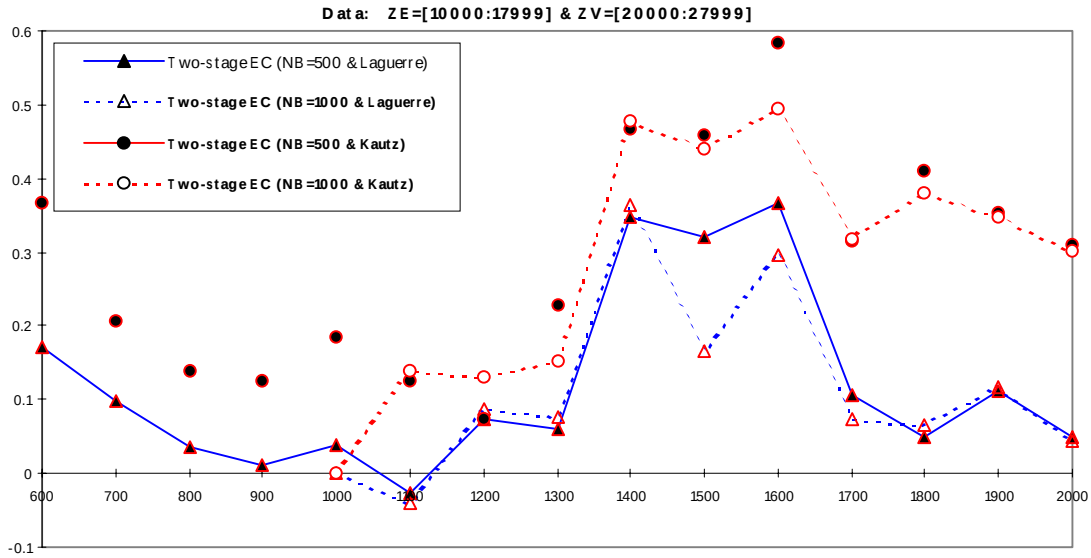

Figure 9.9: The DB improvement of a two-stage echo canceller with respect to a FIR model at each model order when ZE≠ZV and the data length of 8000 samples.

From Figs.9.8-9.9, we can roughly conclude the results as follows:

- The proposed models given in section 6 generally yield better performance than a FIR model when the dominating pole is chosen properly at each model order. The methods to find the optimal dominating pole as given in section 7 have shown satisfied results.
- The performance of a two-stage echo canceller is comparable to that of Laguerre and Kautz models but it has gained in terms of reducing the computational complexity.
- We obtain small DB improvement on the real acoustic echo data by using the proposed models given in section 6 if compared with a FIR model. This might be because the system (acoustic echo path) may have some non-linearity and time variation.

10. Conclusions

As known in the literature, the effect of acoustic echo is quite complicated as discussed in section 1. FIR model is widely used to model the room impulse response due to simplicity and stability. However, it leads to the approximation of very high order, probably in the order of 4000. By exploiting a priori information about the dominating pole of the system, an approximation of the room impulse response by means of the Laguerre and Kautz functions is proposed. In general, such a priori information about the dominating pole of the system may sometimes not be available. To deal with this problem, we have presented three methods to find such an optimal dominating pole in section 7.

The first method given in section 7.1 is restricted only in the case of a real-valued dominating pole, and it requires a given set of the system impulse response to calculate an optimal Laguerre pole by minimizing the performance index J as given in Eq.(7.11). The advantage of using such a performance index is that it forces rapid convergence of the Laguerre spectrum and gives an analytical solution. The resulting optimal dominating pole will primarily depend on the characteristics of the system impulse response, such as its rate of decay, its smoothness and the time delay.

The second method is much more efficient than the first one with an expense of the higher computational complexity. This is because, since this method is based on minimizing the loss function which is a function of the model order and its dominating pole, there exists an optimal dominating time at each model order. The optimal dominating pole is needed to calculate at each model order by Eq.(7.23) and its value can be either real-valued or complex-valued depending on our intended use. The third method relies on the fact that the squared error or loss function is a function of a dominating pole. The optimal dominating pole can then be derived mathematically by setting the derivative of the squared error with respect to a dominating pole equal to zero. Given the n^{th} model order, the optimal condition of Laguerre model is given in Eq.(7.34), while that of Kautz model is given in Eq.(7.42)-Eq.(7.43). Obviously, the major shortcoming of these two methods is the necessity of subdividing the whole interval of a dominating pole into smaller subintervals of more manageable size, in order to avoid the computational complexity of approximation at very high order.

According to the results in simulated examples given in section 8, when modeling a stable LTI system, all proposed models discussed in section 6 have performed quite better than a FIR model, especially for a Kautz model. Therefore, Kautz model seems to have large potential in real applications such as in signal processing and in control system, because the system in reality is likely to be resonant system. In addition, the performance of two-stage echo canceller is comparable to that of Laguerre and Kautz models but it has gained in terms of reducing the computational complexity. The concept of model reduction given in section 7.4 has also shown that the model order of the proposed models can be further reduce but still preserve enough energy of the system at the expense of a higher fit of error. Evidently, this is a compromise between the model order and the fit of error that one should take into consideration for each specific purpose.

Finally, we obtain small DB improvement on the real acoustic echo data by using the proposed models if compared with a FIR model This might be because the system (acoustic echo path) may have some non-linearity and time variation.

11. Suggestions for Future Works

All proposed models given in section 6 and the methods used to find the dominating pole given in section 7 have illustrated satisfied results. Their parameters is estimated and solved by means of an off-line method in the least squares sense. Since the true echo path in reality has some non-linearity and time variation, it might be difficult to obtain a large improvement from modeling the true echo path by the proposed models, if compared with a FIR model.

In order to enhance and improve the performance of the proposed models, it is then essential to deal with the problems of non-linearity and time variation. Hence, several future works of interest should be suggested here as follows:

- Implement the proposed models in the neural network to deal with the non-linearity of the true echo path.
- Apply the on-line method to estimate the model parameters used in the proposed models in order to cope with a constraint of time variation of the true echo path.
- Investigate and expand the concept of model reduction, as described in section 7.4, in real applications.

12. References

[1] Albertus C. Den Brinker and Frank P.A. Benders and Tomas A.M. Oliveira e Silva, *Optimality Conditions for Truncated Kautz Series*, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol.43, No.2, February 1996, pp.117-121.

[2] Albertus C. Den Brinker and Harm J.W. Belt, *Model Reduction by Orthogonalized Exponential Sequences*, IEEE Workshop on Circuits, Systems and Signal Processing, Mierlo (NL), November 27-28, 1996, pp.77-82.

[3] Bo Wahlberg, *On the Use of Orthogonalized Exponentials in System Identification*, Technical report LiTH-ISY-I-1099, Link?pings university, Sweden, 1990-06-15.

[4] Bo Wahlberg, *System Identification Using Laguerre Models*, IEEE Transactions on Automatic Control, Vol. 36, No.5, May 1991.

[5] Bo Wahlberg, *System Identification Using Kautz Models*, IEEE Transactions on Automatic Control, Vol. 39, No.6, June 1994.

[6] Bo Wahlberg and E.J. Hannan, *Parametric Signal Modelling Using Laguerre Filters*, Technical report LiTH-ISY-I-1086, Link?pings university, Sweden, 1990-05-02.

[7] Bo Wahlberg and P.M. M?kil?, *On Approximation of Stable Linear Dynamical Systems using Laguerre and Kautz Functions*, Automatica, Vol. 32, No.5, 1994, pp.693-708.

[8] Brett Ninness and Fredrik Gustafsson, *A Unifying Construction of Orthonormal Bases for System Identification*, IEEE Transactions on Automatic Control, Vol. 42, No.3, March 1997.

[9] Gordon W. Davidson and David D. Falconer, *Reduced Complexity Echo Cancellation Using Orthonormal Functions*, IEEE Transactions on Circuits and Systems, Vol. 38, No.1, January 1991, pp.20-28.

[10] J. Schoukens and R. Pintelon, *Identification of Linear Systems: A Practical Guideline to Accurate Modeling*, Pergamon Press, Great Britain, 1991.

[11] Lennart Ljung, *System Identification*, Technical report, Department of Electrical Engineering, Link?ping University, Sweden, May 29, 1995, pp. 62.

[12] Lennart Ljung, *System Identification Theory for the user*, Prentice-Hall, Inc., New Jersey, 1987.

[13] Lennart Ljung, *System Identification Toolbox: For Use with MATLAB*, The MathWorks, Inc., 1995.

[14] Lester S.H. Ngia, *An overview of the Fundamental Problems and Solutions to Echo Cancellation*, Technical report CTH-TE-62, Dept. of Applied Electronics, Chalmers University of Technology, G?teborg, Sweden, September 18, 1997.

[15] L. Wang and W. R. Cluett, *Optimal Choice of Time-scaling Factors for Linear System Approximations Using Laguerre Models*, IEEE Transactions on Automatic Control, Vol. 39, No.7, July 1994, pp.1463-1467.

[16] Mohammad A. Masnadi-Shirazi and N. Ahmed, *Optimum Laguerre Networks for a Class of Discrete-Time Systems*, IEEE Transactions on Signal Processing, Vol. 39, No.9, September 1991, pp.2104-2108.

[17] Monson H. Hayes, *Statistical Digital Signal Processing and Modeling*, Jonh Wiley & Sons, Inc. USA., 1996.

[18] N. Tanguy, P. Vilbe, amd L. C. Calvez, *Optimum Choice of Free Parameter in Orthonormal Approximations*, IEEE Transactions on Automatic Control, Vol. 40, No.10, October 1995, pp.1811-1813.

[19] Per Bodin, *On the selection of Best Orthonormal Basis in System Identification and Signal Processing*, Phd. Thesis, Automatic Control Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, Sweden, 1997.

[20] S?derstr?m, T. And Stoica, P. G., *System Identification*, Prentice-Hall, Hemel Hempstead, U.K. 1989.

[21] Tomas Oliveira e Silva, *On the determination of the Optimal Pole Position of Laguerre Filters*, IEEE Transactions on Signal Processing, Vol. 43, No.9, September 1995, pp.2079-2087.

[22] Tomas Oliveira e Silva, *Optimality Conditions for Truncated Laguerre Networks*, IEEE Transactions on Signal Processing, Vol. 42, No.9, September 1994, pp.2528-2530.

[23] Y.Fu and G.A.Dumont, *An optimum Time Scale for Discrete Laguerre Network*, IEEE Transactions on Automatic Control, Vol. 38, No.6, June 1993, pp.934-938.

Appendix A (Used for section 8)

A1: Fits of error of the *1st* order system when **ZE≠ZV** and data length of **4000** samples. (Fig.8.4(Left))

Model order	FIR	M12 (0.9048)		Laguerre		
		Fit	DB	Fit	Pole	DB
	1.0e-003 *	1.0e-003 *	(M12/FIR)	1.0e-003 *		(Laguerre/FIR)
1	178.9000	0.7822	**47.19**	0.7856	0.9049	**47.15**
2	161.9000	0.7737	**46.41**	0.7737	0.9050	**46.41**
3	146.5000	0.7541	**45.77**	0.7541	0.9059	**45.77**
4	132.6000	0.7356	**45.12**	0.7365	0.8890	**45.11**
5	120.0000	0.7211	**44.42**	0.7211	0.8950	**44.42**
6	108.5000	0.7070	**43.72**	0.7091	0.9321	**43.69**
7	98.2000	0.6979	**42.97**	0.6972	0.8450	**42.98**
8	88.9000	0.6877	**42.23**	0.6864	0.8239	**42.25**
9	80.4000	0.6801	**41.45**	0.6812	0.9411	**41.44**
10	72.8000	0.6708	**40.71**	0.6724	0.9350	**40.69**
20	26.7172	0.6318	**32.52**	0.6330	0.9763	**32.51**
30	9.7987	0.6251	**23.90**	0.6658	0.9750	**23.36**
40	3.6444	0.6237	**15.33**	0.8591	0.9850	**12.55**

Remark:

- **FIR** column is the fit of error of a FIR model.
- **M12**(0.9048) column is the fit of error of Laguerre model which uses an optimal pole of 0.9048 derived from Eq.(7.11) to calculate the fit of error at every model order.
- **DB**(M12/FIR) is the improvement of the fit of error between M12 and FIR columns which is calculated by $DB = -20 * \log_{10}(M12 / FIR)$.
- **Laguerre** column consists of three columns, i.e. ***Fit*** column is the fit of error of Laguerre model according to each optimal pole, ***Pole*** column is the optimal pole computed from Eq.(7.23), and ***DB***(Laguerre/FIR) is the improvement of the fit of error between Laguerre and FIR columns.

A2: Fits of error of the *1st* order system when **ZE=ZV** and data length of **4000** samples. (Fig.8.4(Right))

Model order	FIR	M12 (0.9048)		Laguerre		
		Fit	DB	Fit	Pole	DB
	1.0e-003 *	1.0e-003 *	(M12/FIR)	1.0e-003 *		(Laguerre/FIR)
1	173.4135	5.1140	**30.61**	5.1136	0.9049	**30.61**
2	156.9210	4.6247	**30.61**	4.6247	0.9050	**30.61**
3	141.9999	4.1838	**30.61**	4.1838	0.9059	**30.61**
4	128.4885	3.7853	**30.62**	3.7852	0.8890	**30.62**
5	116.2698	3.4246	**30.62**	3.4246	0.8950	**30.62**
6	105.2122	3.0985	**30.62**	3.0985	0.9321	**30.62**
7	95.1795	2.8033	**30.62**	2.8030	0.8450	**30.62**
8	86.1095	2.5366	**30.62**	2.5365	0.8239	**30.62**
9	77.9039	2.2954	**30.61**	2.2954	0.9411	**30.61**
10	70.4811	2.0778	**30.61**	2.0776	0.9350	**30.61**
20	25.8956	0.7829	**30.39**	0.7810	0.9763	**30.41**
30	9.5107	0.3481	**28.73**	0.3356	0.9750	**29.05**
40	3.5030	0.2397	**23.30**	0.0941	0.9850	**31.42**

A3: Fits of error of the *1st* order system when **ZE≠ZV** with data length of **8000** samples. (Fig.8.6)

Model order	FIR	M12 (0.9048)		Laguerre		
		Fit	DB	Fit	Pole	DB
	1.0e-003 *	1.0e-003 *	(M12/FIR)	1.0e-003 *		(Laguerre/FIR)
1	174.4000	0.3190	**54.76**	0.3158	0.9000	**54.84**
2	157.8000	0.2923	**54.65**	0.2923	0.9000	**54.65**
3	142.8000	0.2665	**54.58**	0.2665	0.9100	**54.58**
4	129.2000	0.2434	**54.50**	0.2433	0.8900	**54.50**
5	116.9000	0.2217	**54.44**	0.2221	0.9300	**54.43**
6	105.8000	0.2019	**54.39**	0.2029	0.9300	**54.34**
7	95.7000	0.1856	**54.25**	0.1853	0.8500	**54.26**
8	86.6000	0.1694	**54.17**	0.1683	0.8400	**54.23**
9	78.4000	0.1555	**54.05**	0.1555	0.9000	**54.05**
10	70.9000	0.1409	**54.03**	0.1416	0.8700	**53.99**
20	26.1000	0.0546	**53.59**	0.0590	0.9600	**52.92**
30	9.6000	0.0290	**50.39**	0.0470	0.9800	**46.20**
40	3.5000	0.0253	**42.82**	0.3277	0.9900	**20.57**

A4: Fits of error of the 2nd order resonant system when **ZE≠ZV** and data length of **4000** samples. (Fig.8.12)

Model order	FIR	M12 (0.5562)		Laguerre			Kautz		
		Fit	DB	Fit	Pole	DB	Fit	Pole	DB
			(M12/FIR)			(Laguerre/FIR)			(Kautz/FIR)
2	1.0897	0.9815	**0.91**	0.9344	0.4550	**1.34**	0.0697	0.80+0.50i	**23.88**
4	0.8933	0.7898	**1.07**	0.7358	0.4700	**1.68**	0.0568	0.80+0.50i	**23.93**
6	0.8749	0.6355	**2.78**	0.5887	0.4970	**3.44**	0.0519	0.80+0.49i	**24.54**
8	0.7525	0.5116	**3.35**	0.4725	0.5106	**4.04**	0.0489	0.79+0.51i	**23.74**
10	0.6358	0.4114	**3.78**	0.3788	0.5188	**4.50**	0.0395	0.79+0.49i	**24.13**
20	0.3583	0.1400	**8.16**	0.1290	0.5900	**8.87**	0.0236	0.75+0.45i	**23.63**
30	0.2149	0.0483	**12.97**	0.0452	0.5000	**13.54**	0.0137	0.73-0.41i	**23.91**
40	0.1220	0.0173	**16.97**	0.0162	0.5200	**17.54**	0.0073	0.67-0.42i	**24.46**

Remark:

- **Kautz** column consists of three columns, i.e. ***Fit*** column is the fit of error of Kautz model according to each optimal pole, ***Pole*** column is the optimal pole computed from Eq.(7.23), and ***DB***(Kautz/FIR) is the improvement of the fit of error between Kautz and FIR columns.

A5: Fits of error of the 2nd order resonant system with zeros at **0.7±0.4i** when **ZE≠ZV** and data length of **4000** samples. (Fig.8.13)

Model order	FIR	M12 (0 4697)		Laguerre			Kautz		
		Fit	DB	Fit	Pole	DB	Fit	Pole	DB
			(M12/FIR)			(Laguerre/FIR)			(Kautz/FIR)
2	0.1444	0.1334	**0.69**	0.1243	0.5800	**1.30**	0.1205	0.61+0.26i	**1.57**
4	0.1415	0.1201	**1.42**	0.1100	0.3500	**2.19**	0.0824	0.57+0.48i	**4.70**
6	0.1215	0.0900	**2.61**	0.0826	0.6000	**3.35**	0.0462	0.67+0.37i	**8.40**
8	0.1028	0.0698	**3.36**	0.0687	0.6000	**3.50**	0.0282	0.63+0.43i	**11.23**
10	0.1008	0.0559	**5.12**	0.0543	0.5800	**5.37**	0.0184	0.67+0.37i	**14.77**
20	0.0532	0.0204	**8.33**	0.0185	0.5200	**9.17**	0.0034	0.66+0.40i	**23.89**
30	0.0283	0.0075	**11.53**	0.0063	0.5600	**13.05**	0.0018	0.66+0.41i	**23.93**
40	0.0166	0.0026	**16.10**	0.0023	0.5342	**17.17**	0.0011	0.65+0.41i	**23.57**
50	0.0098	0.0010	**19.82**	0.0009	0.6010	**20.27**	0.0006	0.65+0.45i	**24.26**

Remark:

- The value (0.4697) in M12 column is the Laguerre pole calculated from Eq.(7.11). Clearly, it differs from the previous one (0.5562) given in A4 because the zeros cause an optimal pole changed to another position.

A6: Fits of error of the system given in Eq.8.3 when **ZE≠ZV** with data length of **8000** samples. (Fig.8.16)

Model order	FIR	M12 (0.0972)		Laguerre			Kautz		
		Fit	DB (M12/FIR)	Fit	Pole	DB (Laguerre/FIR)	Fit	Pole	DB (Kautz/FIR)
2	0.4431	0.4664	**-0.45**	0.3904	-0.3000	**1.10**	0.2666	-0.71+0.50i	**4.41**
4	0.3664	0.3549	**0.28**	0.3044	-0.3800	**1.61**	0.2655	-0.63+0.48i	**2.80**
6	0.3438	0.3538	**-0.25**	0.2758	-0.6000	**1.91**	0.2637	-0.64+0.37i	**2.30**
8	0.2696	0.2818	**-0.38**	0.2564	-0.2300	**0.44**	0.2564	-0.23	**0.44**
10	0.2663	0.2375	**0.99**	0.2373	0.1000	**1.00**	0.2361	0.10+0.15i	**1.05**
20	0.1780	0.1514	**1.41**	0.1371	0.2000	**2.27**	0.1339	0.21+0.13i	**2.47**
30	0.1191	0.0992	**1.59**	0.0677	0.3889	**4.91**	0.0663	0.35+0.27i	**5.09**
40	0.0776	0.0624	**1.89**	0.0318	0.3200	**7.75**	0.0308	0.34+0.25i	**8.03**
50	0.0525	0.0374	**2.95**	0.0145	0.3100	**11.18**	0.0140	0.36+0.14i	**11.48**
60	0.0353	0.0214	**4.35**	0.0066	0.3100	**14.56**	0.0065	0.35+0.14i	**14.70**
70	0.0219	0.0118	**5.37**	0.0031	0.3600	**16.98**	0.0029	0.35-0.15i	**17.56**
80	0.0129	0.0065	**5.95**	0.0014	0.3600	**19.29**	0.0014	0.36-0.07i	**19.29**
90	0.0079	0.0037	**6.59**	0.0007	0.3534	**21.38**	0.0007	0.35-0.08i	**21.05**
100	0.0051	0.0021	**7.71**	0.0004	0.3482	**23.18**	0.0003	0.35-0.08i	**24.61**

A7: Fits of error of the system given in Eq.8.3 when **ZE≠ZV** with data length of **8000** samples. (Fig.8.16)

Model order	Two-stage echo canceler with NB=30						Two-stage echo canceler with NB=50					
	Laguerre			Kautz			Laguerre			Kautz		
(NA+NB)	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB
1-30												
40	0.0538	0.53	**3.18**	0.0070	0.77+0.46i	**20.90**						
50	0.0233	0.59	**7.06**	0.0045	0.65+0.45i	**21.34**						
60	0.0097	0.54	**11.22**	0.0031	0.64+0.45i	**21.13**	0.0223	0.46	**3.99**	0.0018	0.77+0.50i	**25.85**
70	0.0039	0.55	**14.99**	0.0016	0.56+0.40i	**22.73**	0.0090	0.51	**7.72**	0.0011	0.71+0.47i	**25.98**
80	0.0016	0.50	**18.13**	0.0009	0.50+0.35i	**23.46**	0.0037	0.59	**10.85**	0.0006	0.69+0.42i	**26.65**
90	0.0007	0.56	**20.61**	0.0005	0.50+0.24i	**23.94**	0.0014	0.53	**15.03**	0.0004	0.62+0.39i	**25.91**
100	0.0003	0.50	**23.43**	0.0003	0.46+0.27i	**25.01**	0.0006	0.60	**18.59**	0.0003	0.60+0.40i	**25.83**

Remark:

- T**wo-stage echo canceller** is implemented by using the first stage as a FIR model with NB parameters and the second stage as Laguerre or Kautz filter with NA parameters. Thus the total model order is equivalent to NB+NA parameters. The optimal pole can be computed in the same manner where the regression vector is modified in accordance with the discussion given in section 6.6.

Appendix B (Used for section 9)

B1: Fits of error as **the reference set** when ZE≠ZV and the data length of 8000 samples. (Fig.9.3)

Model order	[8000] ze=(10000:17999) & zv=(20000:27999)								
	FIR	M12 (0.0812)		Laguerre			Kautz		
		Fit	DB	Fit	Pole	DB	Fit	Pole	DB
2	0.2094	0.2094	**0.0000**	0.2073	-0.7033	**0.0875**	0.1895	0.80+0.50i	**0.8673**
10	0.1463	0.1371	**0.5641**	0.1366	0.1173	**0.5959**	0.1345	0.14+0.30i	**0.7304**
50	0.1195	0.1192	**0.0218**	0.1191	0.4445	**0.0291**	0.1185	0.53+0.45i	**0.0730**
100	0.1174	0.1172	**0.0148**	0.1175	0.3230	**-0.0074**	0.1145	0.56+0.59i	**0.2173**
200	0.1100	0.1098	**0.0158**	0.1100	0.3137	**0.0000**	0.1063	0.44+0.64i	**0.2972**
300	0.1049	0.1042	**0.0582**	0.1027	0.1489	**0.1841**	0.0994	0.41+0.50i	**0.4678**
400	0.0991	0.0985	**0.0527**	0.0968	0.3236	**0.2040**	0.0931	0.35+0.39i	**0.5425**
500	0.0928	0.0925	**0.0281**	0.0901	0.2998	**0.2565**	0.0882	0.35+0.31i	**0.4416**

B2: Fits of error for **two different segmented data** using **the same dominating pole as given in table B1**, when ZE≠ZV and the data length of 8000 samples. (Fig.9.4)

Model order	[8000] ze=(1:8000) & zv=(30000:37999)							[8000] ze=(30000:37999) & zv=(10000:17999)						
	FIR	M12 (0.0812)		Laguerre		Kautz		FIR	M12 (0.0812)		Laguerre		Kautz	
		Fit	DB	Fit	DB	Fit	DB		Fit	DB	Fit	DB	Fit	DB
2	0.2125	0.2125	**0.0000**	0.2100	**0.1028**	0.1929	**0.8405**	0.2090	0.2090	**0.0000**	0.2075	**0.0626**	0.1905	**0.8050**
10	0.1469	0.1375	**0.5744**	0.1370	**0.6060**	0.1346	**0.7595**	0.1457	0.1359	**0.6048**	0.1354	**0.6368**	0.1327	**0.8118**
50	0.1194	0.1191	**0.0219**	0.1188	**0.0438**	0.1182	**0.0877**	0.1181	0.1178	**0.0221**	0.1172	**0.0664**	0.1168	**0.0961**
100	0.1171	0.1171	**0.0000**	0.1171	**0.0000**	0.1138	**0.2483**	0.1163	0.1163	**0.0000**	0.1158	**0.0374**	0.1122	**0.3117**
200	0.1094	0.1090	**0.0318**	0.1090	**0.0318**	0.1057	**0.2988**	0.1093	0.1092	**0.0080**	0.1089	**0.0318**	0.1061	**0.2581**
300	0.1049	0.1044	**0.0415**	0.1033	**0.1335**	0.0991	**0.4940**	0.1041	0.1037	**0.0334**	0.1027	**0.1176**	0.0998	**0.3664**
400	0.0999	0.0991	**0.0698**	0.0976	**0.2023**	0.0936	**0.5658**	0.1002	0.0994	**0.0696**	0.0973	**0.2551**	0.0946	**0.4995**
500	0.0940	0.0929	**0.1022**	0.0901	**0.3681**	0.0881	**0.5630**	0.0946	0.0944	**0.0184**	0.0905	**0.3849**	0.0891	**0.5203**

B3: T**he new dominating pole** and the fits of error for **the same segmented data used in table B2**, when ZE≠ZV and the data length of 8000 samples. (Fig.9.5)

Model order	[8000] ze=(1:8000) & zv=(30000:37999)						[8000] ze=(30000:37999) & zv=(10000:17999)					
	Laguerre			Kautz			Laguerre			Kautz		
	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB
2	0.2100	-0.7095	**-0.0249**	0.1928	0.80+0.51i	**0.7174**	0.2075	-0.6982	**0.0792**	0.1906	0.80+0.51i	**0.8171**
10	0.1370	0.1193	**0.5705**	0.1347	0.14+0.29i	**0.7175**	0.1354	0.1167	**0.6725**	0.1327	0.14+0.30i	**0.8475**
50	0.1188	0.4777	**0.0510**	0.1182	0.47+0.53i	**0.0950**	0.1173	0.5750	**0.1614**	0.1169	0.48+0.53i	**0.1911**
100	0.1171	0.3357	**0.0222**	0.1141	0.55+0.61i	**0.2476**	0.1158	0.3360	**0.1192**	0.1131	0.28+0.57i	**0.3241**
200	0.1091	0.3090	**0.0714**	0.1056	0.47+0.63i	**0.3546**	0.1091	0.0770	**0.0714**	0.1066	0.51+0.54i	**0.2727**
300	0.1033	0.1493	**0.1335**	0.0996	0.35+0.54i	**0.4503**	0.1027	0.1496	**0.1841**	0.1022	0.40+0.50i	**0.2265**
400	0.0974	0.2482	**0.1503**	0.0934	0.33+0.60i	**0.5145**	0.0977	0.2489	**0.1236**	0.0946	0.35+0.39i	**0.4037**
500	0.0902	0.3010	**0.2468**	0.0885	0.24+0.55i	**0.4121**	0.0906	0.2893	**0.2084**	0.0891	0.35+0.31i	**0.3534**

B4: Fits of error of the two different sizes of data, 4000 and 12000 samples, using **the same dominating pole as given in table B1**, when ZE≠ZV. (Fig.9.6)

Model order	[4000] ze=(10000:13999) and zv=(20000:23999)							[12000] ze=(10000:21999) and zv=(20000:31999)						
	FIR	M12 (0.0812)		Laguerre		Kautz		FIR	M12 (0.0812)		Laguerre		Kautz	
		Fit	DB	Fit	DB	Fit	DB		Fit	DB	Fit	DB	Fit	DB
2	0.2090	0.2090	**0.0000**	0.2076	**0.0584**	0.1904	**0.8096**	0.2110	0.2110	**0.0000**	0.2089	**0.0869**	0.1911	**0.8604**
10	0.1443	0.1361	**0.5082**	0.1357	**0.5337**	0.1339	**0.6497**	0.1471	0.1376	**0.5799**	0.1371	**0.6115**	0.1349	**0.7520**
50	0.1180	0.1177	**0.0221**	0.1172	**0.0591**	0.1170	**0.0739**	0.1195	0.1193	**0.0145**	0.1190	**0.0364**	0.1186	**0.0657**
100	0.1159	0.1160	**-0.0075**	0.1169	**-0.0746**	0.1138	**0.1588**	0.1174	0.1172	**0.0148**	0.1171	**0.0222**	0.1143	**0.2324**
200	0.1099	0.1102	**-0.0237**	0.1100	**-0.0079**	0.1069	**0.2404**	0.1094	0.1093	**0.0079**	0.1093	**0.0079**	0.1058	**0.2906**
300	0.1063	0.1049	**0.1152**	0.1036	**0.2235**	0.1011	**0.4356**	0.1044	0.1038	**0.0501**	0.1024	**0.1680**	0.0985	**0.5053**
400	0.1008	0.1006	**0.0173**	0.0994	**0.1215**	0.0961	**0.4147**	0.0990	0.0982	**0.0705**	0.0959	**0.2763**	0.0921	**0.6275**
500	0.0952	0.0956	**-0.0364**	0.0929	**0.2124**	0.0916	**0.3348**	0.0924	0.0916	**0.0755**	0.0889	**0.3354**	0.0870	**0.5231**

B5: **The new dominating pole** and the fits of error for **the same set of data size used in table B4**, when ZE≠ZV. (Fig.9.7)

Model order	[4000] ze=(10000:13999) and zv=(20000:23999)						[12000] ze=(10000:21999) and zv=(20000:31999)					
	Laguerre			Kautz			Laguerre			Kautz		
	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB
2	0.2076	-0.7015	**0.0584**	0.1904	0.80+0.50i	**0.8096**	0.2089	-0.7029	**0.0869**	0.1911	0.80+0.50i	**0.8604**
10	0.1357	0.1186	**0.5337**	0.1339	0.14+0.30i	**0.6497**	0.1371	0.1180	**0.6115**	0.1349	0.14+0.30i	**0.7520**
50	0.1179	0.6111	**0.0074**	0.1174	0.50-0.51i	**0.0443**	0.1190	0.4447	**0.0364**	0.1184	0.45+0.52i	**0.0803**
100	0.1170	0.4486	**-0.0820**	0.1142	0.55-0.60i	**0.1283**	0.1171	0.3250	**0.0222**	0.1140	0.57+0.60i	**0.2553**
200	0.1100	0.3141	**-0.0079**	0.1074	0.50-0.57i	**0.1999**	0.1092	0.3052	**0.0159**	0.1054	0.50+0.62i	**0.3235**
300	0.1036	0.1481	**0.2235**	0.1014	0.50-0.40i	**0.4099**	0.1024	0.1493	**0.1680**	0.0987	0.40+0.50i	**0.4877**
400	0.0995	0.2490	**0.1127**	0.0961	0.35-0.39i	**0.4147**	0.0960	0.2495	**0.2673**	0.0923	0.32+0.53i	**0.6087**
500	0.0929	0.3096	**0.2124**	0.0916	0.35-0.31i	**0.3348**	0.0890	0.2987	**0.3256**	0.0865	0.23+0.42i	**0.5731**

B5: Fits of error of FIR, M12, Laguerre and Kautz models for the real acoustic echo data, when ZE≠ZV and the data length of 8000 samples. (Fig.9.8)

Model	FIR	M12 (0.0812)		Laguerre			Kautz		
order	Fit	Fit	DB	Fit	Pole	DB	Fit	Pole	DB
2	0.2094	0.2094	**0.0000**	0.2073	-0.7033	**0.0875**	0.1895	0.80+0.50i	**0.8673**
4	0.2094	0.2094	**0.0000**	0.2016	0.5960	**0.3297**	0.1786	0.52+0.56i	**1.3819**
6	0.2089	0.2037	**0.2189**	0.1762	0.5898	**1.4787**	0.1605	0.45+0.36i	**2.2893**
8	0.1768	0.1521	**1.3071**	0.1368	0.1429	**2.2279**	0.1368	0.14+0.02i	**2.2279**
10	0.1463	0.1371	**0.5641**	0.1366	0.1173	**0.5959**	0.1345	0.14+0.30i	**0.7304**
20	0.1225	0.1223	**0.0142**	0.1218	0.2761	**0.0498**	0.1217	0.28	**0.0569**
30	0.1218	0.1212	**0.0429**	0.1204	0.3036	**0.1004**	0.1199	0.42+0.27i	**0.1366**
40	0.1208	0.1200	**0.0577**	0.1194	0.4701	**0.1013**	0.1193	0.54+0.23i	**0.1085**
50	0.1195	0.1192	**0.0218**	0.1191	0.4445	**0.0291**	0.1185	0.53+0.45i	**0.0730**
60	0.1193	0.1191	**0.0146**	0.1189	0.6072	**0.0292**	0.1176	0.45+0.58i	**0.1247**
70	0.1191	0.1189	**0.0146**	0.1187	0.5692	**0.0292**	0.1168	0.53+0.53i	**0.1694**
80	0.1188	0.1188	**0.0000**	0.1184	0.3565	**0.0293**	0.1154	0.49+0.63i	**0.2522**
90	0.1186	0.1182	**0.0293**	0.1177	0.5086	**0.0662**	0.1149	0.50+0.63i	**0.2753**
100	0.1174	0.1172	**0.0148**	0.1175	0.3230	**-0.0074**	0.1145	0.56+0.59i	**0.2173**
200	0.1100	0.1098	**0.0158**	0.1100	0.3137	**0.0000**	0.1063	0.44+0.64i	**0.2972**
300	0.1049	0.1042	**0.0582**	0.1027	0.1489	**0.1841**	0.0994	0.41+0.50i	**0.4678**
400	0.0991	0.0985	**0.0527**	0.0968	0.3236	**0.2040**	0.0931	0.35+0.39i	**0.5425**
500	0.0928	0.0925	**0.0281**	0.0901	0.2998	**0.2565**	0.0882	0.35+0.31i	**0.4416**
600	0.0874	0.0852	**0.2214**	0.0838	0.1800	**0.3653**	0.0828	0.21+0.40i	**0.4696**
700	0.0811	0.0800	**0.1186**	0.0790	0.1600	**0.2279**	0.0778	0.16+0.28i	**0.3608**
800	0.0759	0.0758	**0.0115**	0.0755	0.1000	**0.0459**	0.0735	0.17+0.32i	**0.2791**
900	0.0706	0.0711	**-0.0613**	0.0705	0.0400	**0.0123**	0.0697	0.05+0.25i	**0.1114**
1000	0.0668	0.0666	**0.0260**	0.0666	0.0743	**0.0260**	0.0649	0.05+0.34i	**0.2506**
1100	0.0631	0.0636	**-0.0686**	0.0631	0.0600	**0.0000**	0.0620	0.31i	**0.1528**
1200	0.0601	0.0598	**0.0435**	0.0598	0.1184	**0.0435**	0.0598	0.10+0.30i	**0.0467**
1300	0.0575	0.0571	**0.0606**	0.0571	0.0911	**0.0606**	0.0563	0.10+0.18i	**0.1832**
1400	0.0559	0.0529	**0.4791**	0.0537	0.1000	**0.3488**	0.0528	0.15+0.23i	**0.4956**
1500	0.0526	0.0512	**0.2343**	0.0507	0.1200	**0.3196**	0.0504	0.15+0.13i	**0.3711**
1600	0.0507	0.0490	**0.2962**	0.0485	0.1000	**0.3853**	0.0477	0.12+0.25i	**0.5298**
1700	0.0472	0.0468	**0.0739**	0.0464	0.1000	**0.1485**	0.0450	0.08+0.28i	**0.4146**
1800	0.0452	0.0450	**0.0385**	0.0450	0.0400	**0.0385**	0.0432	0.12+0.33i	**0.3931**
1900	0.0432	0.0431	**0.0201**	0.0426	0.0403	**0.1215**	0.0411	0.10+0.38i	**0.4328**
2000	0.0411	0.0410	**0.0212**	0.0409	0.0650	**0.0424**	0.0394	0.05+0.33i	**0.3669**

B5(Continue): Fits of error of a two-stage echo canceller models for the real acoustic echo data, when ZE≠ZV and the data length of 8000 samples. (Fig.9.9)

Model	Two-Stage Echo Canceller by Laguerre model						Two-Stage Echo Canceller by Kautz model					
	NB=500			NB=1000			NB=500			NB=1000		
	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB	Fit	Pole	DB
1-500												
600	0.0857	0.46	**0.1706**				0.0838	0.38-0.45i	**0.3653**			
700	0.0802	0.24	**0.0969**				0.0792	0.32-0.50i	**0.2059**			
800	0.0756	0.12	**0.0344**				0.0747	0.30-0.35i	**0.1384**			
900	0.0705	0.10	**0.0123**				0.0696	0.15+0.36i	**0.1239**			
1000	0.0665	0.10	**0.0391**				0.0654	0.12+0.47i	**0.1840**			
1100	0.0633	0.05	**-0.0275**	0.0634	0.18	**-0.0412**	0.0622	0.20+0.45i	**0.1248**	0.0621	0.20+0.54i	**0.1388**
1200	0.0596	0.13	**0.0726**	0.0595	0.28	**0.0872**	0.0596	0.17+0.35i	**0.0726**	0.0592	0.32+0.60i	**0.1311**
1300	0.0571	0.10	**0.0606**	0.0570	0.26	**0.0759**	0.0560	0.15+0.20i	**0.2296**	0.0565	0.30+.048i	**0.1524**
1400	0.0537	0.16	**0.3488**	0.0536	0.28	**0.3649**	0.0530	0.20+0.32i	**0.4674**	0.0529	0.24+0.50i	**0.4791**
1500	0.0507	0.20	**0.3196**	0.0516	0.22	**0.1667**	0.0499	0.20+0.30i	**0.4577**	0.0500	0.32+0.37i	**0.4403**
1600	0.0486	0.14	**0.3674**	0.0490	0.26	**0.2962**	0.0474	0.15+0.33i	**0.5846**	0.0479	0.23+0.36i	**0.4934**
1700	0.0466	0.07	**0.1071**	0.0468	0.10	**0.0739**	0.0455	0.10+0.30i	**0.3164**	0.0455	0.18+0.43i	**0.3186**
1800	0.0450	0.07	**0.0480**	0.0449	0.10	**0.0652**	0.0431	0.15+0.40i	**0.4113**	0.0433	0.20+0.45i	**0.3812**
1900	0.0426	0.05	**0.1116**	0.0426	0.08	**0.1168**	0.0415	0.15+0.38i	**0.3535**	0.0415	0.20+0.47i	**0.3487**
2000	0.0409	0.10	**0.0490**	0.0409	0.16	**0.0439**	0.0397	0.08+0.33i	**0.3092**	0.0397	0.10+0.38i	**0.3010**

Appendix C

Matlab script files for “Modeling the Impulse Response of an office room”.

function fit = fit_arx(ze,zv,NN)

```
%FIT_ARX  computes the fit of error of LS-estimates of ARX or FIR model
%
%   FIT = FIT_ARX(ze,zv,NN)
%
%   FIT: returned as the fit of error calculated from ARX model — A(q) y(t) = B(q) u(t-nk) + e(t) —
%          by using cross validation approach.
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   NN: NN = [na nb nk], the orders and the true delay of the model. If na=0, the model will correspond to FIR model.

%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/15

vz=1;  if size(ze)==size(zv), if norm(ze-zv)<eps, vz=0;  end;   end;
maxlength = length(ze(:,1));
NA=NN(1);  NB=NN(2);  NK=NN(3);

PHI=[ ];  PHIV=[ ];  TH=[ ];
[TH,PHI,PHIV]= mo_arxstruc(ze,zv,[NA NB NK]);
nmax = max([NA+1, NB+NK]);
jj=nmax:maxlength;
if vz==0, ysim=PHI*TH; else, ysim=PHIV*TH; end
fit  = norm(ysim-zv(jj,1))/sqrt(length(jj));
```

function [TH,PHI,PHIV] = mo_arxstruc(ze,zv,NN);

```
%MO_ARXSTRUC computes the regression matrix and the estimated parameters of ARX-model.
%
%   [TH,PHI,PHIV] = MO_ARXSTRUC(ze,zv,NN)
%
%   This function was modified from a traditional "ARXSTRUC.M" in order to extract the  regression matrices
%          (PHI and PHIV) and the estimated parameter vector(TH).
%
%   TH: returned as the estimated parameters of the ARX model — A(q) y(t) = B(q) u(t-nk) + e(t) —.
%
%   PHI: returned as a regression matrix which is generated by a set of estimation data (ze).
%
%   PHIV: returned as a regression matrix which is generated by a set of validation data (zv).
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   NN: NN = [na nb nk], the orders and the true delay of the model. If na=0, the model will correspond to FIR model.

%   Modified from arxstruc.m which was written by L. Ljung in Matlab “System Identification Toolbox”
%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/15
z = ze;                    nn = NN;
```

```
[Ncap,nz] = size(z);    nu = nz-1;
[nm,nl]   = size(nn);   [Ncapv,nzv] = size(zv);

if nz>1, na=nn(:,1);nb=nn(:,2:1+nu);nk=nn(:,2+nu:1+2*nu);
else na=nn(:,1); nb=zeros(nm,1);nk=zeros(nm,1);  end;

nma=max(na);     nbkm=max(nb+nk)-ones(1,nu);
nkm=min(nk);     n=nma+sum((nbkm-nkm))+nu;
vz=1;  if size(z)==size(zv),  if norm(z-zv)<eps, vz=0;  end;   end;

% ---Set up default values
maxsdef=idmsize(Ncap,n);       maxsize=maxsdef;

% ---Construct regression matrix
nmax=max(max([na+ones(nm,1) nb+nk]'))-1;
M=10000;  (To avoid “out of memory”)
R=zeros(n);            F=zeros(n,1);
if vz,Rv=zeros(n);  Fv=zeros(n,1);  end;

for k=nmax:M:max(Ncap,Ncapv)  %k=1:25000:4 yields 1
  if min(Ncap,k+M)<k+1, ntz=0; else ntz=1; end
  if min(Ncapv,k+M)<k+1, ntzv=0; else ntzv=1; end
  if ntz,jj=(k+1:min(Ncap,k+M));phi=zeros(length(jj),n);end
    if vz & ntzv,jjv=(k+1:min(Ncapv,k+M));phiv=zeros(length(jjv),n);end
    for kl=1:nma,
      if ntz,phi(:,kl)=-z(jj-kl,1);end
      if vz & ntzv ,phiv(:,kl)=-zv(jjv-kl,1);end
    end
    ss=nma;

    for ku=1:nu
      for kl=nkm(ku):nbkm(ku),
        if ntz,phi(:,ss+kl+1-nkm(ku))=z(jj-kl,ku+1);end
        if vz & ntzv,phiv(:,ss+kl+1-nkm(ku))=zv(jjv-kl,ku+1);end
      end
      ss=ss+nbkm(ku)-nkm(ku)+1;
    end
    if ntz,R=R+phi'*phi; F=F+phi'*z(jj,1);end
    if vz & ntzv,Rv=Rv+phiv'*phiv; Fv=Fv+phiv'*zv(jjv,1);end
end

jj=0;
for j=1:nm
  estparno=na(j)+sum(nb(j,:));
  if estparno>0
    jj=jj+1;
    s=[1:na(j)];
    rs=nma;
    for ku=1:nu
      s=[s,rs+nk(j,ku)-nkm(ku)+1:rs+nb(j,ku)+nk(j,ku)-nkm(ku)];
      rs=rs+nbkm(ku)-nkm(ku)+1;
    end
    RR=R(s,s);  %RR=X'*X
    FF=F(s);    %FF=X'*D
    if vz,RRv=Rv(s,s);FFv=Fv(s);end
    TH=(RR\FF);
  end % if estparno>0
end;
PHI=phi;   PHIV=[ ];  if vz==1, PHIV=phiv;  end;
```

function optimal_pole = M12_pole(impulse_response,delay)

```
%M12_POLE computes the optimal Laguerre pole by minimizing the performance index J as given in section 7.1.
%          It requires a given set of the impulse response of the true system.
```

```
%
%   OPTIMAL_POLE = M12_POLE(impulse_response, delay)
%
%   OPTIMAL POLE: returned as the optimal Laguerre pole which is derived by minimizing the performance index J.
%
%   IMPULSE_RESPONSE: The impulse response of the true system.
%
%   DELAY: The true delay of the system.

%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/15

ir_length = length(impulse_response);
h = impulse_response(:);
square_h = h'*h;

%-----------------------M1 calculation
m1_temp=0;
for i=0:ir_length-1
   m1_temp = m1_temp + i*(h(i+1)^2);
end
M1= m1_temp/square_h;

%-----------------------M2 calculation
h(ir_length+1)=0;   h(ir_length+2)=0;
m2_temp=0;
for i=0:ir_length-1
   m2_temp = m2_temp + ( (h(i+2)-h(i+1)) * ( i*( h(i+2)-h(i+1) ) ) );
end
M2= m2_temp/square_h;

%-----------------------Optimum pole calculation
if delay > 1
  ii = 1:ir_length;
  D3 = (h(ii)'*h(ii+1))/square_h;
  M1 = M1+delay-1;
  M2 = M2+(2*(delay-1)*(1-D3));
end
optimal_pole = (2*M1-1-M2) / ( 2*M1 - 1 + sqrt( 4*M1*M2 - M2^2 - 2*M2 ) );
```

function fit = fit_orthonormal(ze,zv,NN,pole)

```
%FIT_ORTHONORMAL computes the fit of error of LS-estimates of Laguerre or Kautz model.
%
%   FIT = FIT_ORTHONORMAL(ze,zv,NN,pole)
%
%   FIT: returned as the fit of error which is calculated from the Laguerre or Kautz model with FIR or ARX structure
 %          — A(q) y(t) = B(q) u(t-nk) + e(t) — by using cross validation approach.
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   NN: NN = [na nb nk], the orders and the true delay of the model. If na=0, the model will correspond to FIR structure.
%
%   POLE: The dominating pole employed in the orthonormal model, where a real-valued pole corresponds to Laguerre
%           model and a complex-valued pole corresponds to Kautz model.
%           We use the same pole for every model order specified by [na nb] to calculate the fit of error.

%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/15

NA = NN(1);   NB=NN(2);   NK = NN(3)-1;   if NK<0, NK=0; end;
vz = 1; if size(ze)==size(zv), if norm(ze-zv)<eps, vz=0;  end;   end;
```

```
maxlength = length(ze(:,1));
nmax  = max([NA+1, NB+NK+1]);   ii=nmax:maxlength;

if abs(imag(pole)) > 0   %---Kautz model
  aPHI =[ ];  aPHIV=[ ];  bPHI=[ ];  bPHIV=[ ];  PHI=[ ];  PHIV=[ ];
  poles=[ ];   poles(1:NB/2)=pole;
  [bPHI,bPHIV] = phi_gobfilt_arx(ze,zv,poles,0);
  if NA~=0   %---ARX structure
    poles=[ ];      poles(1:NA/2)=pole;
    [aPHI,aPHIV] = phi_gobfilt_arx(ze,zv,poles,1);
    PHI = [aPHI(ii,:), bPHI(ii-NK,:)];
    if vz==1, PHIV=[aPHIV(ii,:),bPHIV(ii-NK,:)]; end;
  else       %---FIR structure
    PHI = bPHI(ii-NK,:);
    if vz==1, PHIV=bPHIV(ii-NK,:); end;
  end;
else  %---Laguerre model
  aPHI =[ ];  aPHIV=[ ];  bPHI=[ ];  bPHIV=[ ];  PHI=[ ];  PHIV=[ ];
  poles=[ ];      poles(1:NB)=pole;
  [bPHI,bPHIV] = phi_gobfilt_arx(ze,zv,poles,0);
  if NA~=0   %---ARX structure
    poles=[ ];      poles(1:NA)=pole;
    [aPHI,aPHIV] = phi_gobfilt_arx(ze,zv,poles,1);
    PHI = [aPHI(ii,:), bPHI(ii-NK,:)];
    if vz==1, PHIV=[aPHIV(ii,:),bPHIV(ii-NK,:)]; end;
  else       %---FIR structure
    PHI = bPHI(ii-NK,:);
    if vz==1, PHIV=bPHIV(ii-NK,:); end;
  end;
end
THETA = PHI\ze(ii,1);
if vz==0, ysim=PHI*THETA; else, ysim=PHIV*THETA; end;
fit = norm(ysim-zv(ii,1))/sqrt(length(ysim));
```

function [mPHI,mPHIV] = phi_gobfilt_arx(ze,zv,poles,NAflag)

```
%PHI_GOBFILT_ARX computes the regression matrices used in estimation/validation process for orthonormal model.
%
%   [mPHI,mPHIV] = phi_gobfilt_arx(ze,zv,poles,NAflag)
%
%   mPHI:  returned as the regression matrix used for estimation process.
%
%   mPHIV: returned as the regression matrix used for validation process.
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   POLES: is a matrix containing the set of chosen poles in a row.
%
%   NAflag: The index used for indicating the type of model structures, i.e. NAflag=1 corresponds to ARX structure, while
%           NAflag=0 corresponds to FIR structure.

%   Modified from gobfilt.m which was written by Brett Ninness (First written 9/8/94)
%   Piya Kovintavewat
%   Revision:1       Date: 1998/10/15

if nargin<4, NAflag=0; end;
vz = 1; if size(ze)==size(zv), if norm(ze-zv)<eps, vz=0; end; end;
ue_ap = ze(:,2);    ye_ap = ze(:,1);
uv_ap = zv(:,2);   yv_ap = zv(:,1);
mPHI = [ ];         mPHIV = [ ];

%---- uap = u filtered through the all pass filter up to pole k-1
```

```
for k = 1:length(poles)
   if ( abs( imag( poles(k) ) ) > 0 )
      denk = [1,-poles(k)];
      %---If the pole is complex we have to include its conjugate automatically
      denk = real(conv(denk,[1,-poles(k)']));
      alpha = 2*real(poles(k))/(1+abs(poles(k))^2);
      %---Kautz Numerator
      numk1 = sqrt((1-alpha^2)*(1+abs(poles(k))^2)*(1-abs(poles(k))^2))*[0 0 1];
      numk2 = sqrt((1+abs(poles(k))^2)*(1-abs(poles(k))^2))*[0 1 -alpha];
      if NAflag==0   %---FIR structure
        phi1  = filter(numk1,denk,ue_ap);      phi2  = filter(numk2,denk,ue_ap);
        mPHI  = [mPHI,phi1(:),phi2(:)];        ue_ap = filter([denk(3:-1:1)'],denk,ue_ap);
        if vz
          phiv1 = filter(numk1,denk,uv_ap);   phiv2 =  filter(numk2,denk,uv_ap);
          mPHIV = [mPHIV,phiv1(:),phiv2(:)];  uv_ap =  filter([denk(3:-1:1)'],denk,uv_ap);
        end;
      else          %---ARX structure
        phi1  = filter(numk1,denk,-ye_ap);     phi2  = filter(numk2,denk,-ye_ap);
        mPHI  = [mPHI,phi1(:),phi2(:)];        ye_ap = filter([denk(3:-1:1)'],denk,ye_ap);
        if vz
          phiv1 = filter(numk1,denk,-yv_ap);  phiv2 =  filter(numk2,denk,-yv_ap);
          mPHIV = [mPHIV,phiv1(:),phiv2(:)];  yv_ap =  filter([denk(3:-1:1)'],denk,yv_ap);
        end;
      end;
    else
      numk = [0 sqrt(1-abs(poles(k))^2)];
      denk = [1,-poles(k)];
      if NAflag==0   %---FIR structure
        phi  = filter(numk,denk,ue_ap);
        mPHI = [mPHI,phi(:)];                  ue_ap=filter([denk(2:-1:1)'],denk,ue_ap);
        if vz
          phiv  = filter(numk,denk,uv_ap);
          mPHIV = [mPHIV,phiv(:)];             uv_ap=filter([denk(2:-1:1)'],denk,uv_ap);
        end;
      else          %---ARX structure
        phi  = filter(numk,denk,-ye_ap);
        mPHI = [mPHI,phi(:)];                  ye_ap=filter([denk(2:-1:1)'],denk,ye_ap);
        if vz
          phiv  = filter(numk,denk,-yv_ap);
          mPHIV = [mPHIV,phiv(:)];             yv_ap=filter([denk(2:-1:1)'],denk,yv_ap);
        end;
      end;
   end;
end;
```

function optimal_pole = arg_max(ze,NN,real_part,imaginary_part)

```
%ARG_MAX computes the optimal dominating pole by minimizing the squared error as given in section 7.2.
%
%   OPTIMAL_POLE = ARG_MAX(ze,NN,real_part,imaginary_part)
%
%   OPTIMAL POLE: returned the optimal pole based on the ARX model — A(q) y(t) = B(q) u(t-nk) + e(t) —
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used  for estimating the model parameters.
%
%   NN: NN = [na nb nk], the orders and the true delay of the model. If na=0, the model will correspond to FIR model.
%
%   REAL_PART: real_part = [started_point:step_size:ended_point] (equally spaced) within unit circle.
%
%   IMAGINARY_PART: imaginary_part = [started_point:step_size:ended_point] (equally spaced) within unit circle.

%   Piya Kovintavewat
%   Revision:1       Date: 1998/10/15
if nargin<4, imaginary_part=0; end;
```

```
NA = NN(1);    NB=NN(2);
NK = NN(3)-1;    if NK<0, NK=0; end;
maxlength =length(ze(:,1));
if length(real_part)>1, step_size_r=real_part(2)-real_part(1); else, step_size_r=0; end;
if length(imaginary_part)>1, step_size_c=imaginary_part(2)-imaginary_part(1); else, step_size_c=0; end;
nmax  = max([NA+1, NB+NK+1]);   ii=nmax:maxlength;   N=length(ii);

a_max = [ ];
for i=1:length(real_part)
  for j=1:length(imaginary_part)
      poles = [ ];     pole = real_part(i)+imaginary_part(j);
      if abs(pole) < 0.99
        aPHI=[ ];  aPHIV=[ ];  bPHI=[ ];  bPHIV=[ ];  PHI=[ ];  PHIV=[ ];
        if abs(imag(pole))>0   %---KAUTZ model
          poles=[ ];      poles(1:NB/2)=pole;
          [bPHI,bPHIV] = phi_gobfilt_arx(ze,ze,poles,0);
          if NA~=0   %---ARX structure
            poles=[ ];      poles(1:NA/2)=pole;
            [aPHI,aPHIV] = phi_gobfilt_arx(ze,ze,poles,1);
            PHI = [aPHI(ii,:),bPHI(ii-NK,:)];
          else       %---FIR structure
            PHI = bPHI(ii-NK,:);
          end;
        else   %---LAGUERRE model
          poles=[ ];      poles(1:NB)=pole;
          [bPHI,bPHIV] = phi_gobfilt_arx(ze,ze,poles,0);
          if NA~=0   %---ARX structure
            poles=[ ];      poles(1:NA)=pole;
            [aPHI,aPHIV] = phi_gobfilt_arx(ze,ze,poles,1);
            PHI = [aPHI(ii,:),bPHI(ii-NK,:)];
          else       %---FIR structure
            PHI = bPHI(ii-NK,:);
          end;
        end;
        Town  = (PHI'*PHI)/N;
        gamma = (ze(ii,1)'*PHI)/N;
        a_max(i,j) = gamma*inv(Town)*gamma';
      else
        a_max(i,j) = 0;
      end;
   end;
end
[yimax,irow] = max(a_max);
[yjmax,jcol] = max(max(a_max));
optimal_pole = (real_part(1)+(irow(jcol)-1)*step_size_r) + (imaginary_part(1)+(jcol-1)*step_size_c);
```

function fit = fit_2stages(ze,zv,NN,pole,firPHI,firPHIV)

```
%FIT_2STAGES computes the fit of error of the two-stage echo canceller
%
%   FIT = FIT_2STAGES(ze,zv,NN,pole)
%
%   FIT: returned as the fit of error which is calculated from the two-stage echo canceller as given in section 6.4, where the
%          first stage is a FIR model of NB parameters and the second stage is an orthonormal model of NA parameters.
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   NN: NN = [na nb nk], the orders of each stage(na, nb) and the true delay(nk) of the model.
%
%   POLE: The optimal pole used in an orthonormal model, where real-valued pole corresponds to Laguerre model and
%          complex-valued pole corresponds to Kautz model. We use the same pole for every model order specified by NA.
%   [  firPHI, firPHIV ]:  The regression matrices corresponding to FIR model which is computed by the first stage NB
```

```
%          parameters from [firPHI, firPHIV] = phi_gobfilt_arx(ze,zv,poles,0), where poles = zeros(1:NB).

%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/15

NA = NN(1);  NB = NN(2);  NK = NN(3)-1; if NK<0, NK=0; end;
vz=1; if size(ze)==size(zv), if norm(ze-zv)<eps, vz=0; end;  end;
maxlength = length(ze(:,1));   nmax  = max([1, NA+NB+NK+1]);
ii = nmax:maxlength;           N = length(ii);

poles = [ ];  if abs(imag(pole))>0, poles = ones(1,NA/2)*pole; else, poles = ones(1,NA)*pole; end;
bPHI = [ ];  bPHIV = [ ];  [bPHI,bPHIV] = phi_gobfilt_arx(ze,zv,poles,0);
PHI   = [ ];  PHI = [firPHI(ii-NK,:),  bPHI(ii-NK-NB,:)];
if vz==1, PHIV = [ ];  PHIV = [firPHIV(ii-NK,:), bPHIV(ii-NK-NB,:)];
THETA = PHI\ze(ii,1);
if vz==0,  ysim=PHI*THETA; else, ysim=PHIV*THETA; end;
fit = norm(ysim-zv(ii,1))/sqrt(length(ysim));
```

function optimal_pole = arg_max_2stages(ze,NN,real_part,imaginary_part,firPHI,firPHIV)

```
%ARG_MAX_2STAGES computes the optimal pole for using in an orthonormal model of the two-stage echo canceller,
%          where the first stage is a FIR model of NB parameters and the second stage is an orthonormal model with FIR
%          structure of NA parameters. The optimal pole is calculated by minimizing the squared error given in section 7.2.
%
%   OPTIMAL_POLE = ARG_MAX_2STAGES(ze,NN,real_part,imaginary_part,firPHI,firPHIV)
%
%   OPTIMAL POLE: returned the optimal pole of the two-stage echo canceller.
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   NN: NN = [na nb nk], the orders of each stages(na,nb) and the true delay(nk) of the model.
%
%   REAL_PART: real_part = [started_point:step_size:ended_point] (equally spaced) within unit circle.
%
%   IMAGINARY_PART: imaginary_part = [started_point:step_size:ended_point] (equally spaced) within unit circle.
%
%   [ firPHI, firPHIV]:  The regression matrices of FIR model which is computed by the first stage NB parameters from
%          [firPHI, firPHIV] = phi_gobfilt_arx(ze,zv,poles,0), where poles = zeros(1:NB).
%
%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/15


NA = NN(1);   NB = NN(2);   NK = NN(3)-1; if NK<0, NK=0; end;
maxlength = length(ze(:,1));
if length(real_part)>1, step_size_r = real_part(2)-real_part(1); else, step_size_r = 0; end;
if length(imaginary_part)>1, step_size_c = imaginary_part(2)-imaginary_part(1); else, step_size_c = 0; end;

nmax  = max([1, NA+NB+NK+1]);  ii=nmax:maxlength;  N=length(ii);
a_max = [ ];
for i=1:length(real_part)
   for j=1:length(imaginary_part)
      poles = [ ];      pole = real_part(i)+imaginary_part(j);
      if abs(pole) < 1
         if abs(imag(pole))>0
            poles = ones(1,NA/2)*pole;
         else
            poles = ones(1,NA)*pole;
         end;
         bPHI = [ ];  bPHIV = [ ];  [bPHI,bPHIV] = phi_gobfilt_arx(ze,ze,poles,0);
         PHI   = [ ];  PHI = [firPHI(ii-NK,:), bPHI(ii-NK-NB,:)];
         Town  = (PHI'*PHI)/N;
         gamma = (ze(ii,1)'*PHI)/N;
         a_max(i,j) = gamma*inv(Town)*gamma';
```

```
      else
        a_max(i,j) = 0;
      end;
   end;
end;
[yimax,irow] = max(a_max);
[yjmax,jcol] = max(max(a_max));
optimal_pole = (real_part(1)+(irow(jcol)-1)*step_size_r)+(imaginary_part(1)+(jcol-1)*step_size_c)
```

function [order, fit, selected_poles, Energy]
= model_reduction(ze,zv,true_delay,real_part,imaginary_part,no_iteration)

```
%MODEL_REDUCTION searches the ordering of the optimal dominating poles based on the criteria of the system energy
%          in the sense that the first sections contribute most to the overall impulse response of the original system in a
%          quadratic sense as discussed in section 7.4.2.
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   TRUE_DELAY: the true delays of the system
%
%   REAL_PART: real_part = [started_point:step_size:ended_point] (equally spaced) within unit circle.
%
%   IMAGINARY_PART: imaginary_part = [started_point:step_size:ended_point] (equally spaced) within unit circle.
%
%   NO_ITERATION: The maximum number of iteration used in this function in order to stop the process.(Default=50)
%
%   ORDER: The minimum model order obtained from this function, based on the system energy criterion.
%
%   FIT: The corresponding fit of error at obtained order and selected poles
%
%   SELECTED_POLES: The ordering of the chosen poles in the order of importance..
%
%   ENERGY: The accumulated energy at each order calculated from the squared absolute value of coefficient weights.

%   Piya Kovintavewat
%   Revision:1      Date: 1998/10/22

if nargin<6,  no_iteration = 50;  end;
if nargin<5,  imaginary_part = 0; end;
NK = true_delay-1; if NK<0, NK=0; end;
vz=1; if size(ze)==size(zv), if norm(ze-zv)<eps, vz=0; end;  end
maxlength = length(ze(:,1));
if length(real_part)>1, step_size_r=real_part(2)-real_part(1); else, step_size_r=0; end;
if length(imaginary_part)>1, step_size_c=imaginary_part(2)-imaginary_part(1); else, step_size_c=0; end;
selected_poles = [ ];  Energy = [ ];  sel_NB=0;

for number_loop=1:no_iteration
   nmax = max([1, sel_NB+1+NK+1]);  ii = nmax:maxlength;

   %---Compute the energy of a first order section (only real poles)
   temp_Energy = [ ];  temp_total_energy = [ ];
   for j=1:length(real_part)
      choice_pole = real_part(j);
      poles = [ ];     poles=[selected_poles, choice_pole];
      bPHI = [ ];     bPHIV=[];  [bPHI,bPHIV] = phi_gobfilt_arx(ze,ze,poles,0);
      THETA = bPHI(ii-NK,:)\ze(ii,1);
      temp_Energy(j) = abs(THETA(sel_NB+1))^2;
      temp_total_energy(j) = sum(THETA.^2);
   end
   [Ymax,imax]  = max(temp_Energy);   Emax=Ymax;
   total_energy1 = temp_total_energy(imax);
   r_pole = real_part(imax);
```

```
   %---Compute the energy of two additional first order sections for either complex-conjugated pole pair or
   %    2 identical real-valued poles.
   temp_Energy = [ ];   temp_total_energy = [ ];
   nmax = max([1, sel_NB+2+NK+1]);   ii = nmax:maxlength;
   for i=1:length(real_part)
      for j=1:length(imaginary_part)
         pole = real_part(i)+imaginary_part(j);
         if abs(pole) < 0.99
           if abs(imag(pole))==0, pole=ones(1,2)*pole; end;
           poles = [ ];  poles = [selected_poles, pole];
           bPHI = [ ];  bPHIV=[];  [bPHI,bPHIV] = phi_gobfilt_arx(ze,ze,poles,0);
           THETA = bPHI(ii-NK,:)\ze(ii,1);
           temp_Energy(i,j) = sum(abs(THETA(sel_NB+1:sel_NB+2)).^2);
           temp_total_energy(i,j) = sum(THETA.^2);
         else
           temp_Energy(i,j) = 0;  temp_total_energy(i,j) = 0;
         end;
      end;
   end;
   [yimax,irow]  = max(temp_Energy);
   [Ymax,imax]  = max(max(temp_Energy));
   total_energy2 = temp_total_energy(irow(imax),imax);
   c_pole = (real_part(1)+(irow(imax)-1)*step_size_r)+(imaginary_part(1)+(imax-1)*step_size_c);
   if abs(imag(c_pole))==0,  c_pole = ones(1,2)*c_pole;  end;

   %---Compare the increasing energy of real pole and that of complex pole
   if Ymax>Emax
     if isempty(Energy),  Energy = total_energy2;  else, Energy = [Energy,total_energy2]; end;
     selected_poles = [selected_poles, c_pole];      sel_NB = sel_NB+2;
   else
     if isempty(Energy),  Energy = total_energy1;  else, Energy = [Energy,total_energy1]; end;
     selected_poles = [selected_poles, r_pole];       sel_NB = sel_NB+1;
   end;

   %---Check an additional energy
   if length(Energy)>1,  Threshold = (Energy(length(Energy))-Energy(length(Energy)-1));  end;
   if (length(Energy)>1) & (Threshold < 0.001)  %---Less than 0.1% of the previous energy
     selected_poles = selected_poles(1:length(selected_poles)-1);
     Energy = Energy(1:length(Energy)-1);     break;
   end;
end;

%---Compute the order and the fit of error from the resulting optimal poles
real_p       =  selected_poles(find(abs(imag(selected_poles))==0));
complex_p =  selected_poles(find(abs(imag(selected_poles))>0));
order         =  length(real_p) + 2*length(complex_p);
poles         =  selected_poles;
nmax = max([1, order+NK+1]);  ii=nmax:maxlength;
bPHI = [ ];   bPHIV = [ ];          [bPHI,bPHIV] = phi_gobfilt_arx(ze,zv,poles,0);
THETA = bPHI(ii-NK,:)\ze(ii,1);
if vz==0,   ysim=bPHI(ii-NK,:)*THETA;  else,  ysim = bPHIV(ii-NK,:)*THETA;  end;
fit = norm(ysim-zv(ii,1))/sqrt(length(ysim));
```

function [mPHI, mPHIV, GAMMA] = orthobase(ze, zv, poles, NAflag, w, T, NK)

```
%ORTHOBASE computes the regression matrices used in estimation/validation process and the matrix used for calculating
%          estimated model frequency response for orthonormal model.
%
%   [ mPHI, mPHIV, GAMMA ] = orthobase( ze, zv, poles, NAflag, w, T, NK)
%
%   mPHI:  returned as the regression matrix used for estimation process.
%
%   mPHIV: returned as the regression matrix used for validation process.
%   GAMMA:  Matrix such that the estimated model frequency response = GAMMA*THETA;
```

```
%
%   ZE: the output-input data, ZE=[ye ue] with ye and ue as column vectors, used for estimation process.
%
%   ZV: the output-input data, ZV=[yv uv] having the same size as ZE, used for validation process.
%
%   POLES: is a matrix containing the set of chosen poles in a row.
%
%   NAflag: The index used for indicating the type of model structures, i.e. NAflag=1 corresponds to ARX structure, while
%           NAflag=0 corresponds to FIR structure.
%
%   W: Vector of frequencies to evaluate freq. response at NOT in normalized, but in real freq.
%
%   T: Normalized sampling time interval.
%
%   NK: The true delay of the system.

%   Modified from orthobase.m which was written by Brett Ninness (First written 9/8/94).
%   Piya Kovintavewat
%   Revision:1      Date: 1998/08/08

if nargin<7, NK = 0;    end;
if nargin<6, T    = 1;    end;
if nargin<5, w    = [ ];  end;
if nargin<4, Naflag=0; end;   Aflag = NAflag;
NK=NK-1; if NK<0, NK=0; end;
vz=1; if size(ze)==size(zv), if norm(ze-zv)<eps, vz=0; end; end

mPHI = [ ];    mPHIV = [ ];  GAMMA = [ ];
ww     = exp(j*w*T);
ww_NK=exp(-j*w*T*NK);

%-------------------------Initialize the construction
num  = sqrt(1-abs(poles(1))^2);
den   = [1,-poles(1)];

if abs(imag(poles(1))) > 0
  %------- If a pole is complex we have to include its conjugate
  den   = conv(den,[1,-poles(1)']);
  alpha = 2*real(poles(1))/(1+abs(poles(1))^2);

  if Aflag==0  %--FIR structure
    phi     = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(1))^2)), den, ze(:,2));          mPHI     = [mPHI, phi(:)];
    if vz,  phiv  = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(1))^2)), den, zv(:,2));  mPHIV  = [mPHIV, phiv(:)];   end;
  else  %--ARX structure
    phi     = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(1))^2)), den, -ze(:,1));          mPHI     = [mPHI, phi(:)];
    if vz,  phiv  = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(1))^2)), den, -zv(:,1));  mPHIV  = [mPHIV,phiv(:)];  end;
  end;
  gamma = polyval(num*sqrt((1-alpha^2)*(1+abs(poles(1))^2)), ww)./polyval(den,ww);  GAMMA = [GAMMA, gamma(:)];

  if Aflag==0  %--FIR structure
    phi     = dlsim(num*sqrt(1+abs(poles(1))^2)*[1,-alpha], den, ze(:,2));          mPHI    = [mPHI, phi(:)];
    if vz,  phiv  = dlsim(num*sqrt(1+abs(poles(1))^2)*[1,-alpha], den, zv(:,2));  mPHIV = [mPHIV,phiv(:)];    end;
  else   %--ARX structure
    phi     = dlsim(num*sqrt(1+abs(poles(1))^2)*[1,-alpha],den,-ze(:,1));            mPHI     = [mPHI, phi(:)];
    if vz,  phiv  = dlsim(num*sqrt(1+abs(poles(1))^2)*[1,-alpha],den,-zv(:,1));    mPHIV  = [mPHIV,phiv(:)];  end;
  end;
  gamma = polyval(num*sqrt(1+abs(poles(1))^2)*[1,-alpha], ww)./polyval(den, ww);   GAMMA = [GAMMA, gamma(:)];

  %-------  Now put complex-conjugated pole automatically for the next iteration.
  num = num*[-poles(1)',1];

else
  if Aflag==0  %--FIR structure
    phi  = dlsim(num, den, ze(:,2));            mPHI   = [mPHI, phi(:)];
    if vz, phiv = dlsim(num, den, zv(:,2));    mPHIV = [mPHIV, phiv(:)]; end;
  else  %--ARX structure
```

```
    phi    = dlsim(num, den, -ze(:,1));          mPHI    = [mPHI, phi(:)];
    if vz, phiv = dlsim(num, den, -zv(:,1));   mPHIV = [mPHIV, phiv(:)];   end;
  end;
  gamma = (polyval(num, ww)./polyval(den, ww)).*ww_NK;   GAMMA = [GAMMA, gamma(:)];
end;

%------------------------- Now iterate through for as many poles as specified.
for k = 2:length(poles)
  if abs(imag(poles(k))) > 0
    %-------  First test to see if the pole is complex or not.
    num   = real(sqrt((1-abs(poles(k))^2)/(1-abs(poles(k-1))^2))*conv(num, [-poles(k-1),1]));
    den    = conv(den,[1,-poles(k)]);
    %------- If a pole is complex we have to include its conjugate automatically.
    den    = real(conv(den, [1,-poles(k)']));
    alpha = 2*real(poles(k))/(1+abs(poles(k))^2);

    if Aflag==0  %--FIR structure
      phi     = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(k))^2)), den, ze(:,2));          mPHI    = [mPHI, phi(:)];
      if vz,  phiv  = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(k))^2)), den, zv(:,2));    mPHIV = [mPHIV, phiv(:)];  end;
    else  %--ARX structure
      phi     = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(k))^2)),den,-ze(:,1));           mPHI   = [mPHI, phi(:)];
      if vz,  phiv  = dlsim(num*sqrt((1-alpha^2)*(1+abs(poles(k))^2)), den, -zv(:,1));   mPHIV = [mPHIV, phiv(:)];  end;
    end;
    gamma = polyval(num*sqrt((1-alpha^2)*(1+abs(poles(k))^2)),ww)./polyval(den,ww); GAMMA=[GAMMA, gamma(:)];

    if Aflag==0  %--FIR structure
      phi     = dlsim(sqrt(1+abs(poles(k))^2)*conv(num,[1,-alpha]), den, ze(:,2));          mPHI    = [mPHI, phi(:)];
      if vz,  phiv  = dlsim(sqrt(1+abs(poles(k))^2)*conv(num,[1,-alpha]), den, zv(:,2));   mPHIV = [mPHIV, phiv(:)]; end;
    else   %--ARX structure
      phi     = dlsim(sqrt(1+abs(poles(k))^2)*conv(num,[1,-alpha]), den, -ze(:,1));          mPHI     = [mPHI, phi(:)];
      if vz,  phiv  = dlsim(sqrt(1+abs(poles(k))^2)*conv(num,[1,-alpha]), den, -zv(:,1));  mPHIV  = [mPHIV, phiv(:)]; end;
    end
    gamma  = polyval(sqrt(1+abs(poles(k))^2)*conv(num, [1,-alpha]), ww)./polyval(den, ww);
    GAMMA = [GAMMA, gamma(:)];

    %------- Now put complex-conjugated pole automatically for the next iteration.
    num = conv(num,[-poles(k), 1]);

  else
    num = real(sqrt((1-abs(poles(k))^2)/(1-abs(poles(k-1))^2))*conv(num,[-poles(k-1)', 1]));
    den  = conv(den,[1, -poles(k)]);

    if Aflag==0  %--FIR structure
      phi     = dlsim(num,den,ze(:,2));          mPHI    = [mPHI, phi(:)];
      if vz,  phiv=dlsim(num,den,zv(:,2));      mPHIV = [mPHIV, phiv(:)];   end;
    else   %--ARX structure
      phi     = dlsim(num,den,-ze(:,1));          mPHI    = [mPHI, phi(:)];
      if vz,  phiv = dlsim(num,den,-zv(:,1));   mPHIV = [mPHIV, phiv(:)];   end;
    end;
    gamma = (polyval(num, ww)./polyval(den, ww)).*ww_NK;    GAMMA = [GAMMA, gamma(:)];
  end;
end
```