The Pennsylvania State University
The Graduate School

SPARSE LINEAR TIME INVARIANT SYSTEM IDENTIFICATION USING WEIGHTED LASSO

A Thesis in
Electrical Engineering
by
Fatih Tutuk

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2014

The thesis of Fatih Tutuk was reviewed and approved* by the following:

Constantino Lagoa
Professor of Electrical Engineering
Thesis Advisor

Kenneth Jenkins
Professor of Electrical Engineering

Kultegin Aydin
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School.

Abstract

In this thesis, the identification of a single-input single-output (SISO) linear time invariant (LTI) dynamical system from a finite set of completely known input signals and noisy output signals is studied. The identification problem studied aims at estimating the transfer function coefficients, the order of the system, and the standard deviation of the measurement noise. Estimations are performed by employing an existing linear shrinkage method, which is the modified least absolute shrinkage and selection operator (LASSO) [1]. In the modified LASSO, the transfer function coefficients and the standard deviation of the measurement noise are estimated by minimizing the ℓ_2 norm of the error with an ℓ_1 norm penalty on the transfer function coefficients. Owing to the nature of the ℓ_1 norm penalty, the true order of the system is obtained by sparsifying the parameter vector, which contains the transfer function coefficients. An iterative algorithm is proposed to solve the modified LASSO optimization problem. We do not give the true order information of the system to the algorithm. This leads the problem of having a sparse and a correct parameter vector estimate at the same time, which causes wrong order estimation. In order to overcome this issue, we introduce the use of two types of weight matrices. The first weight type is the iterative reweighting for the LASSO, which is a well-known weighting for ℓ_1 norm minimization problems [2]. The second weight type is a fixed weighting containing constant numbers. In this context, we show the applicability of the fixed weighting for sparsification. Finally, we test the validation of the proposed algorithm by synthetic randomly generated bounded-input bounded-output (BIBO) stable systems with noisy measurements.

Table of Contents

List of Figures

List of Tables

List of Symbols

Y	Output vector
X	Input-output data design matrix
p	Number of measurements
n	Order of the discrete linear time invariant system
r	Actual number of variables
s	Given number of variables
β	Parameter vector
ϵ	Zero mean Gaussian white noise
σ	Standard deviation of the noise
λ	The penalty parameter on the ℓ_1 norm
θ	The bound on the ℓ_1 norm constraint
W	Weight matrix
γ	Reweighting parameter
μ	Convergence tolerance
ϕ	Elements in the parameter vector

Acknowledgments

I am very grateful to my advisor, Prof. Constantino Lagoa, whose excellent guidance and invaluable constructive criticism helped me throughout my master's study. I also thank my committee member, Prof. Kenneth Jenkins, for his precious time and suggestions. I would also like to thank my friends in our research laboratory for their countless hours of collaboration and help. In addition, I thank all my friends whom I met during my master's education for their fellowship and understanding.

Most importantly, I would like to express my gratitude to my family for their boundless love and support, and for their encouragement that helped me to arrive where I am today. Finally, I would like to give special thanks to my master's education sponsors, The Ministry of National Education of Turkey and The Turkish Petroleum Corporation, for giving me the opportunity to advance my career.

Dedication

I dedicate this thesis with a special feeling of gratitude to my loving parents, Hatice Tutuk and Ömer Tutuk.

Chapter 1

Introduction

Identification of systems such as control systems, economic systems, and chemical reactions is achievable with high accuracy despite the difficulties in analysis arising from inherent complexities and environmental noise. Although linear time invariant (LTI) systems can be thought of as simple types of dynamical systems, they can be employed for approximating many real systems with high accuracy [3]. Henceforth, the identification of LTI systems has arisen as a fundamental concern in modern engineering theory and applications over the years.

Studies show that the identification of an LTI system can be performed in three ways: state-space identification, finite impulse response identification, or transfer function estimation [3]. In this dissertation, we focus on transfer function coefficients estimation. Since almost every real system is contaminated by environmental and/or non-environmental disturbances, as illustrated in Figure 1.1, we also aim at estimating the standard deviation of the present noise on the output data. We deal with estimations by employing an existing linear shrinkage method, which is the modified least absolute shrinkage and selection operator (LASSO) [1]. In the modified LASSO, the transfer function coefficients and the standard deviation of the measurement noise are estimated by minimizing the ℓ_2 norm of the error with an ℓ_1 norm penalty on the transfer function coefficients. Moreover, owing to the nature of ℓ_1 norm penalization, the true order of the system is revealed. Before presenting a brief literature review, this chapter provides a general background on system identification as follows.

If a data set is completely or partially known for a complex system, and the underlying behavior of the system is not known analytically, then the closest analytical model for the concerned system can be estimated by experimenting with mathematical models. After defining the structural model of the system, the model can be estimated by an appropriate method in order to accomplish the identification. This process is known as system identification. In other words, system identification allows the building of a mathematical description of a system from given measured data. Once the dynamical behavior of the system is obtained by means of identification, the system can be represented by mathematical equations, graphs, curves, and tables. Despite being an effective tool for model analysis, system identification suffers from the following three major problems: the data record, the model structure, and the solution method [3].

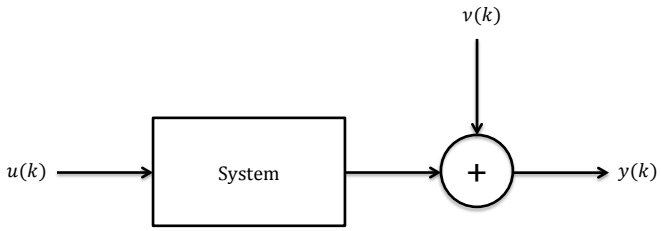

Figure 1.1. Disturbed Dynamical System

The first problem arises in obtaining a data set by means of experiments, which may not always be straightforward due to factors such as technical problems, budget and time availability. The second problem is constructing the appropriate mathematical structure. In other words, the model should be compatible with the system's behavior within a permissible tolerance. For example, the determination of the coefficients of reducing poverty and rising welfare in an economics problem may require a model structure concerned with the impact of environment dynamics, e.g., wars or earthquakes. On the other hand, in aerospace, identifying an optimal control system for a plane, where the purpose is to find the most efficient path from one state to another state, may not require including the environment dynamics as in the economy example. Consequently, choosing an appropriate model is one of the obstacles in achieving the goal of identification. The importance of the model selection and model validation is addressed in many studies such as [4], [5]. The third problem is solving the mathematical structure accurately and

efficiently, i.e., computationally fast. To deal with the third problem, identification is performed through many techniques as diverse as probabilistic, statistical, and control theories.

Identification methods are mainly divided into two parts, namely the parametric and the non-parametric method of system identification [6], which are defined as follows. Parametric methods includes the system's parameters explicitly by means of equations. They use parametric relationships between the input and the output as fixed order transfer functions or state-space representations. Non-parametric system identification implicitly provides the behaviors of the system. This means the relationship between inputs and the correspondence responses is conveyed via graphs or tables such as impulse responses or similar frequency responses. For example, consider an LTI system, one of whose features can be identified from its impulse response. Then its complete behavior can be revealed completely. As a result, the system is considered to be identified. Note that the identification of LTI systems in this study is a type of non-parametric system identification.

1.1 Literature Review

Studies of LTI system identification have increased consistently during the last decades in statistical studies, in signal theory, and especially in control engineering. In this part, we will mention only some of the most significant contributions to the field of LTI system identification. Reader is recommended to read the comprehensive survey of [7] for a broader perspective. All of the methods mentioned here only approximate the characteristics of a system, so there is no actual or exact identification of a real system or a problem. To achieve the closest approximation, one needs to try out various model structures and identification methods. Studies on LTI system identification can be mainly categorized into two parts: subspace model identification (SMI) which covers state-space identification and prediction error methods (PEM) which cover order and transfer function parameter estimations.

SMI based LTI system identification aims at directly estimating state-space representation of the system. The identification of state-space started through the development of state-space representation by Kalman and Hu [8]. The state-space representation enabled identification of the system from the impulse response function. This method was

improved by Kung's algorithm [9] for state space realization, which was based on singular value decomposition (SVD). SVD allowed extracting true order information of the system from data with less tuning parameter. Then, the two most traditional LTI system state-space identification techniques were proposed: the prediction error variance method (PEV) [10] and the instrumental variable method (IVM) [11]. IVM is generalized, yielding subspace-based LTI state-space system identification [12]. Subspace research has been developed over the years with powerful identification methods [13]. Compared to PEM, SMI is very straightforward in terms of computational complexity, especially, when input-output measurement and/or the order of the system is large [14]. In SMI methods, there is not any cost-function to be optimized as in PEM. Instead, the solution is based on geometric properties of signal spaces. However, recent studies show that SMI are biased under closed-loop, and estimations are not as correct as PEM estimations [15].

PEM based LTI system identification aims to find transfer function parameters. A statistical method is applied to the linear model structures such as autoregressive exogenous (ARX) models or autoregressive moving average exogenous (ARMAX) models [10], which contain input-output data. A cost-function is created to be optimized with respect to the selected model. By tuning model parameters, the prediction error for a given data set is minimized, yielding the correct estimation of the transfer function parameters [14].

PEM based LTI system identification was introduced through the use of the maximum likelihood estimation framework [16]. The maximum likelihood estimation principle was extended by Akaike Information Criterion (AIC) [17] due to the problem of choosing the order of the system. Afterwards, AIC was slightly modified by Bayesian Information Criterion (BIC) [18]. These statistical methods have been developed over the years into shrinkage methods, such as the least absolute shrinkage and selection operator (LASSO) [19], the smoothly clipped absolute deviation (SCAD) [20], and Elastic Net [21].

In this study, the identification of SISO discrete LTI systems is performed in the PEM framework. We restrict ourselves to SISO and discrete LTI systems. The modified LASSO [1] is used for obtaining transfer function parameters as well as noise level. Moreover, true order information is obtained via weighted modified LASSO, which will be discussed in more detail further on in the thesis. The reasons for using the modified LASSO in this thesis can be given as follows.

First, many PEM approaches could be used for the estimation of transfer function coefficients from noisy measurement. However, this study also aims to obtain the standard deviation of the measurement noise. Although the subspace identification is very fast compared to iteratively solved LASSO optimization problems, it is not used in this study since it does not provide the standard deviation of the noise corrupting the output signal.

Second, before the modified LASSO, the estimation of the parameter coefficients and noise level at the same time was proposed in some studies such as [22]. In [22], the estimation of the parameter coefficients and noise level was performed by maximizing their joint log-likelihood. It is stated in [23], and [24] that the proposed method may give a bias for noise level estimation. However, the bias problem for estimating the noise level is handled by Sun et al. [1], which reveals that an iterative algorithm proposed in [23] for the LASSO is equivalent to jointly minimizing Huber's concomitant loss function with the ℓ_1 penalty as in [24], which guarantees the convergence.

Third, the LASSO can have parameter estimation bias, and it may not satisfy some oracle properties as stated in [20], [25], [26]. The modified LASSO, in contrast, helps eliminate the estimation bias by means of the least-squares estimation after the model selection. In addition, it provides estimation under some regularity conditions that satisfies some certain oracle properties [1].

Last but not least, the modified LASSO is jointly convex [1], so it is easy to implement. While the LASSO utilizes convex optimization, the AIC and the BIC methods can have a serious computational load since they are based on a combinatorial search scheme [27].

1.2 Contributions

This thesis contributes a method for identification of the transfer function coefficients of a single-input single-output (SISO) discrete LTI system and the estimation of the standard deviation of the measurement noise. In addition, the true order estimation is also performed by sparsification of the parameter vector, which contains the transfer function coefficients.

Estimations are performed by employing an existing linear shrinkage method, which is the modified least absolute shrinkage and selection operator (LASSO) [1]. In the modified LASSO, the transfer function coefficients and the standard deviation of the measurement noise are estimated by minimizing the ℓ_2 norm of the error with an ℓ_1 norm penalty on the transfer function coefficients. Owing to the nature of the ℓ_1 norm penalty, the true order of the system is estimated by sparsifying the parameter vector.

An iterative algorithm is proposed to solve the modified LASSO optimization problem. We do not give true order information of the system to the algorithm. This leads to the problem of having a correct and a sparse parameter vector estimate at the same time, which causes wrong order estimation. In order to overcome this issue, we introduce the use of two types of weight matrices. The first weight type is the iterative reweighting for the LASSO, which is a well-known weighting for ℓ_1 norm minimization problems [2]. The second weight type is a fixed weighting containing constant numbers. We show the applicability of the fixed weighting for sparsification. Finally, we test the validation of the proposed algorithm by synthetic randomly generated bounded-input bounded-output (BIBO) stable systems with noisy measurements.

1.3 Thesis Overview

The rest of this dissertation is organized as follows:

In Chapter 2, basic definitions for further usage are given. Then, the chapter details the concept of sparsity by explaining why the best sparsification approach for ℓ_0 norm minimization is an NP-hard problem. After this, this chapter briefly reviews some of the existing variable selection techniques. Finally, the LASSO is explained in detail.

In Chapter 3, the formal definition of the problem that we study in this dissertation is explained. The appropriate mathematical model construction for a SISO discrete LTI system for p measurement is detailed. In addition, Chapter 3 provides the modified LASSO optimization problem in order to solve the constructed model.

In Chapter 4, the calculations of a given noise's standard deviation and the parameter vector are explained. Additionally, weight matrices and weighted modified LASSO are presented. Finally, this chapter proposes an iterative algorithm for the optimization problem.

In Chapter 5, the experimental simulation results of the proposed algorithm is presented. The contribution of weight matrices in terms of error and sparsification are emphasized in this chapter. Finally, the chapter concludes with a few closing remarks, discussions, and directions for future research.

Chapter 2

Preliminaries

2.1 Basic Definitions

In this section, basic definitions for further usage in this dissertation are introduced.

Definition 1: Vector Norm

Given an n-dimensional of a real or complex vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

the norm $\|x\|_p$ is a mapping from R^n to R defined as

$$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p} \quad \text{for} \;\; 1 \leq p < \infty$$

Definition 1.1: ℓ_0 Norm (Pseudo Norm)

The ℓ_0 norm is defined as $\|x\|_0 = \#(n|x_n \neq 0)$, which is the number of non-zero elements in the vector $x \in R^n$. Many optimization problems try to minimize the ℓ_0 norm subject to constraints for the purpose of sparsity. These kinds of problems are known as a non-deterministic polynomial-time hard (NP-hard) problem since a mathematical representation of ℓ_0 norm does not exist. It is not easy to solve an NP-hard problem, so in most cases, the ℓ_0 norm is substituted by an ℓ_1 or an ℓ_2 norms, which is called an ℓ_0

norm minimization relaxation. This is mentioned in detail in the section below on The Concept of Sparsity.

Definition 1.2: ℓ_1 Norm (Manhattan Norm)

The ℓ_1 norm is the summation of the absolute value of all elements in the vector $x \in R^n$.

$$\|x\|_1 = \sum_{k=1}^{n} |x_k|$$

Definition 1.3: ℓ_2 Norm (Euclidean Norm)

The ℓ_2 norm, also known as the Euclidean norm, is the length of a vector $x \in R^n$.

$$\|x\|_2 = \sqrt{\sum_{k=1}^{n} |x_k|^2} = \sqrt{x^T x}$$

Definition 2: Toeplitz Matrix

An $n \times n$ toeplitz matrix X is a matrix which has constant values descending diagonally from left to right. It is constructed as follows:

$$X = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{11} & a_{12} & \ddots & a_{1(n-1)} \\ a_{31} & a_{21} & a_{11} & \ddots & a_{1(n-2)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{(n-1)1} & a_{(n-2)1} & \ddots & \ddots & a_{12} \\ a_{n1} & a_{(n-1)1} & \cdots & \cdots & a_{11} \end{bmatrix}$$

Definition 3: Convex Optimization

Definition 3.1: Convexity

A function $f : R^n \to R^n$ is convex on an interval $[a, b]$ if the line segment between $(x_1, f(x_1))$ and $(x_2, f(x_2))$ lies above the graph of f, where any point $x_1, x_2 \in [a, b]$ [28].

Convexity is mathematically defined as

$$f(\zeta x_1 + (1-\zeta)x_2) \leq \zeta f(x_1) + (1-\zeta) f(x_2) \tag{2.1}$$

where $0 \leq \zeta \leq 1$. The function f is strictly convex if the inequality (2.1) is strict for all x_1 and x_2 whenever $x_1 \neq x_2$ and $0 < \zeta < 1$.

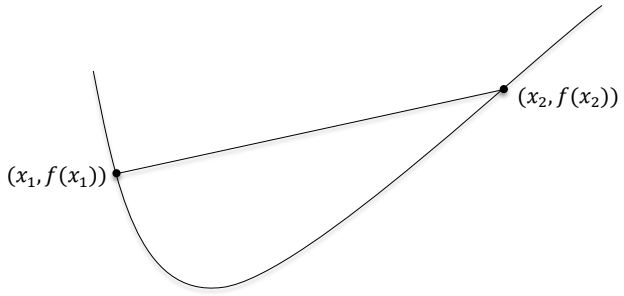

Figure 2.1. Graph of a convex function. The line segment between any two points $x_1, x_2 \in [a, b]$ lies above the graph of the function.

Definition 3.2: Convex Optimization Problem

An optimization problem in the form

$$\begin{aligned} \hat{x} \quad = \quad & \min_x \ f(x) \\ & \text{subject to} \\ & g_i(x) \leq 0 \quad i = 1, 2, \cdots, m \\ & h_j(x) = 0 \quad j = 1, 2, \cdots, n \end{aligned}$$

is called a convex optimization problem if the following conditions are satisfied: the objective function $f(x)$ is convex, the inequality constraints $g(x)$ are convex, and the equality constraints $h(x)$ are affine [28].

Definition 5: White Noise

A stochastic process $\nu(k)$ is called a white noise process if the autocovariance

$$C_{XX}(k_1, k_2) = E[(x(k_1) - \mu_{k_1})(x(k_2) - \mu_{k_2})] = 0$$

where $k_1 \neq k_2$ [29]. In other words, a signal process is white noise if the random variables $\nu(k_1)$ and $\nu(k_2)$ are uncorrelated and statistically independent, which implies that the joint probability density function (PDF) equals the production of an individual PDF of all stochastic processes. A white noise process that contaminates a system is defined by its probabilistic characteristics, which are the mean and the variance. The mean of a white noise process is zero in general provided that a specific statement is not indicated. In this thesis, the disturbance signal is assumed to be zero mean Gaussian white noise, which means each sample in the error signal has a normal distribution with zero mean.

2.2 The Concept of Sparsity

In signal theory, recent studies have focused on identifying a signal vector for a system in such a way that sparse approximation becomes crucial for the interpretation of the concerned system [30]. The assumption is made that the collection of signals is linearly dependent. According to that assumption, it is possible to reveal the linear relationship in the signal sequences. At that point, the concern is to seek an effective approximation such that very few connections between signals represent the whole signal, which sparse approximation provides. The concept is detailed as follows.

Consider a system which is mathematically modeled by the linear equation $Y = X\beta$ where $Y \in R^m$ and $X \in R^{mxn}$ contain the data, and $\beta \in R^n$ is the unknown parameter vector to be estimated. The vector β identifies the correlation between signals in the data. Since the system model is linear, there are infinitely many solutions in general if $m < n$, which means the number of unknowns is larger than the number of equations. Consequently, there is no unique solution to that problem. Thus, the concern is which one is the correct solution within the infinitely many solutions. In many systems, the solution to the objective $\beta \in R^n$ is expected to be sparse, which implies that instead of a full column β vector, a smaller number of unknown parameters identifies the system. In other words, if the insignificant coefficients are held, the system is not identified properly. This causes a complex system that does not define the real system, as well as hard

interpretation, unnecessary data storage and other problems. Therefore, sparsification becomes crucial for solving of these kinds of problems. The definition for sparsity can be given as follows.

Sparsity is the minimum number of non-zero elements in the $\beta \in R^n$ or similarly, maximum number of zero elements. It is measured by an ℓ_0 norm, for example, suppose we have two vectors β_1 and β_2. If $\|\beta_1\|_0 < \|\beta_2\|_0$, then the vector β_1 is considered sparser than β_2. However, in practice, the ℓ_0 solution is not applicable. To explain why, consider the standard sparse decomposition:

$$\begin{aligned} \hat{\beta} &= \min_{\beta \in R^n} \|\beta\|_0 \\ &\text{subject to} \\ &Y = X\beta \end{aligned} \tag{2.2}$$

The solution to this optimization problem recovers the sparsest β vector. The concern is that the optimization problem (2.2) is an NP-hard problem, for which it is almost impossible to find a solution since it is a non-convex optimization problem. Moreover, there is no mathematical description for $\|\beta\|_0$, as it is mentioned in the definition of the ℓ_0 norm.

However, mathematically, the objective function in the optimization problem (2.2) can be replaced by its convex envelope $\min_{\beta \in R^n} \|\beta\|_1$ [2]. This alternative approach is a convex relaxation of the ℓ_0 norm minimization or a relaxation of sparsity. The relaxation of an ℓ_0 norm minimization can be solved with high efficiency, yielding a sparse solution since it is convex [28].

In conclusion, by relaxation of ℓ_0 norm minimization, sparsification is encouraged. Many methods have been developed so far to obtain a sparse solution. Leading methods are Matching Pursuit [31], the LASSO, Basis Pursuit [32], Orthogonal Matching Pursuit [33], and Dantzig Selector [34]. These algorithms have been applied in very wide range in statistic and engineering sciences for the aim of system identification.

2.3 Review of Variable Selection Methods

In statistics, the linear regression is a method for an equation in the matrix form

$$Y = X\beta + \epsilon \tag{2.3}$$

to estimate an unknown regressor vector β. Y is the linear combination of X and β, and ϵ is the prediction error or bias. There are many different techniques to fit the linear model (2.3) to a data set. However, least square approaches are the foremost method in the literature. Least squares require some necessary assumptions to produce parameter estimation, see [35]. However, two of these assumptions can be mentioned here as a reminder. The first assumption is that the model must be linear in the parameters. Secondly, the residuals must be normally distributed with a constant variance i.e. $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. In this section, the concept of least squares will be explained as well as the maximum likelihood approach. Moreover, the ridge regression is mentioned in order to compare with the LASSO in terms of sparsification.

The Maximum Likelihood Approach

The maximum likelihood approach is one of the first methods used within parameter estimation techniques to obtain the parameter vector β in the model (2.3). The maximum likelihood estimation provides parameter estimation by minimizing the likelihood function, assuming that the residuals has a Gaussian distribution with a zero mean and variance $\sigma^2 I$. The likelihood function is the product of the probability density function (PDF) for normally distributed n observations, and it is given by:

$$L(\beta, \sigma^2) = \prod_{i}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\frac{1}{2}\left(\frac{y_i - X_i\beta}{\sigma} \right)^2 \right] \tag{2.4}$$

Taking the log of equation (2.4), and taking the first derivative of this log-likelihood function with respect to β, we have:

$$\frac{\partial(lnL(\beta, \sigma^2))}{\partial\beta} = -\frac{1}{\sigma^2}(-X'y + X'X\beta) = 0 \tag{2.5}$$

The maximum likelihood estimator will result with the estimate $\hat{\beta} = (X^T X)^{-1} X^T y$. For detailed information about maximum likelihood, see [36].

Ordinary Least Squares

The ordinary least squares (OLS) is one of the most basic methods of estimating an unknown parameter β in the model (2.3). It minimizes the residual sum of squares (RSS) as:

$$\hat{\beta} = \arg\min_{\beta} \ \|Y - X\beta\|_2^2 = \arg\min_{\beta} \ (Y - X\beta)^T(Y - X\beta) \tag{2.6}$$

The OLS estimation is equivalent to the maximum likelihood estimation if the error is normally distributed. To find the unknown parameter vector β, the derivative of (2.6) is taken with respect to β, and the partial differentiation will lead to an estimated $\hat{\beta} = (X^TX)^{-1}X^TY$. If the order of the system is not known, which causes the prediction accuracy concern, the OLS is not sufficient for parameter estimation, i.e, the insignificant parameter coefficients in $\hat{\beta}$ are not set to zero. This concern is dealt with by shrinkage methods.

Ridge Regression

Ridge regression is a very commonly used shrinkage method. In this method, RSS is penalized by the convex ℓ_2 norm of the parameter vector β, which is a continuous penalty [37]. The ridge regression in optimization problem form is given by

$$\hat{\beta} = \arg\min_{\beta} \ \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2 \tag{2.7}$$

where $0 \leq \lambda$ is the tuning parameter that controls the shrinkage. The larger λ is, the more the coefficients are shrunk. The solution is given by $\hat{\beta} = (X^TX + \theta I)^{-1}X^TY$ where $0 \leq \theta$ is a constant. Using the ℓ_2 norm enables shrinkage and improves prediction accuracy compared to OLS. However, ridge regression does not perform a perfect subset selection since it does not set any parameter in the regression vector β to zero. This behavior is detailed in the LASSO section. To handle the sparsification issue in the ridge regression, some approaches have been developed, for instance, the LASSO, the smoothly clipped absolute deviation (SCAD) [20] and Elastic Net [21]. Each of these methods has some advantages and disadvantages: for details, see [19], [20], and [21].

2.4 LASSO

The least absolute shrinkage and selection operator (LASSO) is a linear shrinkage method for parameter selection, which penalizes the regression coefficient vector β with an ℓ_1 norm [19]. Mathematically, it is defined by:

$$\hat{\beta} = \underset{\beta \in R^p}{\arg\min} \ \|Y - X\beta\|_2^2 + \lambda \sum_{k=1}^{p} |\beta_k| \tag{2.8}$$

The equation (2.8) can be rewritten equivalently by the Lagrangian duality [28]:

$$\begin{aligned} \hat{\beta} \quad = \quad & \underset{\beta \in R^p}{\arg\min} \ \|Y - X\beta\|_2^2 \\ & \text{subject to} \\ & \|\beta\|_1 \leq \theta \end{aligned} \tag{2.9}$$

LASSO is expressed as an ℓ_1 penalized least squares optimization in (2.8) while it is expressed as an ℓ_1 constrained least squares optimization problem in (2.9). $\lambda \geq 0$ is a tuning parameter which controls parameter selection and sets the shrinkage degree. If λ is very large, the result will have all parameters shrunken to zero, which causes a null β vector, because of the nature of the ℓ_1 norm constraint. In other words, the bigger the λ is, the more elements in β are shrunk to zero. The ℓ_1 norm constraint bound θ in equation (2.9) is a penalty parameter which is a one-to-one correspondence of the tuning parameter λ. Equivalently as λ, small coefficient estimations go to zero if the bound θ is sufficiently small. There is a relationship between λ and θ such that as λ goes to 0, θ goes to ∞, and vice-versa. The bound θ in equation (2.9) should be chosen in such a way that β shrinks, and the objective function reaches its minimum. This is called as a sparse solution, which means that only some coefficients are estimated to be non-zero.

LASSO has an advantage of using the ℓ_1 norm penalty. This is the main advantage of the LASSO because it uses neither an ℓ_2 norm nor the best subset selection but a non-convex ℓ_0 norm. We may take a look at the behavior of an ℓ_1 constraint in the optimization problem (2.9) in order to illustrate the superiority of LASSO by comparing it to ridge regression. Suppose $\beta \in R^2$, such that $\beta = [b_1 \ b_2]^T$. When contour lines of the

residual sum of squares touch the diamond shaped constraint $|b_1| + |b_2| \leq \theta$ in a corner, one of these b_i's becomes zero while the objective function in (2.9) reaches its minimum. This situation does not occur in ridge regression due to the constraint $b_1^2 + b_2^2 \leq \theta$. This geometry is illustrated in Figure 2.2. For $\beta \in R^p$ where $p > 2$, the constraint region will have multiple corners which leads more parameters to be zero for the LASSO equation (2.9) [35].

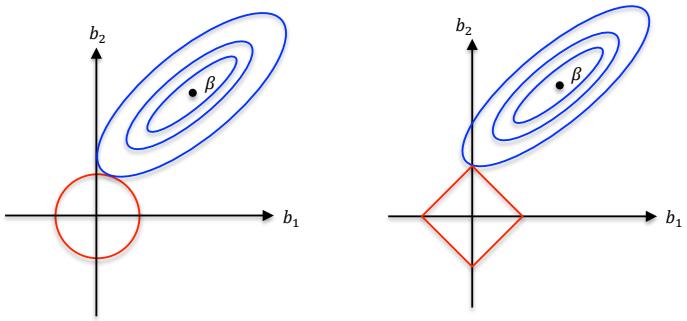

Figure 2.2. Contours of the error and constraint functions for the ridge regression (left) and the LASSO (right). The red lines and inside are constraint regions while blue ellipses are the contours of the least squares error function. The figure is the same as in the Elements of Statistical Learning textbook, second edition.

Due to the advantage of using ℓ_1 norm, LASSO is used in this study for the estimates of LTI system transfer coefficients, which will be detailed further on. Finally, since LASSO is a convex optimization problem, it can be solved efficiently by many algorithms, such as the modified Newton Raphson or the shooting method [38], and the least angle regression (LARS) [39].

Chapter 3

Problem Statement

3.1 Definition of The Problem

In this study, the problem is divided into three parts. The first problem is the correct estimations of the transfer function coefficients of a single-input single-output (SISO) discrete LTI system. The second problem is to estimate true order of the system. The third problem is the estimation of the standard deviation of the measurement noise, which is assumed to be zero mean Gaussian white noise. In light of these problems, it is more convenient to use linear difference equations to represent input-output data. Moreover, due to the measurement noise to be estimated, our difference equations will be in the form of auto-regressive with external input (ARX) model structure. The following section details a formulation of the model structure for noisy p measurements as well as the solution approach to the constructed model.

3.2 Formulation of The Identification Problem

In discrete-time systems, the relationship between model variables is described with difference equations. In principle, the relationship between input and output for an n-th order SISO systems is given as $y(z) = H(z)u(z)$, where $H(z)$ is given by:

$$H(z) = \frac{y(z)}{u(z)} = \frac{(b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n})}{(1 + a_1 z^{-1} + \ldots + a_n z^{-n})} \tag{3.1}$$

$H(z)$ is called the transfer function of the system, which is the z transform of the impulse response sequence. z^{-1} in the transfer function (3.1) is the unit time delay. Suppose we

have a SISO discrete LTI system transfer function as

$$H(z) = \frac{y(z)}{u(z)} = \frac{2 + z^{-1}}{1 + 0.4z^{-1} + 0.7z^{-2}}$$

which indicates in the time domain as:

$$y[k] = -0.4y[k-1] - 0.7y[k-2] + 2u[k] + u[k-1]$$

The coefficients a_i's and b_j's in the transfer function equation (3.1) are the system's characteristics. If they are known, the system is known. According to the transfer function equation (3.1), in the time domain, the output signal value at time k for an n-th order SISO discrete LTI system is represented such that $y(k)$ depends on its weighted past values. In the corresponding difference equation form, it is represented by

$$y(k) = -a_1y(k-1) - a_2y(k-2) - \ldots - a_ny(k-n) + b_0u(k) + b_1u(k-1) + \ldots + b_nu(k-n) \quad (3.2)$$

or in vector form

$$y(k) = \begin{bmatrix} y(k-1) & y(k-2) & \cdots & y(k-n) \end{bmatrix} \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_n \end{bmatrix} + \begin{bmatrix} u(k) & u(k-1) & \cdots & u(k-n) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Our assumption is that the measured output signals are contaminated with zero mean Gaussian white noise. Therefore, the input-output data which is represented in (3.2) is converted into n-th order ARX model with stochastic noise disturbance. The model is mathematically given by:

$$y(k) = -a_1y(k-1) - \ldots - a_ny(k-n) + b_0u(k) + \ldots + b_nu(k-n) + \epsilon_k \quad (3.3)$$

Consider p measurement as following

$$Y = \begin{bmatrix} X_y & X_u \end{bmatrix} \begin{bmatrix} \beta_y^T \\ \beta_u^T \end{bmatrix} + \begin{bmatrix} \epsilon \end{bmatrix}$$

where

$$Y = \begin{bmatrix} y(k) & y(k+1) & \cdots & y(k+p) \end{bmatrix}^T$$

$$X_y = \begin{bmatrix} y(k-1) & y(k-2) & \cdots & y(k-n) \\ y(k) & y(k-1) & \cdots & y(k-n+1) \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ y(k+p-1) & y(k+p-2) & \cdots & y(k+p-n) \end{bmatrix}$$

$$X_u = \begin{bmatrix} u(k) & u(k-1) & \cdots & u(k-n) \\ u(k+1) & u(k) & \cdots & u(k-n+1) \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ u(k+p) & u(k+p-1) & \cdots & u(k+p-n) \end{bmatrix}$$

$$\beta_y = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \end{bmatrix}, \quad \beta_u = \begin{bmatrix} b_0 & b_1 & \cdots & b_n \end{bmatrix}, \text{ and } \epsilon = \begin{bmatrix} \epsilon_1 & \epsilon_2 & \cdots & \epsilon_p \end{bmatrix}^T$$

Now the system with noisy p measurement can be written as

$$Y = X\beta + \epsilon \tag{3.4}$$

where $Y \in R^{p\times 1}$ is the vector containing outputs. $X = [X_y \ X_u] \in R^{p\times r}$ is the data design matrix containing the inputs and outputs of the dynamical system such that it is the addition of the Toeplitz matrices X_y and X_u. $\beta = [\beta_y \ \beta_u]^T \in R^{r\times 1}$, where $r = 2n+1$, is the parameter vector containing both denominator and numerator coefficients of the transfer function. $\epsilon \sim \mathcal{N}(0, \sigma^2 I) \in R^{p\times 1}$ is the noise signal.

Equation (3.4) is a general representation of the linear structure of the problem in this work, which covers all three problems that are mentioned in the previous section. In order to solve this equation with respect to β and σ, the LASSO used is the modified version provided in [1]. The reason for using the modified LASSO is that the standard LASSO equation (2.8) does not provide the noise's standard deviation estimation. The objective function of the modified LASSO is mathematically given by:

$$L(\beta, \sigma) = \frac{\|y - X\beta\|_2^2}{2p\sigma} + \frac{\sigma}{2} + \lambda\|\beta\|_1 \tag{3.5}$$

where $y \in R^{p\times 1}$, $X \in R^{p\times r}$, and $\beta \in R^{r\times 1}$. This estimator is a convex problem i.e., jointly convex regarding the regressor β and the variance σ [1], so the optimization problem is given as:

$$(\hat{\beta}, \hat{\sigma}) = \underset{\beta,\sigma}{\arg\min}\ L(\beta, \sigma) \tag{3.6}$$

By Lagrangian Duality [28], the optimization problem (3.6) can be rewritten as:

$$\begin{aligned} (\hat{\beta}, \hat{\sigma}) \quad = \quad & \underset{\beta,\sigma}{\arg\min}\ \frac{\|y - X\beta\|_2^2}{2p\sigma} + \frac{\sigma}{2} \\ & \text{subject to} \\ & \|\beta\|_1 \leq \theta \end{aligned} \tag{3.7}$$

Two optimization problems (3.6) and (3.7) are considered equivalent such that, the solution of the equation (3.6) is also the solution of equation (3.7) [28]. Because of computational issues, the Lagrangian dual optimization problem (3.7) can be solved more easily than the original optimization problem (3.6). The optimization problem (3.7) provides an estimation of the constant transfer function coefficients a_i's and b_j's

in equation (3.3) as well as the standard deviation of the measurement noise vector ϵ in (3.4). In addition, constraining the parameter vector β with the bound θ enables the true order estimation, yielding a sparse β. The number of non-zero terms in the denominator parameter vector β_y is equal to the order of the system, which depends on the sparsity of β_y. The choice of the bound θ depends on the aim of the estimations because the standard deviation prediction may demand smaller θ than coefficient estimation and vice-versa. Consequently, the three problems mentioned in the previous section are handled when the optimization problem (3.7) is solved by an appropriate method. However, the optimization problem (3.7) itself may not be able to produce the correct and sparse parameter vector estimate β at the same time while providing a close estimation of the standard deviation. The next chapter explains why the ℓ_1 norm constraint bound θ may not provide the desired shrinkage, which gives the order information, and the closest parameter vector estimation β at the same time. Moreover, this concern is handled by applying weight matrices which are seen in the next chapter.

Chapter 4

Identification Algorithm

4.1 Proposed Identification Method

First, the standard deviation of the measurement noise is calculated as follows. If initially a randomly generated parameter vector β is provided, the standard deviation σ is obtained by taking the first derivative of the objective function in the optimization problem (3.7) with respect to σ. Since the optimization problem (3.7) is strictly convex in σ [1], there is always a unique $\hat{\sigma}$, which is calculated by

$$\frac{\partial L(\beta,\sigma)}{\partial \sigma} = -\frac{\|y - X\beta\|_2^2}{2p\sigma^2} + \frac{1}{2} = 0 \tag{4.1}$$

and

$$\hat{\sigma} = \sqrt{\frac{\|y - X\beta\|_2^2}{p}} \tag{4.2}$$

Now, since $\hat{\sigma}$ is known, in order to find the parameter vector $\hat{\beta}$, the optimization problem (3.7) is given as:

$$\hat{\beta} = \min_{\beta} \frac{\|y - X\beta\|_2^2}{2p\hat{\sigma}} + \frac{\hat{\sigma}}{2} \tag{4.3}$$

subject to

$$\|\beta\|_1 \leq \theta$$

The estimation of the standard deviation of the measurement noise and transfer function coefficients are obtained by solving the equations (4.2) and (4.3) iteratively. However, using (4.3), it may not be possible to estimate both the true order of the system and the parameter vector accurately. The reason is that larger coefficients are penalized more heavily than smaller coefficients in an ℓ_1 norm [2]. Hence, penalizing β with an ℓ_1 norm as a whole vector may not give us the sparsest and closest solution simultaneously. In order to overcome this problem, we propose to utilize a weight matrix which is detailed in the following section.

4.1.1 Weighted LASSO

The estimation of the true order of the system depends on the number of non-zero terms in the denominator parameter vector β_y. The number of non-zero terms in β_y depends on the amount of shrinkage which can be controlled by only the ℓ_1 norm constraint bound θ in equation (4.3). However, estimating the true order of the system and the correct parameter vector at the same time may not be possible by controlling only θ. Hence, each element in the parameter vector β is penalized individually by diagonally placed elements in the weight matrix

$$W = \begin{bmatrix} w_{11} & 0 & 0 & \cdots & 0 \\ 0 & w_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & w_{ss} \end{bmatrix}$$

where $W \in R^{s\times s}$, s is the arbitrarily given large number, which is the length of the parameter vector $\beta \in R^{s\times 1}$ such that $r < s$, $r = 2n + 1$ and n is the true order of the system. Each element w_{ii} in the weight matrix W is multiplied by a corresponding element in the parameter vector β, such that the weighted penalized modified LASSO of the optimization problem (3.6) is given by

$$(\hat{\beta}, \hat{\sigma}) = \arg\min_{\beta,\sigma} \ \frac{\|y - X\beta\|_2^2}{2p\sigma} + \frac{\sigma}{2} + \lambda \sum_{i=1}^{s} w_{ii}|\beta_i| \tag{4.4}$$

or equivalently by Lagrangian Duality [28]:

$$\begin{aligned} \hat{\beta} \ &= \ \min_{\beta} \ \frac{\|y - X\beta\|_2^2}{2p\hat{\sigma}} + \frac{\hat{\sigma}}{2} \\ &\text{subject to} \\ &\|W\beta\|_1 \leq \theta \end{aligned} \tag{4.5}$$

Now, the sparse and accurate parameter vector β estimation is calculated with respect to the weighted modified LASSO Lagrangian dual optimization problem (4.5) and the standard deviation calculation equation (4.2). The estimated order is the number of non-zero terms in the sparse denominator parameter vector β_y. After defining the weight matrix and the weighted modified LASSO, the next step is to choose an appropriate weight type.

The accuracy of the estimates depends on the weight function type as well as the choice of the ℓ_1 norm constraint bound θ in the optimization problem (4.5). In this study, two types of weight matrices are presented. The first weight type is a well-known weighting for ℓ_1 norm minimization problems, which is an iterative reweighting for the LASSO [2]. The second weight type is a fixed weighting containing constant numbers. These weight types are detailed as follows.

The first weight type is the reweighting which is the iterative weight such that the weight matrix W consists of $w_{ii} = 1/(|b_i| + \gamma)$'s, where b_i is an element of the parameter vector β, and γ is a very small number, for example 0.0001. Each diagonal term in

the reweighting matrix W is multiplied by the element-wise of β. Owing to the small γ value, the weight matrix W detects smaller elements in the parameter vector β, and puts larger weights on them so that the optimization problem (4.5) can eliminate these coefficients. In other words, with reweighting, the smaller coefficients are penalized more heavily compared to the larger coefficients at each iteration. A general representation of reweighting matrix is given by:

$$W = \begin{bmatrix} 1/(|b_1| + \gamma) & 0 & 0 & \cdots & 0 \\ 0 & 1/(|b_2| + \gamma) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1/(|b_s| + \gamma) \end{bmatrix}$$

This weight matrix is initialized to a diagonal matrix with 1 on all the diagonal entries. At each iteration, the elements of reweighting matrix are updated based on the values of parameter vector estimate at the previous iteration. The solution process with reweighting can be explained as follows. Iterations start with a randomly generated parameter vector β without providing the true order of the system. Then, the standard deviation is calculated using equation (4.2). After obtaining the standard deviation, the optimization problem (4.5) is solved to obtain a parameter vector β. Due to the reweighting matrix, the solution process is carried out iteratively by updating both the standard deviation σ and terms in the parameter vector β, which is sparsified at each iteration. We may ask the question, "How does reweighting performs the sparsification by eliminating unwanted coefficients, yielding a sparse and accurate $\hat{\beta}$ estimate?" One possible answer would be that at each iteration, the least square and the ℓ_1 norm penalization together in equation (3.7) give us a $\hat{\beta}$ vector which includes both wanted and unwanted coefficients when an appropriate bound θ is given. These unwanted coefficients stem from the measurement noise, and they are very small compared to wanted coefficients. Thus, reweighting can easily detect these small coefficients, and puts larger weights by means of γ.

The iteration process can be detailed as follows: At iteration i, the standard deviation $\hat{\sigma}_i$ is calculated in equation (4.2) with respect to $\hat{\beta}_{i-1}$. After obtaining the standard deviation estimate $\hat{\sigma}_i$, the parameter vector estimate $\hat{\beta}_i$ is obtained by solving the op-

timization problem (4.5). Reweighting matrix is updated by $\hat{\beta}_i$ since the terms in $\hat{\beta}_{i-1}$ are replaced with terms in $\hat{\beta}_i$. Depending on how we determine the reweighting matrix parameter γ and the constraint θ in equation (4.5), the iterations are terminated when the indicated tolerance μ is satisfied. The tolerance μ is based on the most recent estimates of the parameter vectors, such that $|\beta_i - \beta_{i-1}| \leq \mu$. If the tolerance is not satisfied, the iterations continue by calculating $\hat{\sigma}_{i+1}$ in equation (4.2) with respect to $\hat{\beta}_i$ and calculating the new parameter vector $\hat{\beta}_{i+1}$ with the reweighting matrix, which is updated in the previous iteration. Section 4.1.2 shows these steps as an iterative algorithm.

The second weight type is formulated by using a monotonically increasing function. For example, an exponential function in the form of $f(x_i) = e^{x_i}$ can be used, where each x_i is a nonnegative sequentially increasing real number such that $x_1 < x_2 < \cdots < x_{(s-1)/2}$. Since the parameter vector $\beta = [\beta_y \ \beta_u]^T$ includes both denominator and numerator coefficients of the transfer function of a SISO discrete LTI system, two fixed weighting matrices are used. Note that β_y and β_u represent denominator and numerator parameter vectors, respectively. Two different function types can be used for β_y and β_u separately, so it is not necessary to use a single function for both parameter vectors. Moreover, there is no certain rule to pick the function type, so any increasing function can be applicable. In this study, an increasing exponential function is used during simulations, which will be seen in detail in the next chapter. As an illustration in the case of $f(x_i) = e^{x_i}$, the weight matrix for the denominator parameter vector is given as:

$$W_y = \begin{bmatrix} e^{x_1} & 0 & 0 & \cdots & 0 \\ 0 & e^{x_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & e^{x_{(s-1)/2}} \end{bmatrix}$$

Similar to the denominator parameter vector weight matrix $W_y \in R^{(s-1)/2 \times (s-1)/2}$, the numerator parameter vector weight matrix $W_u \in R^{(s+1)/2 \times (s+1)/2}$ is constructed. The overall fixed weighting matrix is written as

$$W = \begin{bmatrix} W_y & 0 \\ 0 & W_u \end{bmatrix}$$

The diagonal elements of the fixed weighting matrices W_y and W_u are set to the elements of the increasing functions whereas in the reweighting technique, the initial reweighting matrix W is set to the identity matrix. As in the reweighting case, each diagonal term in the overall fixed weighting matrix W is multiplied by the element-wise of β. The reason for using an increasing function is that less significant coefficients are penalized more heavily than more significant coefficients. Unlike reweighting, the fixed weighting is used only once, where it comes at the beginning of the iterations. Using it at each iteration would make it hard to find accurate estimates for the order and parameters simultaneously since the actual order information is not given to the algorithm.

This weight can be used together with reweighting. In that case, the initial weight matrix is the fixed weighting matrix instead of the initial reweighting matrix which is the identity matrix. Using these weights together mainly has two contributions which enables better estimation, especially for limited input-output measurement. The first contribution is that using fixed weighting together with reweighting decreases the number of iterations because some of the insignificant coefficients are directly eliminated after the first iteration. In other words, reweighting focuses more on finding the best solution within the remaining coefficients and does not deal with the coefficients which are already exterminated by fixed weighting penalization. Thus, it can be said that if the reweighting is used together with fixed weighting, the solution process speeds up. The second contribution is that the fixed weighting together with reweighting produces a more sparse and accurate parameter vector estimate as demonstrated in the next chapter. As a result of introducing these changes, output estimation error decreases compared to using only the reweighting penalization.

In this study, the iterative identification algorithm that is presented in the following section attempts to find the sparsest and most accurate parameter vector $\hat{\beta}$ estimation. By performing the iteration process, the actual order of the system is obtained, which is the number of non-zero elements in the estimated denominator coefficients vector $\hat{\beta}_y$. However, the iteration process does not contribute to the accuracy of estimation of the standard deviation of the measurement noise. During iterations, the standard deviation estimate does not change very much at each iteration. The reason is that after the first iteration, the optimization problem (4.5) provides a parameter vector estimate $\hat{\beta}$ that

satisfies the equation (3.4) although it may not be the desired vector. Therefore, the calculation of the standard deviation with respect to 4.2 will provide a close estimation at each iteration. As a result, at the end of the iteration process, the transfer function coefficients and the order of the SISO discrete LTI system are found. In other words, the concerned system is identified. In addition, the standard deviation of the measurement noise is also obtained.

4.1.2 Identification Algorithm

Algorithm 1 Reweighted LASSO Algorithm for Sparse System Identification

Give data, input $u \in R^{p\times 1}$ and output $y \in R^{p\times 1}$
Set s to a large number
Set t, θ, γ, and μ
Create the toeplitz matrix X, depending on p and s
Initialize the parameter vector $\beta_0 = \beta_{in}$
Set $W \in R^{s\times s}$ to identity matrix or to the elements of the functions diagonally
for $i = 1$ to t **do**
    Calculate $\hat{\sigma_i}^2 = \dfrac{\|y - X\beta_{i-1}\|_2^2}{p}$
    Solve
    $\hat{\beta}_i = \min\limits_{\beta_i} \dfrac{\|y - X\beta_i\|_2^2}{2p\hat{\sigma}_i} + \dfrac{\hat{\sigma}_i}{2}$
    subject to
    $\|W\beta_i\|_1 \leq \theta$
    if $|\beta_i - \beta_{i-1}| \leq \mu$ **then**
        $\beta \leftarrow \hat{\beta}_i$ and $\sigma \leftarrow \hat{\sigma}_i$, **exit**
    else
        Set $W = diag(1/(|\phi_{currentvalues}| + \gamma))$
    end if
end for

Chapter 5

Experiments and Conclusion

5.1 Synthetic Data Simulations

In this section, performance of the algorithm proposed in Chapter 4 is studied with experimental simulations by using MATLAB [40] and optimization software package CVX [41]. First, we discuss synthetic systems and noisy data generation followed by a discussion of the evaluation criteria we use for the results. Then, four SISO discrete LTI system examples are given in order to detail the effect of no weighting, reweighting, and fixed weighting together with reweighting cases. Evaluations and interpretations of the simulation results are based on the order estimation of the concerned system and percentage errors of the standard deviations and outputs. In addition, estimated and the true transfer functions' coefficients are provided as well as the impulse responses for the four examples. Finally, the three weighting cases are tested by 1000 randomly generated systems including all stable second, third, fourth, and fifth order systems to complete the simulations.

To begin with, synthetic systems and noisy data generations are detailed as follows: BIBO stable SISO discrete time LTI systems are randomly generated by a MATLAB command *drss*, which produces stable systems for all the simulations. Four types of input signals are applied to these systems. These input signals are binary, Gaussian, a sum of sinusoids, and uniformly distributed pseudorandom signals. The signals are generated using MATLAB commands *randn* for Gaussian signal and *idinput* for the others. A generated input signal is applied to the generated system in order to obtain an output signal. The noiseless output of the system is generated by using the *lsim*

command with random initial states. Then, zero mean Gaussian white noise is obtained by the *randn* command. This noise signal is added to the noiseless output sequence in the form of ARX model discussed in Chapter 3. We add 10% noise level as measurement noise for all simulations. When generating the noise, 10% of the maximum absolute value of the noiseless output signal of the concerned system is applied as the standard deviation of the noise, such that $\epsilon \in 0.1 \times max(|Y|) \times \mathcal{N}(0,1)$. Therefore, each system has its own measurement noise depending on the maximum absolute value of the output. The best simulation results are chosen such that minimum output errors are considered within various ℓ_1 norm constraint bound θ values. During simulations, 150 different θ values are applied for each system as $\theta = [1 : 0.1 : 15]$ since each system's best results require different values. In fact, the choice of the constraint bound θ depends on the summations of the absolute values of true transfer function coefficients of the concerned system. However, for a real system, the transfer function coefficients are not available. Hence, many θ values are used in order to observe when the algorithm has the sparse parameter vector and the minimum output percentage error for the concerned system among various θ values. The output estimation error formula is the percentage error (PE) in the sense of ℓ_2 norm which is given as:

$$PE = \frac{\|Y_{actual} - Y_{estimated}\|_2}{\|Y_{actual}\|_2} \times 100$$

The error between the true and estimated standard deviation is calculated by the classical percentage error, which is $((|\sigma - \hat{\sigma}|)/\sigma) \times 100$. Finally, during the simulations, the estimated transfer function coefficients less than 0.00001 are assumed to be zero.

Having defined generations of the synthetic systems and noisy data, the transfer functions of the four randomly generated systems from second to fifth orders are given as follows.

$$H_1(z) = \frac{-0.3095 + 1.323z^{-1} + 0.7079z^{-2}}{1 + 0.9405z^{-1} + 0.2109z^{-2}}$$

$$H_2(z) = \frac{0.4136 - 0.8426z^{-1} - 1.421z^{-2} - 0.4563z^{-3}}{1 + 1.9057z^{-1} + 1.163z^{-2} + 0.2155z^{-3}}$$

$$H_3(z) = \frac{-1.092z^{-1} + 1.169z^{-2} - 0.5859z^{-3} + 0.2228z^{-4}}{1 - 1.011z^{-1} + 0.7838z^{-2} - 0.3656z^{-3} - 0.019z^{-4}}$$

$$H_4(z) = \frac{1.7706z^{-1} + 1.3531z^{-2} + 0.3039z^{-3} - 0.8899z^{-4} - 1.0229z^{-5}}{1 + 1.5704z^{-1} + 0.6547z^{-2} - 0.453z^{-3} - 0.7526z^{-4} - 0.3119z^{-5}}$$

The purpose of using these transfer functions is to observe the effect of no weighting, reweighting, and fixed weighting together with reweighting in detail. Simulation inputs are given in Table 5.1. Standard deviations of the measurement noises for the four systems are $0.2073, 0.1752, 0.1003$, and 0.4730, respectively. During simulations, 100 data points are generated because it is clear that as the number of measurements increases, the effect of noise decreases, and the algorithm provides perfect estimation results. However, we are interested in estimating the parameter vector β and standard deviation σ with low measurements. We observed that second and third order systems give acceptable results with 50 measurements, whereas systems with orders higher than third do not in general. For simplicity's sake, all of the simulations are performed with 100 measurements to show results together.

Data Points	100
Input Signal	Randomly generated binary signal
Measurement Noise	10% zero mean Gaussian white noise
Arbitrarily Given Initial Parameter Order	10

Table 5.1. Simulation Inputs

Three cases which are no weighting, only reweighting, and fixed weighting and reweighting together are presented in the following sections.

Case-1: No Weighting

In this case, the algorithm is applied to the optimization problem (3.5), where there is no weighting used. Tables 5.2 and 5.4 show that the true order and the transfer function coefficients are not estimated correctly in this case. It is seen that the algorithm provides an estimate of the required coefficients. However, irrelevant coefficients are found within, leading incorrect order estimation. Existence of the irrelevant coefficients creates extra poles and zeros, causing wrong pole-zero locations. In Figure 5.1, most of the impulse responses of the actual systems and their estimations do not match on account of these extra poles and zeros whereas some parts are matched due to the acceptable wanted coefficients estimation.

	Constraint Bound Value (θ)	System's Order Estimation	Std. Dev. Estimation Percentage Error	Output Estimation Percentage Error
System-1	5.6	10	21.8760	16.3612
System-2	8.5	10	22.0938	19.2206
System-3	8.4	10	11.0009	13.1951
System-4	11.4	10	7.9172	14.8248

Table 5.2. Simulation Results For No Weighting Case

	System-1	System-2	System-3	System-4
σ	0.2073	0.1752	0.1003	0.4730
$\hat{\sigma}$	0.1620	0.1601	0.0893	0.4356

Table 5.3. Standard Deviations of Given and Estimated Noise Signals For No Weighting Case

In addition to the incorrect order estimation, Table 5.2 indicates that standard deviation and output sequence estimation percentage errors are considerably high compared to the other weight cases, displayed in the Tables 5.5 and 5.8. The output estimation percentage errors are expected to be approximately 10% according to the given noise level. However, there is no output percentage error close to 10% for any of the four systems.

As a result, none of these four systems are identified properly in the no weighting case.

During simulations, it is observed that increasing the number of measurement decreases output sequence errors between actual and estimated systems due to required coefficients in parameter vectors approaching the true ones. However, increasing the number of measurements also does not provide a sparse and an accurate parameter vector estimation. This issue is handled by reweighting and fixed weighting cases simulations in the following sections.

	System-1		System-2		System-3		System-4	
	Actual Values	Estimated Values	Actual Values	Estimated Values	Actual Values	Estimated Values	Actual Values	Estimated Values
Numerator Coefficients	-0.3095 1.3230 0.7079	-0.2864 1.3661 0.5137 -0.0027 0.3756 0 0.1995 0 0 0.2802 0.2206	0.4136 -0.8426 -1.4210 -0.4563	0.3923 -0.8635 -1.4197 -0.6915 -0.0862 -.0985 -0.0226 -0.0058 -0.0583 0.0045 -0.0173	0 -1.0920 1.1690 -0.5859 0.2228	0.0056 -1.0876 1.2650 -0.6207 0.2938 -0.0339 -0.0731 0.3550 -0.5830 0.2484 -0.0720	0 1.7706 1.3531 0.3039 -0.8899 -1.0229	-0.0010 1.7457 1.4753 0.4224 -0.8924 -1.1084 -0.1231 0.2823 0.3794 0.0879 -0.0901
Denominator Coefficients	0.9495 0.2109	0.8373 0.1947 0.2840 0.1430 0.1457 0.0638 0.0597 0.2567 0.2631 0.1047	1.9057 1.1630 0.2155	1.9516 1.3657 0.3836 -0.0355 -0.0393 0 0.0242 0.0290 0.0037 0.0072	-1.0110 0.7838 -0.3656 -0.0190	-1.1025 0.8052 -0.4629 -0.0065 0.0526 -0.3150 0.5362 -0.2939 0.1746 0.0082	1.5704 0.6547 -0.4530 -0.7526 -0.3119	1.5918 0.7249 -0.4285 -0.8577 -0.3762 0.2120 0.3631 0.2060 0 -0.0317

Table 5.4. True and Estimated System's Coefficients For No Weighting Case

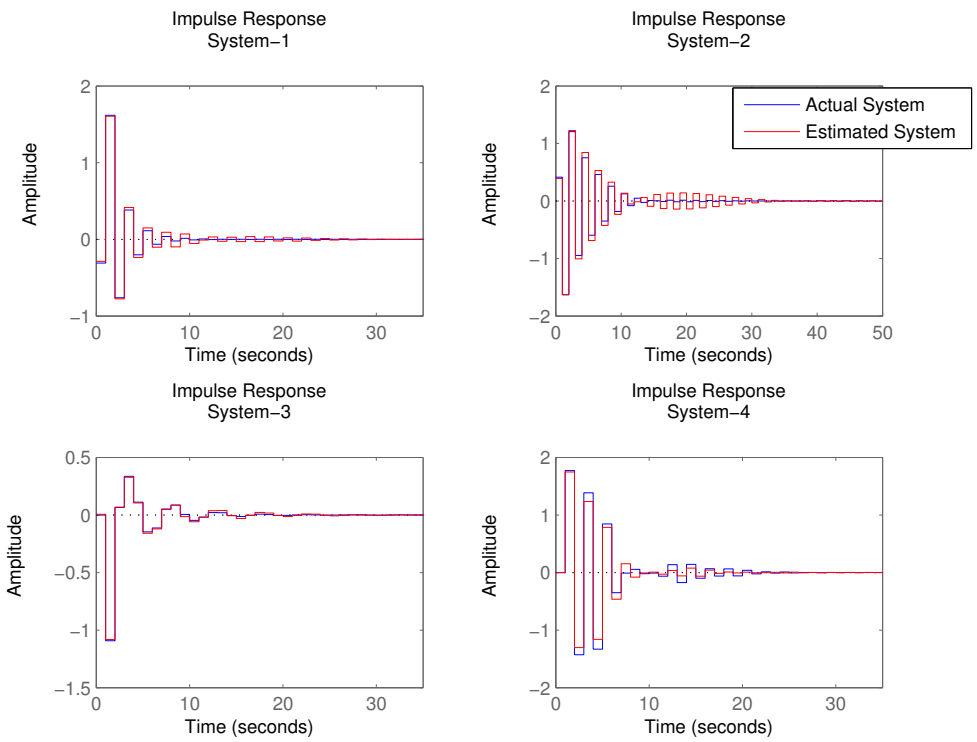

Figure 5.1. Impulse Responses of True and Estimated Systems For No Weighting Case

Case-2: Reweighting

In this case, the algorithm is applied to the optimization problem (4.5), where reweighting is used as the weight matrix. During the simulations, various reweighting parameter (γ) values are tested. The best performance is achieved when $\gamma = 0.0001$ such that $w_{ii} = 1/(|b_i| + 0.0001)$, where each w_{ii} is diagonally placed in $W \in R^{21\times10}$ and b_i is a term in the parameter vector $\beta \in R^{21\times1}$. We observe that values of $\gamma < 0.0001$ and $\gamma > 0.0001$ cause the sparsification problem and increase the errors.

Having said that, the simulation results are interpreted as follows. Tables 5.5 and 5.7 show that the reweighting case can find accurate estimates of the order and transfer function coefficients although a small number of unwanted coefficients are found within the parameter vectors for some systems. If these small unwanted coefficients are ignored, it can be said that reweighting performs estimating both true order and the correct pa-

rameters. Figure 5.2 shows that impulse responses of estimated and true systems are closely matched.

	Constraint Bound Value θ	System's Order Estimation	Std. Dev. Estimation Percentage Error	Output Estimation Percentage Error
System-1	5.6	2	6.4726	9.5851
System-2	8.5	3	2.4849	12.6866
System-3	8.4	7	0.1211	9.7044
System-4	11.4	6	2.8458	11.9980

Table 5.5. Simulation Results For Reweighting Case

	System-1	System-2	System-3	system-4
σ	0.2073	0.1752	0.1003	0.4730
$\hat{\sigma}$	0.1939	0.1708	0.1004	0.4595

Table 5.6. Standard Deviations of Given and Estimated Noise Signals For Reweighting Case

In addition, Table 5.5 indicates that the standard deviation and output sequence estimation percentage errors considerably decrease compared to the no weighting case. The output estimation percentage errors are approximately 10%, as expected.

As a result, reweighting performs sparsification better compared to the no weighting case since it dramatically eliminates most of the unwanted coefficients in the parameter vector estimate $\hat{\beta}$. However, some systems have at least one unwanted coefficient either in the numerator part, denominator part or both. This causes wrong order estimation. This problem can be resolved by lowering the arbitrarily given initial order, but the performances of all weight cases are observed under the same conditions. Since the purpose is to demonstrate the contribution of the fixed weighting method under the same conditions, all the parameters are fixed for all the simulations except weights. The next section shows simulations for the usage of the fixed weight together with reweighting.

	System-1		System-2		System-3		System-4	
	Actual Values	Estimated Values	Actual Values	Estimated Values	Actual Values	Estimated Values	Actual Values	Estimated Values
Numerator Coefficients		-0.2898		0.3828		0		0
		1.3314		-0.8421		-1.0850		1.7587
		0.6802		-1.4143		1.3327		1.3656
		0		-0.5255	0	-0.7709	0	0.2882
	-0.3095	0	0.4136	0	-1.0920	0.3112	1.7706	-0.8453
	1.3230	0.0080	-0.8426	0	1.1690	0	1.3531	-0.8605
	0.7079	0	-1.4210	0	-0.5859	0	0.3039	0
		0	-0.4563	0	0.2228	0	-0.8899	0
		0		0		-0.0150	-1.0229	0
		0		0		0		0
		0		0		0		0
Denominator Coefficients		0.9453		1.9362		-1.1716		1.5304
		0.2076		1.2501		0.9436		0.5977
		0		0.2718		-0.4902		-0.4510
		0		0	-1.0110	0	1.5704	-0.6805
	0.9495	0	1.9057	0	0.7838	0	0.6547	-0.2280
	0.2109	0	1.1630	0	-0.3656	0	-0.4530	0.0410
		0	0.2155	0	-0.0190	0.0129	-0.7526	0
		0		0		0	-0.3119	0
		0		0		0		0
		0		0		0		0

Table 5.7. True and Estimated System's Coefficients For Reweighting Case

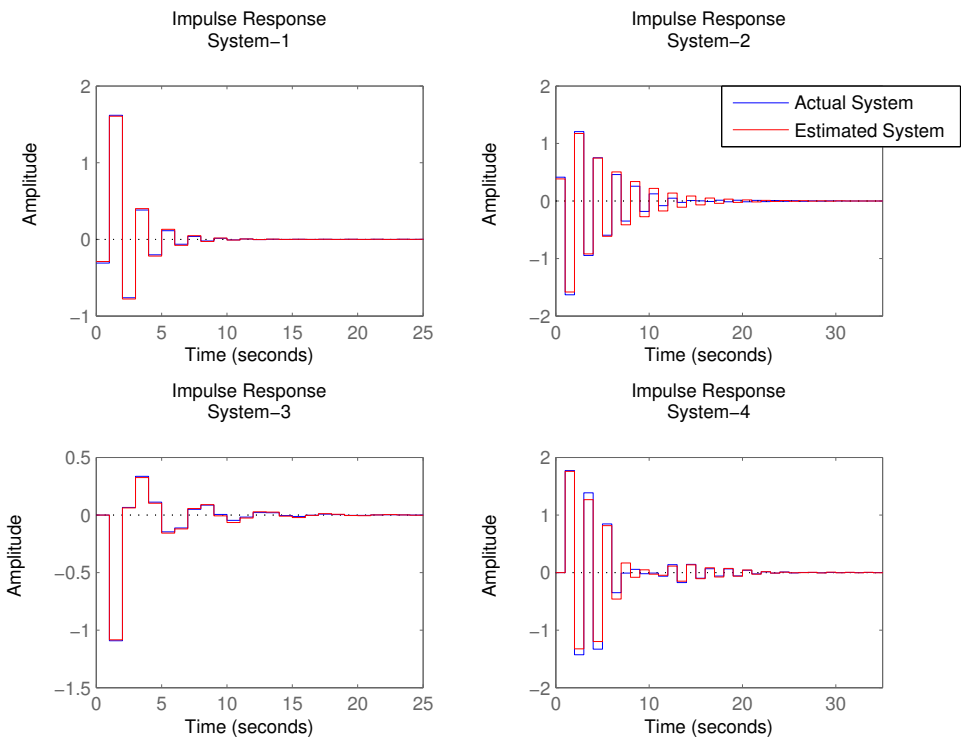

Figure 5.2. Impulse Responses of True and Estimated Systems For Reweighting Case

Case-3: Fixed Weighting and Reweighting

In this case, a fixed weighting and the reweighting are used together. The functions $f_u = e^{(0.1\,:\,0.3\,:\,3.3)}$ and $f_y = e^{(0.1\,:\,0.3\,:\,3)}$ are used as weights. Elements of these functions are diagonally placed in the weight matrices W_u and W_y, where these matrices penalize the numerator and denominator coefficients, respectively. Reweighting parameter (γ) is chosen as in Case-2. Having said that, the simulation results are interpreted as follows.

From Tables 5.8 and 5.10 we see that using this method can lead to accurate estimates of the true order and the transfer function coefficients. The transfer function coefficient estimates in Table 5.10 show that the number of non-zero terms in the denominator parameter vectors, which is equal to the order of the systems, are 2, 3, 4, and 5 as we expect them to be. Figure 5.3 shows that impulse responses of the true and estimated systems are highly matched due to both the correct order and accurate transfer function

coefficient estimations.

	Constraint Bound Value (θ)	System's Order Estimation	Std. Dev. Estimation Percentage Error	Output Estimation Percentage Error
System-1	5.6	2	6.2211	9.5652
System-2	7.5	3	2.3733	11.1720
System-3	8.4	4	0.2754	9.4315
System-4	11.4	5	0.4648	10.5732

Table 5.8. Simulation Results For Fixed Weighting and Reweighting Case

	System-1	System-2	System-3	System-4
σ	0.2073	0.1752	0.1003	0.4730
$\hat{\sigma}$	0.1944	0.1710	0.1006	0.4648

Table 5.9. Standard Deviations of Given and Estimated Noise Signals For Fixed Weighting and Reweighting Case

In addition to the true order and accurate parameter vector estimation, Table 5.8 indicates that standard deviation percentage error decreases compared to the reweighting case. This means a better standard deviation of the measurement noise estimation is performed. Furthermore, output sequence estimation percentage errors are approximately 10% for all systems, which is a desirable result. As a result, all four systems are considered correctly identified using this scheme of weighting.

	System-1		System-2		System-3		System-4	
	Actual Values	Estimated Values	Actual Values	Estimated Values	Actual Values	Estimated Values	Actual Values	Estimated Values
Numerator Coefficients		-0.2906		0.3809		0		0
		1.3280		-0.8341		-1.0854		1.7507
		0.6990		-1.4350		1.2803		1.3213
		0		-0.5353	0	-0.6759	0	0.2330
	-0.3095	0	0.4136	0	-1.0920	0.2784	1.7706	-0.8262
	1.3230	0	-0.8426	0	1.1690	0	1.3531	-0.8426
	0.7079	0	-1.4210	0	-0.5859	0	0.3039	0
		0	-0.4563	0	0.2228	0	-0.8899	0
		0		0		0	-1.0229	0
		0		0		0		0
		0		0		0		0
Denominator Coefficients		0.9584		1.9539		-1.1225		1.5065
		0.2141		1.2689		0.8581		0.5485
		0		0.2695		-0.4498		-0.4565
		0		0	-1.0110	-0.0236	1.5704	-0.6629
	0.9495	0	1.9057	0	0.7838	0	0.6547	-0.2605
	0.2109	0	1.1630	0	-0.3656	0	-0.4530	0
		0	0.2155	0	-0.0190	0	-0.7526	0
		0		0		0	-0.3119	0
		0		0		0		0
		0		0		0		0

Table 5.10. True and Estimated System's Coefficients For Fixed Weighting and Reweighting Case

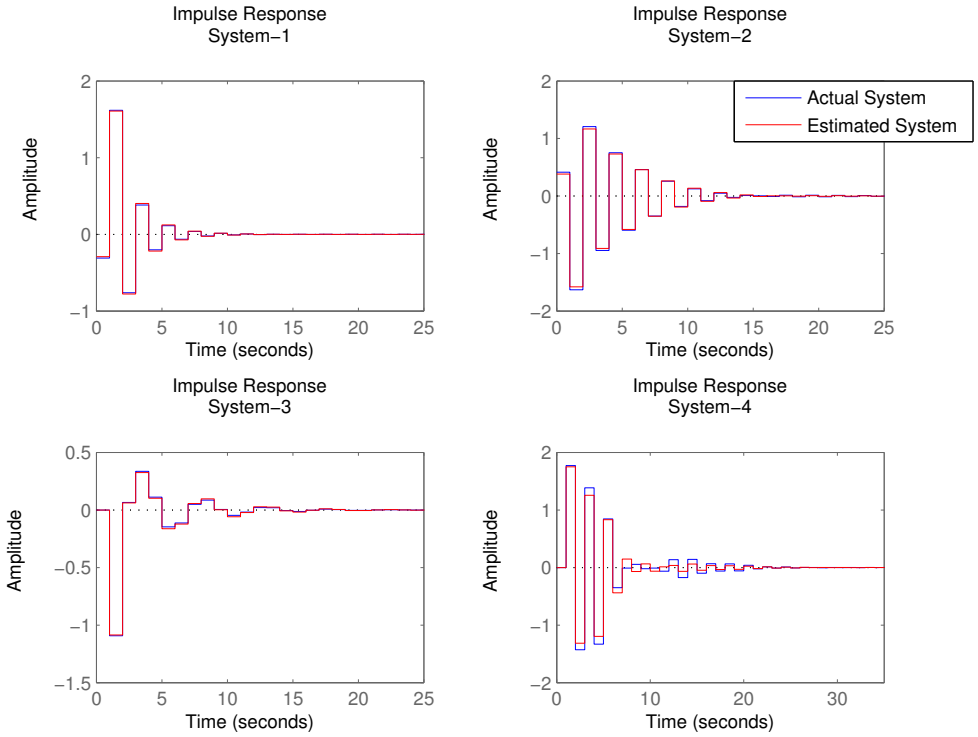

Figure 5.3. Impulse Responses of True and Estimated Systems For Fixed Weighting and Reweighting Case

As mentioned earlier, the accuracy of the algorithm proposed in Chapter 4 is tested by 1000 randomly generated systems including second, third, fourth, and fifth order systems. Simulation inputs are expressed in Table 5.1, except for input signals. Reweighting parameter (γ) is used as in Case-2. Weight functions are selected the same as in Case-3. During simulations, the minimum output sequence percentage errors and the corresponding standard deviation percentage errors are saved for each system. Then the summation of the saved errors for both outputs and standard deviations are divided by 1000 to find the average errors. Tables 5.11 and 5.12 show average errors for various input signals. As shown in the tables, the fixed weighting together with reweighting shows the best performance compared to the other weight cases for all the input signal types.

	Output Percentage Errors		
Input Signals	No Weighting	Reweighting	Fixed Weighting and Reweighting
Binary	22.0994	14.0892	13.6418
Gaussian	26.6957	17.3862	15.9418
Sum of sinusoids	25.0951	20.0635	18.5013
Uniformly distributed pseudorandom	24.4242	19.7243	18.1652

Table 5.11. Average Output Estimation Errors For Various Input Signals

	Standard Deviation Percentage Errors		
Input Signals	No Weighting	Reweighting	Fixed Weighting and Reweighting
Binary	15.6417	2.5212	1.4953
Gaussian	12.0886	1.6800	1.1437
Sum of sinusoids	10.1853	1.7265	1.2070
Uniformly distributed pseudorandom	13.2834	2.3356	1.3807

Table 5.12. Average Standard Deviation Estimation Errors For Various Input Signals

5.2 Conclusion and Discussions

This thesis demonstrated the applicability of the LASSO for identification of SISO discrete time LTI systems from noisy measurements. The identification of the system included estimations of the transfer function coefficients and the order of the system. In addition, the standard deviation estimation of the measurement noise was also discussed. We summarize the entire work as follows.

Chapter 1 and Chapter 2 addressed the background information for the thesis. In Chapter 3, the problem and a suitable mathematical model were presented. Since the standard deviation of the measurement noise was to be estimated, the modified LASSO was used instead of the standard LASSO. In Chapter 4, the algorithm that handled the identification problem was proposed. We did not give the true order information for a system to the algorithm. Thus, the true order of the system was assumed to be unknown. However, this led to the problem of having both sparse and the correct parameter vector estimate. Weight matrices were introduced in order to overcome this issue. Two types of weight matrices were employed: reweighting and fixed weighting matrices. Reweighting has been known for several years as a solution for ℓ_1 norm minimization problems. In this study, however, we introduced the applicability of fixed weighting. Later, the iterative algorithm was constructed in the light of these weight types. In chapter 5, the consistency and the accuracy of the proposed algorithm was tested by experimental simulations. Here we briefly summarize the results obtained in simulations.

First, data satisfying equation (3.4) was constructed by synthetic random systems, which means that every parameter was generated randomly. Therefore, the performance of the proposed algorithm depended on the randomly generated SISO discrete LTI system, the input signal, zero mean Gaussian white noise, arbitrarily given initial order, the choice of weight matrices and the ℓ_1 norm constraint bound θ. Even though the performance of the system depends on the parameters given above, the results mainly depended only on two parameters: weight matrix and the bound θ. During simulations, as explained in section 5.1, a number of θ values were used. It was straightforward to reach the θ value corresponding to the minimum output percentage error among all θ values. However, the choice of the weight matrix was challenging, especially for fixed weighting. Determining a function $f(x_i)$ and its parameters x_i's as the weight was harder than determining only the reweighting parameter (γ).

Secondly, we observed that better estimation results could be obtained by using fixed weighting together with reweighting. The two main contributions of the fixed weighting were observed. One of them was the true order estimation when input-output measurement was limited, which decreased the standard deviation and output percentage errors. The other contribution was we observed an improvement in the running time of the algorithm. When the fixed weighting was used together with reweighting, the solution process sped up.

Third, during the fixed weighting together with reweighting experiments, using small function values increased the estimation accuracy for small amount of data but did not have a great impact on the number of iterations. When it comes to larger amount of data, in contrast, using large function values in the fixed weighting not only showed better results for the estimation accuracy but also sharply decreased the number of iterations such that three or four iterations showed the desired result. In addition to these, using both fixed weighting and reweighting together was not needed for big amount of data since one of these weights could perform estimations of true order, the transfer function parameters, and the standard deviation with high accuracy. The only distinction between these two weightings was the number of iterations for estimations. Compared to reweighting, fixed weighting needs less iterations to achieve estimations.

The fourth and final observation was that there was not a significant change in the standard deviation estimation after the second iteration. The reason was that after the first iteration, the solution of the optimization problem (4.5) provided a parameter vector estimate $\hat{\beta}$. This estimated parameter vector $\hat{\beta}$ satisfies the equation (3.4) even though it might not be the desired sparse parameter vector. Hence, the calculation of the standard deviation provided a close estimation at each iteration. In light of this, it can be said that the estimation of the standard deviation of the measurement noise does not need many iterations.

Additional final concluding remarks regarding the simulation results and discussions about future works can be given as follows. First, the model ARX in Chapter 3 could be easily solved by many algorithms if the true order of the system is known. For example, with the MATLAB arx command, it is very straightforward to solve any ARX model. However, if the true order of the system is unknown, the usage of arx will neither provide the accurate parameter estimation nor the true order information. Moreover, this com-

mand is not able to find the standard deviation of the measurement noise. Henceforth, with an arbitrarily given initial order, the modified LASSO [1] is applied in order to obtain the standard deviation of the disturbance, to have the best parameter estimation, and to reveal the true order information of the system at the same time.

Second, only the exponential function was used as a fixed weighting. However, the choice of the function as a weight is completely empirical. There is no certain rule for choosing the function; therefore, any increasing function can be used. More precise estimation can be performed by trying various functions. By doing so, the identification in terms of errors and computational speed can be improved in future. Moreover, two different function types can also be used for β_u and β_y; hence, it is not necessary to use a single function for both parameter vectors.

Third, instead of using a fixed weighting, initially given randomly generated parameter vector can be adjusted. It can be chosen in a way that the elements in the parameter vector decrease from top to bottom for numerator and denominator coefficients vectors separately. Using this kind of initial parameter vector can enable having a sparse and accurate parameter vector.

Finally, in this study, we restrict ourselves only for SISO systems; thus, applicability of the LASSO for multi-input multi-output (MIMO) systems can be explored in future work. Furthermore, the algorithm proposed in this study was tested by only randomly generated synthetic systems and data. It can be applied for identification of real systems with real measurements in the future.

Bibliography

[1] Sun, T. and C.-H. Zhang (2012) "Scaled sparse linear regression," *Biometrika*, **99**(4), pp. 879–898.

[2] Candes, E. J., M. B. Wakin, and S. P. Boyd (2008) "Enhancing sparsity by reweighted 1 minimization," *Journal of Fourier analysis and applications*, **14**(5-6), pp. 877–905.

[3] Lennart, L. (1999) *System identification: theory for the user*, Prentice Hall PTR, USA.

[4] Bohlin, T. (1991) *Interactive system identification: prospects and pitfalls*, Springer-Verlag New York, Inc.

[5] Burnham, K. P. and D. R. Anderson (2002) *Model selection and multimodel inference: a practical information-theoretic approach*, Springer.

[6] Isermann, R. and M. Mnchhof (2011) *Identification of dynamic systems : an introduction with applications*, Springer, Heidelberg, New York.
URL `http://opac.inria.fr/record=b1132914`

[7] Gevers, M. (2006) "A personal view of the development of system identification: A 30-year journey through an exciting field," *Control Systems, IEEE*, **26**(6), pp. 93–105.

[8] HO, B. and R. E. Kálmán (1966) "Editorial: Effective construction of linear state-variable models from input/output functions," *at-Automatisierungstechnik*, **14**(1-12), pp. 545–548.

[9] Kung, S.-Y. (1978) "A new identification and model reduction algorithm via singular value decomposition," in *Proc. 12th Asilomar Conf. Circuits, Syst. Comput., Pacific Grove, CA*, pp. 705–714.

[10] Ljung, L. (1987) "System identification: theory for the user," *Preniice Hall Inf and System Sciencess Series, New Jersey*, **7632**.

[11] Soderstorm, T. and P. Stoica (1989) "System identification," *UK: Prentice-Hall*.

[12] Van Overschee, P. and B. De Moor (1994) "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, **30**(1), pp. 75–93.

[13] Katayama, T. (2006) *Subspace methods for system identification*, Springer.

[14] Haverkamp, L. (2001) *State Space Identification: Theory and Practice*. URL `http://books.google.com/books?id=zvLhGwAACAAJ`

[15] Qin, S. J. (2006) "An overview of subspace identification," *Computers & Chemical Engineering*, **30**(10-12), pp. 1502–1513.

[16] Åström, K.-J. and T. Bohlin (1966) "Numerical identification of linear dynamic systems from normal operating records," in *Theory of self-adaptive control systems*, Springer, pp. 96–111.

[17] Akaike, H. (1974) "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, **19**(6), pp. 716–723.

[18] Schwarz, G. et al. (1978) "Estimating the dimension of a model," *The annals of statistics*, **6**(2), pp. 461–464.

[19] Tibshirani, R. (1996) "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288.

[20] Fan, J. and R. Li (2001) "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, **96**(456), pp. 1348–1360.

[21] Zou, H. and T. Hastie (2005) "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**(2), pp. 301–320.

[22] Städler, N., P. Bühlmann, and S. Van De Geer (2010) " 1-penalization for mixture regression models," *Test*, **19**(2), pp. 209–256.

[23] Sun, T. and C.-H. Zhang (2010) "Comments on: 1-penalization for mixture regression models," *Test*, **19**(2), pp. 270–275.

[24] Antoniadis, A. (2010) "Comments on: 1-penalization for mixture regression models," *Test*, **19**(2), pp. 257–258.

[25] Fan, J., H. Peng, et al. (2004) "Nonconcave penalized likelihood with a diverging number of parameters," *The Annals of Statistics*, **32**(3), pp. 928–961.

[26] Zhang, C.-H. (2010) "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, pp. 894–942.

[27] Tóth, R., B. M. Sanandaji, K. Poolla, and T. L. Vincent (2011) "Compressive system identification in the linear time-invariant framework," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, IEEE, pp. 783–790.

[28] Boyd, S. and L. Vandenberghe (2004) *Convex Optimization*, Cambridge University Press, Cambridge.

[29] Papoulis, A. and S. U. Pillai (2002) *Probability, random variables, and stochastic processes*, Tata McGraw-Hill Education.

[30] Candès, E. J. (2006) "Compressive sampling," in *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pp. 1433–1452.

[31] Mallat, S. G. and Z. Zhang (1993) "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, **41**(12), pp. 3397–3415.

[32] Chen, S. S., D. L. Donoho, and M. A. Saunders (1998) "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, **20**(1), pp. 33–61.

[33] Tropp, J. A. (2004) "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, **50**(10), pp. 2231–2242.

[34] Candes, E. and T. Tao (2007) "The Dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, **35**(6), pp. 2313–2351.

[35] Hastie, T., R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani (2009) *The elements of statistical learning*, vol. 2, Springer.

[36] Duda, R. O., P. E. Hart, and D. G. Stork (2012) *Pattern classification*, John Wiley & Sons.

[37] Hoerl, A. E. and R. W. Kennard (1970) "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, **12**(1), pp. 55–67.

[38] Fu, W. J. (1998) "Penalized regressions: the bridge versus the lasso," *Journal of computational and graphical statistics*, **7**(3), pp. 397–416.

[39] Efron, B., T. Hastie, I. Johnstone, R. Tibshirani, et al. (2004) "Least angle regression," *The Annals of statistics*, **32**(2), pp. 407–499.

[40] MATLAB (2014) *version 8.3 (R2014a)*, The MathWorks Inc., Natick, Massachusetts.

[41] Grant, M. and S. Boyd (2014), "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," `http://cvxr.com/cvx`.