

APPLICATION OF GAUSS - SEIDEL METHOD AND
SINGULAR VALUE DECOMPOSITION
TECHNIQUES TO RECURSIVE LEAST SQUARES
ALGORITHM

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Atila Mulas

September 1991

QA
402
.M35
1991

APPLICATION OF GAUSS-SEIDEL METHOD AND
SINGULAR VALUE DECOMPOSITION
TECHNIQUES TO RECURSIVE LEAST SQUARES
ALGORITHM

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF
MASTER OF SCIENCE

By

Atilla Malaş

Atilla Malaş

September 1991

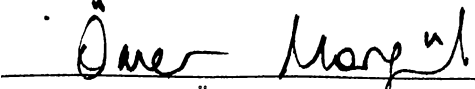
tezis no: 1991-10-10

GA
402
-M35

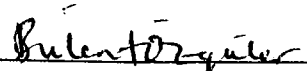
1991

B. 9330

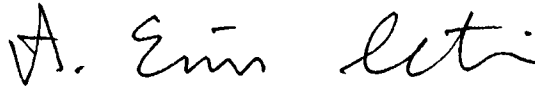
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.


Assist. Prof. Dr. Ömer Morgül (Principal Advisor)


I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.


Assoc. Prof. Dr. Bülent Özgüler

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.


Assoc. Prof. Dr. Enis Çetin

Approved for the Institute of Engineering and Sciences:


Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

APPLICATION OF GAUSS-SEIDEL METHOD AND SINGULAR VALUE DECOMPOSITION TECHNIQUES TO RECURSIVE LEAST SQUARES ALGORITHM

Atilla Malaş

M.S. in Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. Ömer Morgül

September 1991

System identification algorithms are utilized in many practical and theoretical applications such as parameter estimation of systems, adaptive control and signal processing . Least squares algorithm is one of the most popular algorithms in system identification, but it has some drawbacks such as large time consumption and small convergence rates. In this thesis, Gauss-Seidel method is implemented on recursive least squares algorithm and convergence behaviors of the resultant algorithms are analyzed. Also in standard recursive least squares algorithm the excitation of modes are monitored using data matrices and this algorithm is accordingly altered. A parallel scheme is proposed in this analysis for efficient computation of the modes. The simulation results are also presented.

Keywords: System identification, Recursive least squares, Gauss-Seidel iterative method, Singular value decomposition

ÖZET

GAUSS-SEIDEL METODUNUN VE TEKİL DEĞER AYRIŞTIRILMASI TEKNİKLERİNİN ARDIŞIL EN KÜÇÜK KARELER ALGORİTMASINA UYGULANMASI

Atilla Malaş

Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans

Tez Yöneticisi: Yard. Doç. Dr. Ömer Morgül

Eylül 1991

Sistem belirleme algoritmaları, parametre tahmini, uyarlamalı denetim ve sinyal işleme gibi pratik ve teorik çevrelerde yararlı olmaktadır. En küçük kareler algoritması bu algoritmalar içinde en popüler olanıdır , ancak büyük zaman tüketimi ve düşük yakınsama hızı gibi bazı aksaklıkları vardır. Bu tezde Gauss-Seidel yöntemi en küçük kareler algoritmasına uygulanmış ve oluşan algoritmanın yakınsama özellikleri incelenmiştir. Ayrıca standart ardışıl en küçük kareler algoritmasında verilerin uyardığı modlar incelenmiş ve bu algoritma incelenen bu modlara göre değiştirilmiştir. Bu inceleme işleminde kullanılan hesaplamaların paralel gerçekleştirilmesi için bir yöntem önerilmiştir. Deney sonuçları, her bölümün sonunda yer almaktadır.

Anahtar sözcükler: Sistem Belirlenmesi, Ardışıl en küçük kareler, Gauss-Seidel yineleme metodu, Tekil Değer Ayrıştırılması.

ACKNOWLEDGMENT

I am grateful to Assist. Prof. Ömer Morgül, for his supervision, guidance, encouragement and patience during the development of this thesis. I am indebted to the members of the thesis committee: Assoc. Prof. Dr. Bülent Özgüler and Assoc. Prof. Dr. Enis Çetin.

I would like to thank my family for their constant support and all my friends, especially B. Fırat Kılıç, Erkan Savran and Taner Oğuzer for many valuable discussions. Finally, I would like to thank Miss. Ayda Soyay for helping me type the thesis.

Contents

1	Introduction	1
1.1	Introduction	1
2	Mathematical Preliminaries	3
2.1	Recursive Least Squares Algorithm	3
2.1.1	Convergence Properties of RLS	5
2.2	Singular Value Decomposition(SVD)	7
2.3	Stability Theorems	8
2.4	Matrix Perturbations	9
3	Gauss-Seidel Method Applied to RLS	11
3.1	Introduction	11
3.2	Gauss-Seidel Iteration	12
3.3	MIMO Representation for System Identification . .	12
3.4	Application of GS Iteration to RLS Algorithm	15
3.5	Convergence Behaviors of GS Applied Algorithms . .	17
3.6	Simulations	22
4	Investigating RLS using SVD computations	27
4.1	Introduction	27

<i>CONTENTS</i>	vii
4.2 Application of SVD to RLS	28
4.3 A Parallel Implementation of SVD	30
4.4 Asymptotic Behavior of Q_t	35
5 Conclusion	40
5.1 Conclusions	40

List of Figures

3.1	Behavior of $\hat{\theta}_2$ of GS, and standard RLS methods	23
3.2	Block GS results on $\hat{\theta}_3$	24
3.3	Second example results, when GS applied to RLS	25
3.4	Second example results, when Block GS applied	26
4.1	A typical $p(\lambda)$ plot	32
4.2	A Parallel architecture For Algorithm 1 using NR	35

List of Symbols

t	: Discrete time instant.
\mathcal{R}	: The space of real numbers.
\mathcal{N}	: Set of nonnegative integers.
RLS	: Recursive Least Squares.
SVD	: Singular Value Decomposition.
ARMA	: Autoregressive Moving Average.
SISO	: Single Input, Single Output.
MIMO	: Multi Input, Multi Output.
GS	: Gauss-Seidel Method.
NR	: Newton-Raphson Iteration.

Chapter 1

Introduction

1.1 Introduction

System identification is a useful essential step in adaptive control, parameter estimation and signal processing. Recursive algorithms are derived for parameter estimation of linear systems on the basis of the least squares cost function. This well-known cost function could be used for derivation of recursive estimation algorithms. Auto Regressive Moving Average (ARMA) models cover a broad class of systems where system identification algorithms are easily applicable because of linearity. Single input, single output (SISO) and multi input multi output (MIMO) systems could be represented by ARMA models. Recursive identification algorithms are applied by computer systems and data are generally samples of continuous time systems. In Recursive Least Squares (RLS) algorithm a bunch of computation is needed to be done during two samples of a real time process. So, there is a requirement of high speed computation, or fast convergence rates to true parameters of the models for good parameter tracking and for applicability of identifying fast systems. In [3], [10], [9] convergence rates are improved in some sense, when the excitation is complete. In this thesis we propose an alternative to standard RLS algorithm. The proposed algorithm utilizes Gauss-Seidel type sequential update rule. In this rule, the components of the parameter vector are updated sequentially, and at each step of the update, previously updated components of the parameter vector are used. Therefore, the proposed algorithm depends on the selection of the sequence according to which the parameters are updated. If the parameter vector has n components, there are $n!$ different such sequences. We note that the convergence rate of proposed algorithm depends on selection of this sequence. Hence the convergence rate can be improved by changing this

sequence. The Gauss-Seidel iteration method is explained in Chapter 3, for details see [1].

The excitation condition is also an important aspect in RLS convergence behaviors. Persistent excitation will lead to a fast convergence, [4], [8], weak excitation will lead to convergence rate with order $\frac{1}{t}$, [4] and incomplete excitation will lead to bounded errors, [11]. The excitation could be monitored by using the singular value decomposition (SVD) of data matrices. In this thesis, this monitoring is used and the standard RLS algorithm is changed accordingly. To find the SVD of the matrices, we propose a parallel architecture. This architecture could improve the time consumption of standard SVD algorithms. In each block of this architecture an iterative root finding operation is done using the well-known Newton-Raphson method. We also give a sufficient condition to guarantee the convergence in the Newton-Raphson method.

The organization of this thesis is as follows: In Chapter 2 we give the required mathematical preliminaries for the developments done in later chapters. Some well-known stability theorems are given and matrix perturbation analysis techniques introduced. In Chapter 3, Gauss-Seidel method is applied to SISO, and MIMO system identification schemes. Gauss-Seidel method is also applied in block fashion. The convergence properties of these algorithms are also investigated. Some simulation results are given when Gauss-Seidel method is applied to standard RLS in block or sequential fashion. In Chapter 4, SVD is used in RLS for monitoring the excitation modes and an algorithm is derived using modal behaviors. Also a parallelization scheme, which could be used in the algorithms, is presented.

Chapter 2

Mathematical Preliminaries

In this chapter we will define the problems we consider, introduce the relevant notation and give the basic results that we use in the sequel. The systems that we consider are linear, time invariant discrete time systems. These are systems described by difference equations of appropriate order and are given the conventional name **ARMA** ;(AutoRegressive Moving Avarage models)(see below). The notations used in this research is standard; for system identification, see [4], [8], for singular value decomposition, see [14], and for perturbation and stability theorems, see [12]. [15].

2.1 Recursive Least Squares Algorithm

The systems that we consider are of the following form:

$$y(t) = -a_1y(t-1) - a_2y(t-2) \dots - a_r y(t-r) + b_1u(t-1) + b_2u(t-2) \dots + b_l u(t-l) \quad (2.1)$$

where $y(t)$ is the output, $u(t)$ is the input of the system ; $r, l > 0$ are integer constants; $a_i, i = 1, \dots, r$ and $b_j, j = 1, \dots, l$ are constant coefficients. Here, input, output and the coefficients are assumed to be real numbers. This model is known as the ARMA model. If we let:

$$\varphi(t) = [-y(t-1) - y(t-2) \dots - y(t-r) u(t-1) \dots u(t-l)]^T \in \mathcal{R}^n$$

as the regressor vector which is a combination of past inputs and observations and defining

$$\theta = [a_1 : a_2 : \dots : a_r : b_1 : b_2 : \dots : b_l]^T \in \mathcal{R}^n$$

as the parameter vector, the output $y(t)$ can be written as

$$y(t) = \varphi(t)^T \theta, t \geq 1 \quad (2.2)$$

where $n = r + l$. Hence, given $\varphi(0) \in \mathcal{R}^n$ we can derive $y(t)$ at each time t , provided that θ is known.

The identification problem is tentatively defined as follows:

Given system input and output for a time interval, find a parameter vector which minimizes the predefined cost function.

To give a more rigorous problem definition, let us define $\hat{\theta}$ as the estimated parameter vector which has the same dimensions as θ , the real parameter vector. Let $\hat{y}(t) = \varphi(t)^T \hat{\theta}$ be the estimated output at each time $t \geq 1$. Then the problem is to minimize a suitable function of $y(t) - \hat{y}(t)$ with respect to $\hat{\theta}$. Different functions will lead to different estimation algorithms. The most widely used function is error squares cost function which is

$$J(\hat{\theta}, N) = \frac{1}{2} \sum_{t=1}^N e^2(t) \quad (2.3)$$

where $e(t) = y(t) - \hat{y}(t)$, and N is an arbitrary natural number. The optimum $\hat{\theta}$ for minimizing the function $J(\hat{\theta}, N)$ is the solution of the following equation:

$$\Phi_N^T \Phi_N \hat{\theta} = \Phi_N^T Y_N.$$

The matrix Φ_N and the vector Y_N are collections of past data and outputs respectively, (i.e., $\Phi_N = [\varphi(1) : \varphi(2) : \dots : \varphi(N)]^T$, $Y_N = [y(1) : y(2) : \dots : y(N)]^T$), see [8]. Of course to find $\hat{\theta}$ we need pseudoinverse of Φ_N to exist. However by choosing N large we can achieve this task, see [8].

The above solution for $\hat{\theta}$ needs off-line computations since a batch of data are needed (i.e., collect input and past observations till time N). An alternative to this approach is updating the estimated parameters recursively using the new coming data (i.e., $\varphi(t), y(t)$). Given an optimum estimate at time N , the aim is to find an optimum estimate at time $N + 1$. The result is well-known **Recursive**

Least S squares (**RLS**) algorithm, for estimating parameters. The algorithm has two update laws: parameter update and data update (covariance update). These are given as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1)) \quad (2.4)$$

$$K(t) = P(t)\varphi(t) \quad (2.5)$$

$$P^{-1}(t) = P^{-1}(t-1) + \varphi(t)\varphi^T(t) \quad (2.6)$$

(2.4) is the parameter update equation, (2.6) is covariance update equation where $P(t)$ is called covariance and (2.5) calculates the gain. To find $P(t)$ from (2.6), the following matrix inversion identity is used :

$$(A + BCD)^{-1} = A^{-1} - A^{-1}D(C + DAB)^{-1}BA^{-1}$$

provided that the required matrices are invertible. By using the above identity in 2.6, we get the following expression for $P(t)$

$$P(t) = P(t-1) - P(t-1)\varphi(t)(I + \varphi^T(t)P(t-1)\varphi(t))^{-1}\varphi^T(t)P(t-1).$$

To start the algorithm, initial conditions at $t = 0$ are needed. The initial condition $\hat{\theta}(0) \in \mathcal{R}^n$ is chosen to satisfy $\|\hat{\theta}(0)\| \leq M < \infty$ where M is a finite real number, and $P(0)$ is chosen as a symmetric positive definite matrix, usually as $P(0) = kI, k \in \mathcal{R}^+$.

There are also other types of **RLS** algorithms which are classified by the way the covariance matrix is updated. For example, if (2.6) is in the form

$$P^{-1}(t) = \alpha(t)P^{-1}(t-1) + \varphi(t)\varphi^T(t)$$

$$0 < \alpha(t) < 1$$

then this algorithm is called the weighted least squares with exponential resetting, or if the $P(t)$ is set to $k_i I, k_i$ is an arbitrary constant, at some time instants t_1, t_2, \dots , the algorithm is called least squares with covariance resetting. For a detailed exposition of the above algorithms see [4].

2.1.1 Convergence Properties of RLS

Lemma(2.1): Let $y(t)$ satisfy (2.2). Then, the RLS algorithm have the following properties :

$$\text{i) } \|\hat{\theta}(t) - \theta\|^2 \leq \kappa \|\hat{\theta}(0) - \theta\|^2, \forall t \geq 0 \text{ where } \kappa = \text{condition number of } P^{-1} = \frac{\lambda_{\max}(P^{-1}(0))}{\lambda_{\min}(P^{-1}(0))}$$

$$\text{ii) } \lim_{N \rightarrow \infty} \sum_{t=1}^N \frac{e^2(t)}{1 + \varphi^T(t)P(t-1)\varphi(t)} < \infty$$

$$\text{iii) } \lim_{t \rightarrow \infty} \|\hat{\theta}(t) - \hat{\theta}(t-k)\| = 0, \text{ for any finite } k.$$

Proof : See [4]

Lemma(2.1) states that the norm of the parameter error $\tilde{\theta} = \hat{\theta} - \theta$ stays bounded regardless of the input-output behavior, and that the measurement error $e(t)$ is also a summable function. In the following theorem, we give a sufficient condition for $\hat{\theta}(t)$ converge to the real parameters of the system. This could be seen easily if we define a lyapunov like function $V(t)$ as :

$$V(t) = \tilde{\theta}(t)P^{-1}(t)\tilde{\theta}(t)$$

where $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$, and use the properties of Lemma(2.1).

Theorem(2.1): The estimate $\hat{\theta}$ given by equation (2.4), converges to θ (the true parameter vector) if

$$\lim_{t \rightarrow \infty} \lambda_{\min}(P^{-1}(t)) = \infty \quad (2.7)$$

where $\lambda_{\min}(A)$ is the minimum eigenvalue of the matrix A . The condition (2.7) will be called the persistence of excitation condition for RLS algorithm.

Proof : See [4].

Above theorem states that if the regressor vectors make $P^{-1}(t)$ satisfy (2.7) then regardless of the initial conditions, $\hat{\theta}(t)$ asymptotically converges to θ . The rate of convergence highly depends on update of covariance and in standard RLS it is proportional to $(\frac{1}{t})$ if input is weakly persistently exciting, [4].

Here we need to clarify what we mean by persistent excitation .

Definition: A scalar input signal $u(t)_{t \in \mathcal{N}}$ is said to be weakly persistently exciting of order n if

$$\rho_1 I \geq \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \begin{bmatrix} u(t+n) \\ \vdots \\ u(t+1) \end{bmatrix} \begin{bmatrix} u(t+n) & \dots & u(t+1) \end{bmatrix} \geq \rho_2 I \quad (2.8)$$

where $\rho_1 > \rho_2 > 0$.

The above definition can easily be converted to a condition on the regressors [4]

2.2 Singular Value Decomposition(SVD)

SVD is an important tool for analyzing the modes of the covariance matrix for system identification. Curly brackets cover the complex case.

Theorem(2.2): Let $A \in \mathcal{R}^{m \times n} \{ \mathcal{C}^{m \times n} \}$ with rank r . Then there exists orthogonal {unitary} matrices $U \in \mathcal{R}^{m \times m} \{ \mathcal{C}^{m \times m} \}$ and $V \in \mathcal{R}^{n \times n} \{ \mathcal{C}^{n \times n} \}$ such that

$$A = U \Sigma V^T \{ U \Sigma V^* \}$$

$$\Sigma = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}$$

$$S = \text{diag}(\sigma_1 \sigma_2 \dots \sigma_r)$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$$

where $r = \text{rank}(A)$.

Proof: See [14]. Here we sketch the proof.

The elements of S are positive square roots of eigenvalues of $A^T A$, which is positive semidefinite. Since $A^T A$ is symmetric and positive semidefinite, we have $\sigma(A) \subseteq [0, \infty)$. Let us denote the eigenvalues of $A^T A$ by σ_i^2 , $i = 1, \dots, n$. Without loss of generality, we assume that $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0 = \sigma_{r+1} = \dots = \sigma_n$. Let $V_1 = [v_1 \dots v_r]$ and $V_2 = [v_{r+1} \dots v_n]$ be matrices formed by the eigenvectors of positive and zero eigenvalues, respectively. Then $S = \text{diag}(\sigma_1, \dots, \sigma_r)$ and we have

$$A^T A V_1 = V_1 S^2, V_2^T A^T A V_2 = 0$$

Then let us define U_1 as

$$U_1 = A V_1 S^{-1}$$

and let us choose U_2 such that $U = [U_1 U_2]$ is orthogonal. Then finally we will have :

$$U^T A V = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}$$

where $V = [V_1 : V_2]$. The numbers σ_i are called singular values of A and the columns The numbers σ_i are called the singular values of A and the columns of U are called the left singular vectors, the columns of V are called the right

singular vectors of A . If the matrix A is symmetric and positive definite, then the calculation of SVD is the same as the eigenvalue-eigenvector calculation of A . For, in this case $A^T A = A^2$, hence the eigenvalues of A are the same as the singular values of A . The above calculations also show that, in this case $U = V$ and the columns of V are the eigenvectors of A .

Throughout the simulations in this thesis, SVD is used. Direct computation of eigenvalues of $A^T A$ is inefficient due to machine precision, [5]. The most widely used algorithm is a two-phase algorithm proposed by Golub,Reinsch [5]. In this algorithm, the given matrix is reduced to a bidiagonal form by means of Householder transformations, and in the second phase, SVD of this bidiagonal matrix which is the SVD of the original matrix, is computed by using QR algorithm, see [5].

2.3 Stability Theorems

Let us consider the equations

$$y(t+1) = f(t, y(t)) + R(t, y(t)) \quad (2.9)$$

$$y(t+1) = f(t, y(t)) \quad (2.10)$$

where $f : \mathcal{R}_+ \times \mathcal{R}^n \rightarrow \mathcal{R}^n$, is such that $f(t, 0) = 0 \forall t \in \mathcal{N}$, Lipschitz function in an open subset B_a of \mathcal{R}^n (i.e., B_a is a ball of radius a which is on the origin of \mathcal{R}^n), and $R(t, 0) = 0 \forall t \in \mathcal{N}$. We shall consider (2.9) as a perturbation of (2.10). Before examining the stability of (2.9) let us define some notions of stability

Definition (Uniform Asymptotic Stability): The solution $y=0$ of (2.10) with initial condition y_0 is said to be uniformly asymptotically stable if

i) Given $\epsilon > 0$ there exist a $\delta = \delta(\epsilon)$ such that for any y_0 being in the δ neighbourhood of zero (i.e., $B_\delta = \{y_0 \in \mathcal{R}^n \mid \|y_0\| < \delta\}$) the solution $y(t) \in B_\epsilon$.

ii) There is a $\delta > 0$ such that for $y_0 \in B_\delta$ one has $\lim_{t \rightarrow \infty} y(t) = 0$. \square

Definition (Total Stability): The solution $y = 0$ of (2.10) is said to be totally stable (or stable with respect to permanent perturbations), if for every $\epsilon > 0$, there exist two positive numbers $\delta_1 = \delta_1(\epsilon)$ and $\delta_2 = \delta_2(\epsilon)$ such that every solution $y(t)$ with initial condition y_0 of (2.10) lies in B_ϵ for $t \geq t_0$; provided that

$$\|f\| < \delta_1$$

and

$$\|R(t, y(t))\| < \delta_2$$

for $y(t) \in B_\varepsilon, t \geq t_0$ \square .

We can now state the following useful result.

Theorem(2.3): Suppose that the trivial solution of (2.10) is uniformly asymptotically stable and suppose that for some $L_r > 0$ the following holds :

$$\|f(t, y') - f(t, y'')\| \leq L_r \|y' - y''\|$$

where $y', y'' \in B_r \subset B_a$ Then the trivial solution of (2.10) is totally stable.

Proof: See [15]

And finally we have the following result.

Corollary(2.1): Suppose that the hypothesis of Theorem(2.3) is satisfied and that, for $y(t) \in B_a$ one has $\|R(t, y(t))\| < g_t \|y(t)\|$ with $g_t \rightarrow 0$ monotonically. Then, the solution of the perturbed equation, (2.9), is uniformly asymptotically stable.

Proof: See [15].

2.4 Matrix Perturbations

The last tool that we use in this thesis is some matrix perturbation results. A typical problem in this area is to investigate how the eigenvalues and eigenvectors of a linear operator T change when T is subjected to a small perturbation. For a nice treatment of this and related problems see [12]. In dealing with such a problem it is often convenient to consider a family of operators of the form:

$$T(x) = T + xT^{(1)}$$

where x is a small complex number, $T(0) = T$ is called the unperturbed operator and $xT^{(1)}$ is the perturbation.

Let $T, T^{(1)}$ be real symmetric matrices and let x be a small complex number. If φ_i, λ_i are eigenvectors and eigenvalues of T , respectively, then

$$\lambda_i(x) = \lambda_i + \sum_{n=1}^{\infty} x^n \hat{\lambda}^{(n)}$$

and

$$\varphi_i(x) = \varphi_i - xST^{(1)}\varphi_i + x^2S(T^{(1)} - \hat{\lambda}^{(1)})ST^{(1)}\varphi_i - \dots$$

are eigenvalues and eigenvectors of

$$T(x) = T + xT^{(1)}.$$

The coefficients $\hat{\lambda}^{(n)}$ are given by :

$$\hat{\lambda}^{(n)} = \frac{1}{mn} \text{trace}(T^{(1)}P^{(n-1)}) \quad , n \geq 1,$$

where m is the multiplicity of λ and

$$P^{(n)} = -(-1)^n \sum_{\substack{v_1+v_2+\dots+v_n=n \\ k_1+k_2+\dots+k_{n+1}=n, \\ k_j \geq 0, v_j \geq 1}} S^{(k_1)}T^{(v_1)} \dots T^{(v_n)}S^{(k_{n+1})}.$$

Here $S^{(0)} = P_\lambda, S^{(n)} = S^n$, where P_λ is a projection mapping whole space \mathcal{N} to the eigenspace of arbitrary eigenvalue. (i.e. $TPX = \lambda PX$) and S is given by :

$$S = \frac{1}{2\pi i} \int_{\Gamma} \zeta^{-1} R(\zeta) d\zeta$$

$$R(\zeta) = (T - \zeta)^{-1}$$

For a detailed information about the above quantities see [12]. Another useful result is the bound for the norm $\|\varphi_i(x) - \varphi_i\|$ which is:

$$\|\varphi_i(x) - \varphi_i\| \leq |x| \frac{s_0}{(psq)^{\frac{1}{2}}} ((ps)^{\frac{1}{2}} + q^{\frac{1}{2}})^2 \|T^{(1)}\varphi_i\|,$$

where

$$p = \|T^{(1)}P\|, q = \|T(1)S\|, s_\alpha = \|S - \alpha P\|$$

for any α .

The above theorems and corollaries are the basic tools used in the thesis. Occasionally some additional results may be used, in which case they will be explained when they are introduced.

Chapter 3

Gauss-Seidel Method Applied to RLS

3.1 Introduction

We have reviewed the recursive least squares algorithm (RLS) in Chapter 1. This algorithm is known to be robust against the measurements errors [4]. The main drawback is its poor convergence behavior, since as time increases the algorithm turns itself off [3], [4]. Another drawback is the time consumption of one sweep, where sweep means a single iteration. Since in practice the systems in which we will use this algorithm is sampled continuous time systems, the sampling period is limited by the Nyquist rate. An efficient way of applying RLS to identification is to share the work of a the single sweep with p processors (i.e., $p \geq 1$) or to parallelize the algorithm. There exists a large number of efficient parallel implementations of RLS and related algorithms [2], [7]. Especially the work of Jover and Kailath is an important contribution on this area. They implement the most time consuming part of the algorithm, the covariance update, on systolic arrays. The improvement introduced is on the modification of well-known Bierman's LDU (Lower Triangular, Diagonal, Upper Triangular) factorization of a matrix. The covariance update is in fact an LDU update problem which is solved in [6], and the modification in the paper of Jover and Kailath consist of parallelizing the algorithm. A more systematic application schedule of parallel architectures can be found in [2]. In [2], systolic arrays are used as blocks for special purposes which are most commonly used structures in algorithms (i.e., RLS, Kalman Filtering, etc). These blocks perform; back substitution, matrix addition and multiplication, orthogonal decomposition and calculation of the schur complement[2].

In this chapter we will first give a simple way of parameterization of the RLS algorithm for a MIMO (Multi Input, Multi Output) system defined by

ARMA model. Then, we will apply a well known iteration method, which is the Gauss-Seidel iteration, for identification of SISO (single input, single output) (and MIMO) model and examine the convergence behavior of the resultant algorithm.

3.2 Gauss-Seidel Iteration

Assume that we have the following iteration algorithm:

$$x(k+1) = f(x(k)) \quad (3.1)$$

where $x(k), f(x(k)) \in \mathcal{R}^n$. A straightforward way to parallelize the above iterative algorithm is assigning a processor to a component of $x(k)$, and letting them compute $x(k)$ in parallel. For example the i^{th} processor computes

$$x_i(k+1) = f_i(x(k)) \quad (3.2)$$

which is the i^{th} component of $x(k+1)$. This type of iteration is called a Jacobi type of iteration. Another method is the Gauss-Seidel method (from now on abbreviated as GS). It introduces some kind of sequentialism on the calculation of the components of $x(k+1)$. In this algorithm the recent updates of vector's components are used to update the other components of the vector, as follows:

$$\begin{aligned} x_1(k+1) &= f_1((x_1(k), x_2(k), \dots, x_n(k))^T) \\ x_2(k+1) &= f_2((x_1(k+1), x_2(k), \dots, x_n(k))^T) \\ &\vdots \\ x_j(k+1) &= f_j((x_1(k+1), x_2(k+1), \dots, x_j(k), x_n(k))^T) \end{aligned} \quad (3.3)$$

For details, see [1]. As seen from the above equations the sequence chosen in the GS method seems to be user dependent, however, in many algorithms the sequence is predefined and the order is clearly set.

3.3 MIMO Representation for System Identification

In this section, we first give a parameterization of a MIMO system suitable for system identification. We will then apply the RLS algorithm for MIMO systems of representation

We consider the linear, time invariant, MIMO discrete time systems given by the following equation:

$$y(t) + A_1 y(t-1) + \dots + A_r y(t-r) = B_1 u(t-1) + \dots + B_l u(t-l) \quad (3.4)$$

where $t \in \mathcal{N}$, $y(t), u(t) \in \mathcal{R}^m$, $A_i, B_j \in \mathcal{R}^{m \times m}$, $i = 1, \dots, r$, $j = 1, \dots, l$.

Our aim is to find A_i , B_j for $i = 1, \dots, r$, $j = 1, \dots, l$ using observations $y(t)$ and inputs $u(t)$. Let

$$A_i = \begin{bmatrix} a_{11}^i & & a_{1m}^i \\ \vdots & & \vdots \\ a_{m1}^i & \dots & a_{mm}^i \end{bmatrix}, B_j = \begin{bmatrix} b_{11}^j & & b_{1m}^j \\ \vdots & & \vdots \\ b_{m1}^j & \dots & b_{mm}^j \end{bmatrix}, i = 1, \dots, r, j = 1, \dots, l$$

and let θ be defined as

$$\theta = (a_{11}^1 : a_{21}^1 : \dots : a_{12}^1 : a_{22}^1 : \dots : a_{11}^2 : a_{21}^2 : \dots : a_{mm}^l : b_{11}^1 : \dots : b_{mm}^l)^T \in \mathcal{R}^{nm^2} \quad (3.5)$$

(i.e., the columns of A_i , $i = 1, \dots, r$ and then the columns of B_j , $j = 1, \dots, l$, are placed one following other respectively)

In order to generalize the RLS algorithm of SISO systems to MIMO systems, we need to have a linear relation between the output $y(t)$ and the parameter vector θ , see(2.2). For this reason, we choose the regressors $\varphi(t)$ as

$$\varphi(t) = \begin{bmatrix} -y_1(t-1) \\ \\ \\ -y_1(t-1) \\ \\ \\ u_m(t-l) \\ \\ \\ u_m(t-l) \end{bmatrix} \in \mathcal{R}^{m^2 n \times m}. \quad (3.6)$$

It is easy to see that by using (3.5) and (3.6), (3.4) can be written as

$$y(t) = \varphi^T(t)\theta \quad (3.7)$$

which is a suitable representation for identification. We will use the well-known error squares cost function

$$J(\hat{\theta}, N) = \frac{1}{2} \sum_{t=1}^N (y(t) - \varphi^T \hat{\theta})^T (y(t) - \varphi^T \hat{\theta}), \quad (3.8)$$

where $\hat{\theta}$ is the estimate of θ . To get RLS from (3.8), let us determine a $\hat{\theta}$ which minimizes the cost function. Let us define $Y = [y^T(1)y^T(2)\dots y^T(N)]^T$, and $\Phi_N = [\varphi(1)\varphi(2)\dots\varphi(N)]^T$, then the error E and the cost function J becomes

$$E = Y - \Phi_N \hat{\theta},$$

$$J = \frac{1}{2} E^T E.$$

The estimate $\hat{\theta}$ that minimizes J is given by

$$\hat{\theta} = (\Phi_N^T \Phi_N)^{-1} \Phi_N^T Y, \quad (3.9)$$

provided that the matrix $\Phi_N^T \Phi_N$ is invertible. To convert (3.9) to RLS algorithm, we assume that we have found a minimum by using $N - 1$ errors in the cost function J and then we will find minimum when an additional observation comes. Note that

$$\Phi_N^T \Phi_N = \sum_{i=1}^N \varphi(i) \varphi^T(i), \quad \Phi^T Y = \sum_{i=1}^N \varphi(i) y(i) \quad (3.10)$$

If we let $P(k) = (\Phi_k^T \Phi_k)^{-1}$, we have

$$\hat{\theta}(N) = P(N) \left(\sum_{i=1}^N \varphi(i) y(i) \right). \quad (3.11)$$

Since $P^{-1}(k) = \Phi_k^T \Phi_k$ and also having the definitions (3.10), we can obtain the relation

$$P^{-1}(N) = P^{-1}(N-1) + \varphi(N) \varphi^T(N) \quad (3.12)$$

We can rewrite (3.11) as

$$\hat{\theta}(N) = P(N) \left(\sum_{i=1}^{N-1} \varphi(i) y(i) + \varphi(N) y(N) \right), \quad (3.13)$$

and also we have

$$\sum_{i=1}^N \varphi(i) y(i) = P^{-1}(N-1) \hat{\theta}(N-1) = P^{-1}(N) \hat{\theta}(N-1) - \varphi(N) \varphi^T(N) \hat{\theta}(N-1). \quad (3.14)$$

Now, using (3.14) in (3.13) we get the estimate at time N as

$$\hat{\theta}(N) = \hat{\theta}(N-1) + P(N) \varphi(N) (y(N) - \varphi^T(N) \hat{\theta}(N-1)) \quad (3.15)$$

The equations (3.12) and (3.15) yield the RLS algorithm for identification of MIMO system. Here the only difference with the SISO case is that the matrix $P(N)$ to be inverted is of a large size (i.e., $m^2n \times m^2n$). Using the matrix inversion identity in (3.12), we obtain

$$P(N) = P(N-1) - P(N-1)\varphi(N)[I + \varphi^T(N)P(N-1)\varphi(N)]^{-1}\varphi^T(N)P(N-1). \quad (3.16)$$

Here, the tedious computational problem is to determine the inverse of

$$M(t) = [I + \varphi^T(t)P(t-1)\varphi(t)] \in \mathcal{R}^{m \times m}$$

Under some assumptions on the system given by (3.4), a simplified RLS algorithm could be found based on the structure of $M(t)$ (i.e., $M(t)$ could be a diagonal matrix), see [4]. Here we did not impose a condition on $M(t)$. The inversion of $M(t)$ could be done in systolic array type processors which perform orthogonal decompositions and back substitutions. To clarify, let A be the matrix that we want to invert. Then A can be made triangular by multiplication of orthogonal matrices

$$AQ_1Q_2Q_3 \dots Q_n = U$$

where U is upper triangular and n , the number of transformations, depends on A . Then to determine A^{-1} , we invert U by n consecutive back substitutions, and then multiply by Q_i 's, as follows:

$$A^{-1} = Q_1Q_2 \dots Q_nU^{-1}$$

This is of course not the most efficient way of performing the inversion but one which could easily be performed by systolic arrays. On conventional computer systems, there also exist other efficient algorithms, see [13].

This technique of MIMO identification shows the applicability of systolic arrays or parallel processing arrays to identification problems. The convergence behavior of the MIMO-RLS will not be dealt with since this is quite similar to SISO case.

3.4 Application of GS Iteration to RLS Algorithm

The application of GS iteration to RLS algorithm will be given in this section and convergence behavior will be analyzed in the next section. For convenience

let us consider the parameter update equation,

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \varphi^T(t)\hat{\theta}(t-1)).$$

This is an iterative algorithm. We will apply GS method on the elements of $\hat{\theta}$ given by the above algorithm. Here the important point is the selection of the sequence to which we apply GS. In some situations, this sequence changes the convergence behavior. For a clear understanding, we choose the sequence,

$$\theta_1(t) = \theta_1(t-1) + K_1(t)(y(t) - \varphi_1(t)\theta_1(t-1) - \varphi_2(t)\theta_2(t-1) - \dots - \varphi_n(t)\theta_n(t-1))$$

$$\theta_2(t) = \theta_2(t-1) + K_2(t)(y(t) - \varphi_1(t)\theta_1(t) - \varphi_2(t)\theta_2(t-1) - \dots - \varphi_n(t)\theta_n(t-1))$$

$$\theta_n(t) = \theta_n(t-1) + K_n(t)(y(t) - \varphi_1(t)\theta_1(t) - \varphi_2(t)\theta_2(t) - \dots - \varphi_n(t)\theta_n(t-1)), \quad (3.17)$$

for updating the elements of $\hat{\theta}(t)$. The subindex shows that it is the associated component of the vector.

If Jacobi iteration is used in RLS, we can assign a processor to each component of $\hat{\theta}$ and perform the iterations in parallel without having to wait for the result of the other processors. But in (3.17) we can see that each processor has to wait for the results of the previous processors to update its own parameter, hence the parallelism is lost. But in general, in solving linear equations (i.e. $Ax = b$), the GS iterations give better convergence rates than Jacobi type iterations, [1]. Here, we only prove the convergence of RLS with GS iteration. We have been unable to obtain the convergence rates of this algorithm. Hence, a theoretical comparison with RLS with Jacobi type iteration is not made.

The GS approach can also be performed in block fashion. We can partition $\hat{\theta}$, φ , K into k ($k < n$) blocks of the form

$$\hat{\theta} = (\underline{\hat{\theta}}_1 : \underline{\hat{\theta}}_2 : \dots : \underline{\hat{\theta}}_k)^T$$

$$\varphi = (\underline{\varphi}_1 : \underline{\varphi}_2 : \dots : \underline{\varphi}_k)^T$$

$$K = (\underline{K}_1 : \underline{K}_2 : \dots : \underline{K}_k)^T$$

then RLS with GS iteration becomes

$$\hat{\underline{\theta}}_1(t) = \hat{\underline{\theta}}_1(t-1) + \underline{K}_1(t)(y(t) - \underline{\varphi}_1^T(t)\hat{\underline{\theta}}_1(t-1) - \underline{\varphi}_2^T(t)\hat{\underline{\theta}}_2(t-1) - \dots - \underline{\varphi}_k^T(t)\hat{\underline{\theta}}_k(t-1))$$

$$\hat{\underline{\theta}}_2(t) = \hat{\underline{\theta}}_2(t-1) + \underline{K}_2(t)(y(t) - \underline{\varphi}_1^T(t)\hat{\underline{\theta}}_1(t) - \underline{\varphi}_2^T(t)\hat{\underline{\theta}}_2(t-1) - \dots - \underline{\varphi}_k^T(t)\hat{\underline{\theta}}_k(t-1))$$

In (3.19), we need $E^{-1}(t)K(t)$ which is the new gain. Hence, we need the powers of L multiplied by $K(t)$. These can be found as follows, where the index t is dropped for convenience:

$$LK = \begin{bmatrix} 0 \\ K_1 K_2 \varphi_1 \\ K_2 K_3 \varphi_1 + K_2 K_3 \varphi_2 \\ \vdots \\ K_n \sum_{i=1}^{n-1} K_i \varphi_i \end{bmatrix}$$

$$L^2 K = \begin{bmatrix} 0 \\ 0 \\ K_3 K_1 \varphi_1 K_2 \varphi_2 \\ K_4 \sum_{(s_1, s_2)=(1,2,3)} K_{s_1} K_{s_2} \varphi_{s_1} \varphi_{s_2} \\ \vdots \\ K_n \sum_{(s_1, s_2)=(1,2,\dots,n-1)} K_{s_1} K_{s_2} \varphi_{s_1} \varphi_{s_2} \end{bmatrix}$$

$$L^{n-1} K = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ K_n (K_1 \varphi_1 K_2 \varphi_2 \dots K_{n-1} \varphi_{n-1}) \end{bmatrix}$$

The desired gain which is $E(t)^{-1}K(t) = \sum_{i=0}^{n-1} (-L)^i K(t)$ can be expressed, using the above results as

$$\begin{bmatrix} K_1 \\ K_2(1 - K_1 \varphi_1) \\ \vdots \\ K_n (1 + \sum_{j=1}^{n-1} \sum_{(s_1, s_2, \dots, s_j)=(1,2,3,\dots,n-1)} K_{s_1} \dots K_{s_j} \varphi_{s_1} \dots \varphi_{s_j} (-1)^j) \end{bmatrix}$$

which is equal to

$$\begin{bmatrix} K_1 \\ K_2(1 - K_1 \varphi_1) \\ \vdots \\ K_n \prod_{j=1}^{n-1} (1 - K_j \varphi_j) \end{bmatrix} = \begin{bmatrix} 1 & & 0 \\ & 1 - K_1 \varphi_1 & \\ & & \ddots \\ 0 & & \prod_{j=1}^{n-1} (1 - K_j \varphi_j) \end{bmatrix} K(t)$$

Finally if we express E^{-1} as $I - \Gamma_t$ then (3.17) becomes

$$\hat{\theta}(t) = \hat{\theta}(t-1) + (I - \Gamma_t)K(t)(y(t) - \varphi^t(t)\hat{\theta}(t-1)). \quad (3.21)$$

In convergence analysis we need to observe the behavior of the parameter error, which is defined as $\tilde{\theta} = \hat{\theta} - \theta$. Using this definition on standard RLS we have

$$\tilde{\theta}(t) = (I - K(t)\varphi^T(t))\tilde{\theta}(t-1). \quad (3.22)$$

If the excitation condition (2.7) is satisfied, the solution of the above error equation tends asymptotically to zero. In fact, it is known that the convergence is in the order of $(\frac{1}{t})$, i.e., $\|\tilde{\theta}(t)\| \sim O(\frac{1}{t})$ for large t see[4]. To examine the convergence behavior of error when GS is applied, let us write the error equation as

$$\tilde{\theta}(t) = (I - (I - \Gamma_t)K(t)\varphi^T(t))\tilde{\theta}(t-1).$$

Rewriting, we have the equation

$$\tilde{\theta}(t) = (I - K(t)\varphi^T(t))\tilde{\theta}(t-1) + \Gamma_t K(t)\varphi^T(t)\tilde{\theta}(t-1). \quad (3.23)$$

We note that (3.23) is a perturbed equation of (3.22). We will now check that whether the hypothesis of Theorem(2.3) is satisfied for the equation (3.22). The first thing that we have to verify is the total stability of (3.22). We will make the assumption of boundness, which is the regressor vectors produced by the system and input is bounded for all t , i.e., for some $M > 0$, $\varphi^T(t)\varphi(t) \leq M < \infty$, $\forall t$. If we let $f(t, y) = (I - K(t)\varphi^T(t))y$, then we have

$$\begin{aligned} \|f(t, \theta_1) - f(t, \theta_2)\| &= \|(I - K(t)\varphi^T(t))(\theta_1 - \theta_2)\| \\ &\leq \|(I - K(t)\varphi^T(t))\| \|\theta_1 - \theta_2\| \\ &\leq (1 + \|K(t)\varphi^T(t)\|) \|\theta_1 - \theta_2\| \end{aligned} \quad (3.24)$$

Since $K(t) = P(t)\varphi(t)$, using the induced L_2 norm of matrix $K(t)\varphi^T$, we obtain

$$\begin{aligned} \|K(t)\varphi^T\| &\leq \sqrt{\varphi^T P^2(t)\varphi} \sqrt{\varphi^T \varphi} \\ &\leq \varphi^T \varphi \lambda_{\max}(P(t)) \\ &\leq \varphi^T \varphi \lambda_{\max}(P(0)) \end{aligned} \quad (3.25)$$

where we used the fact that $\lambda_{\min}(P^{-1}(t)) \geq \lambda_{\min}(P^{-1}(t-1))$, and hence $\lambda_{\max}(P(t)) \leq \lambda_{\max}(P(t-1))$, see (2.7). Using the assumption of boundedness of regressors (3.24),(3.25), we obtain

$$\|f(t, \theta_1) - f(t, \theta_2)\| \leq [1 + M\lambda_{\max}(P(0))] \|\theta_1 - \theta_2\|. \quad (3.26)$$

The inequality (3.26) gives us a candidate for L_r as $1 + M\lambda_{\max}(P(0))$ for all $\theta_1, \theta_2 \in \mathcal{R}$. Let the regressors satisfy the persistence of excitation condition given

by(2.7), Then since $\tilde{\theta} = 0$ is a uniformly asymptotically stable equilibrium of (3.22), by (3.26) and Theorem(2.3), it is also totally stable. Now let us investigate the perturbed equation (3.17). We can put a bound on the perturbation as follows:

$$\begin{aligned} \|\Gamma_t K(t) \varphi^T \theta\| &\leq \|\Gamma_t K(t) \varphi^T\| \|\theta\| \\ &= \sqrt{\lambda_{\max}(\varphi K^T(t) \Gamma_t^2 K(t) \varphi^T)} \|\theta\| \\ &\leq \sqrt{K^T(t) \Gamma_t^2 K(t)} \|\varphi(t)\| \|\theta\| \\ &\leq \lambda_{\max}(\Gamma_t) \|K(t)\| \|\varphi(t)\| \|\theta\| \\ &\leq \lambda_{\max}(\Gamma_t) \lambda_{\max}(P(0)) \|\varphi(t)\|^2 \|\theta\|. \end{aligned}$$

Let $g_t = \lambda_{\max}(\Gamma_t) \lambda_{\max}(P(0)) \|\varphi(t)\|^2$. Then, since $\lambda_{\max}(\Gamma_t)$ is either zero or is a multiple of K_i , which asymptotically approaches to zero, and since regressors are bounded by assumption, we have

$$\|\Gamma_t K(t) \varphi^T \theta\| \stackrel{def}{=} \|R(t, \theta)\| < g_t \|\theta\|,$$

$g_t \rightarrow 0$ monotonically. Hence, by using the Corollary(2.1), we conclude that $\tilde{\theta} = 0$ is also an asymptotically stable equilibrium of (3.23). We now summarize the above results.

Theorem(3.1): Consider the system given by (3.23), which results from the application of GS to standard RLS for SISO system. Let the persistence of excitation condition given by (2.7) be satisfied. Furthermore, assume that the regressors are bounded, i.e., there exists $M > 0$, such that $\|\varphi(t)\| \leq M, \forall t \in \mathcal{N}$. Under these conditions, $\tilde{\theta} = 0$ is an asymptotically stable equilibrium of (3.23). \square

Theorem(3.1) can be extended easily when GS is applied in a block fashion.

When the system to be identified is a MIMO system which is given in the previous section of this chapter we can still use the analysis presented above.

First thing to be done is choosing the sequence applied for GS method in the parameter update equation for MIMO case. We choose n blocks with m^2 elements each: For example

$$\theta_i = (a_{11}^i : a_{21}^i : \dots : a_{12}^i : a_{22}^i : \dots : a_{mm}^i)$$

i.e., each block θ_i is set by A_i or B_j . When GS is applied, the RLS update can be done in the way shown below:

$$\theta_1(t) = \theta_1(t-1) + K_1(t)(y(t) - \varphi_1^T(t)\theta_1(t-1) - \varphi_2^T(t)\theta_2(t-1) - \dots - \varphi_n^T(t)\theta_n(t-1))$$

where L_r is the required constant of Theorem(2.3). Hence, according to the Theorem (2.3) $\hat{\theta} = 0$ is a totally stable equilibrium point for MIMO RLS error equation, given by (3.28). Now, the perturbation term in(3.29) satisfies

$$\begin{aligned} \|\mathcal{T}_t K(t) \varphi^T \hat{\theta}\| &\leq \|\mathcal{T}_t\| \|\varphi(t)\|^2 \|P(t)\| \|\hat{\theta}\| \\ &\leq M^2 \lambda_{max}(P(t)) \|\mathcal{T}_t\| \|\hat{\theta}\| \end{aligned} \quad (3.31)$$

It can easily be seen that the right hand side of (3.31) asymptotically approaches to zero, hence, by the Corollary (2.1), we conclude that $\hat{\theta} = 0$ is an uniformly asymptotically stable equilibrium point for (3.29). We summarize this result in the following theorem:

Theorem(3.2): Consider the system given by (3.29), which results from the application of GS to standard RLS for a MIMO system. Let the persistence of excitation condition given by (2.7) be satisfied. Furthermore, assume that the regressors are bounded, i.e., there exists $M > 0$ such that $\|\varphi(t)\|, \forall t \in \mathcal{N}$. Under these conditions, $\tilde{\theta} = 0$ is an asymptotically stable equilibrium of (3.29). \square

3.6 Simulations

In this section we show the results obtained in the simulations done in SUN systems written in C-language. In the first one a linear system is identified the system is defined by the difference equation

$$y(t) = 1.7y(t-1) - 1.01y(t-2) + 0.247y(t-3) + 0.021u(t-1) + 0.3u(t-2) - u(t-3). \quad (3.32)$$

We can define

$$\theta = (-1.7 : 1.01 : -.247 : 0.021 : 0.3 : -1)^T$$

and

$$\varphi(t) = [-y(t-1) : -y(t-2) : -y(t-3) : u(t-1) : u(t-2) : u(t-3)]^T.$$

We take initial estimate as $\hat{\theta}(0) = (.005 : .005 : .005 : .005 : .005 : .005)^T$ and the initial regressor as $\varphi(0) = (.05 : .05 : .05 : .05 : .05 : .05)^T$ and the initial covariance as $P(0) = 2000I_{6 \times 6}$. The input to the system is

$$u(t) = 20 \sin\left(\frac{\pi t}{180}\right) + 20 \sin\left(\frac{60\pi t}{180}\right) + 20 \sin\left(\frac{100\pi t}{180}\right).$$

To identify the system RLS is used and GS method is applied to standard RLS using the sequence in (3.17). The asymptotic behavior of a particular element

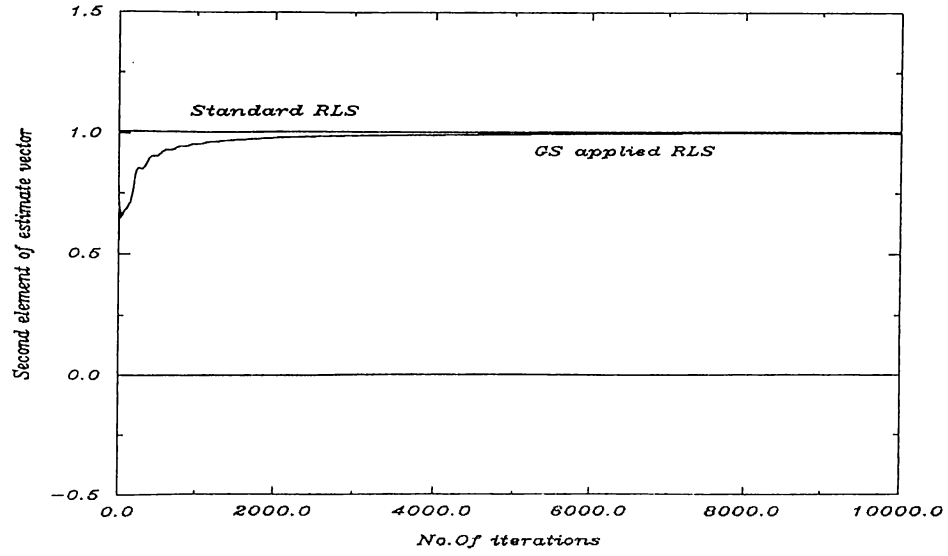


Figure 3.1: Behavior of $\hat{\theta}_2$ of GS, and standard RLS methods

of $\hat{\theta}(t)$, $\hat{\theta}_2(t)$, resulting from standard RLS and GS applied RLS are presented in Figure (3.1). In the second simulation the block GS method is used. For this purpose $\hat{\theta}(t)$ is partitioned to two blocks as

$$\hat{\theta}(t) = [\hat{\theta}_1(t)^T : \hat{\theta}_2(t)^T]^T. \quad (3.33)$$

We, first take

$$\hat{\theta}_1(t) = (\hat{\theta}_1(t) : \hat{\theta}_2(t))^T,$$

and

$$\hat{\theta}_2(t) = (\hat{\theta}_3(t) : \hat{\theta}_4(t) : \hat{\theta}_5(t) : \hat{\theta}_6(t))^T.$$

Then for a second algorithm we take

$$\hat{\theta}_1(t) = (\hat{\theta}_1(t)),$$

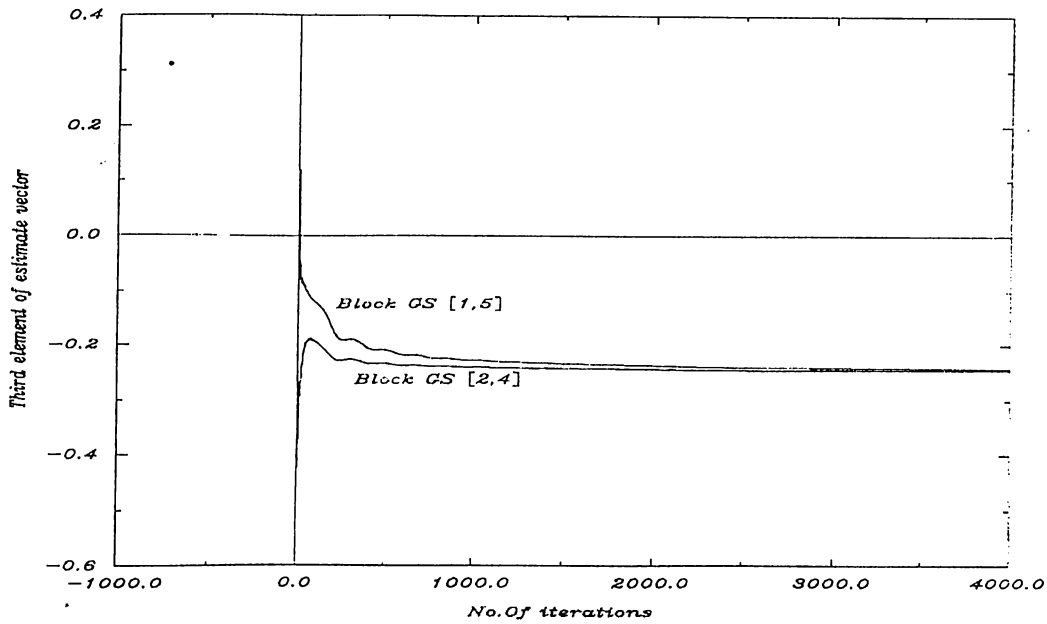
and

$$\hat{\theta}_2(t) = (\hat{\theta}_2(t) : \hat{\theta}_3(t) : \hat{\theta}_4(t) : \hat{\theta}_5(t) : \hat{\theta}_6(t))^T.$$

The resultant behavior of $\hat{\theta}_3(t)$ in both of the two algorithms are given in Figure (3.2).

The second example we consider is a system modelled by the differential equation

$$y(t) = 2.1y(t-1) - 1.8y(t-2) + 0.5y(t-3) + 0.9u(t-1) - u(t-1). \quad (3.34)$$

Figure 3.2: Block GS results on $\hat{\theta}_3$

The regressor and the parameter vectors are defined in the same way as we did in first example. The input is

$$u(t) = 20\sin\left(\frac{6\pi t}{180}\right) + 20\sin\left(\frac{100\pi t}{180}\right).$$

The initial parameter estimate, regressor and covariance, for the iterative algorithms, are,

$$\hat{\theta}(0) = [0.8 : 0.8 : 0.8 : 0.8 : 0.8]^T,$$

$$\varphi(0) = [0.05 : 0.05 : 0.05 : 0.05 : 0.05]^T,$$

$$P(0) = 1000I.$$

Using the same sequencing, in the application of GS, as above we get behavior of $\hat{\theta}_2(t)$ in Figure (3.3).

Then we apply GS in block fashion. $\hat{\theta}(t)$ is partitioned into two blocks like in (3.33), and this is done in two different ways. In the first one

$$\underline{\hat{\theta}_1(t)} = (\hat{\theta}_1(t) : \hat{\theta}_2(t) : \hat{\theta}_3(t))^T,$$

and

$$\underline{\hat{\theta}_2(t)} = (\hat{\theta}_4(t) : \hat{\theta}_5(t))^T.$$

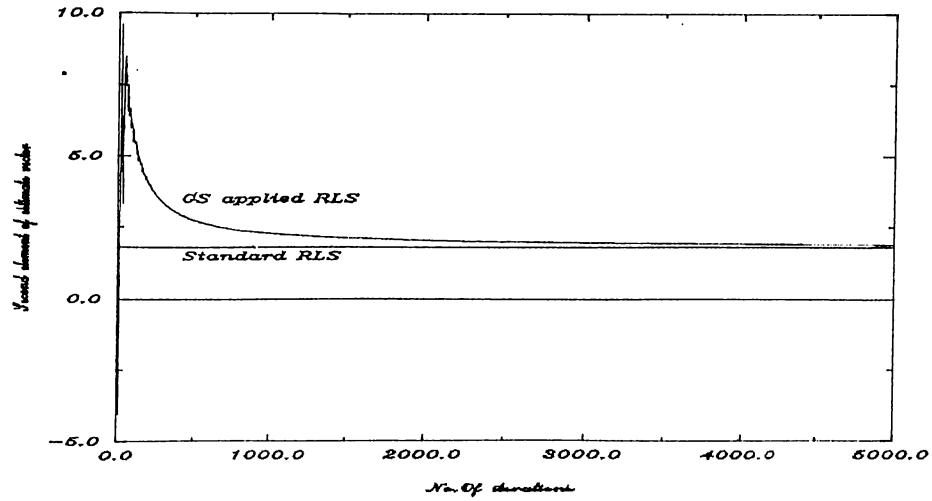


Figure 3.3: Second example results, when GS applied to RLS

In the second one

$$\hat{\theta}_1(t) = (\hat{\theta}_1(t))^T,$$

and

$$\hat{\theta}_2(t) = (\hat{\theta}_2(t) : \hat{\theta}_4(t) : \hat{\theta}_5(t))^T.$$

We display the behavior of $\hat{\theta}_3(t)$ in Figure (3.4). The difference between these two examples is that at the second example the poles of the system is lying on the unit circle. This shows that the system may generate oscillations. It could be seen that the convergence is poor in second example when GS applies to RLS, Figure(3.3). This could happen because of this oscillatory behavior.

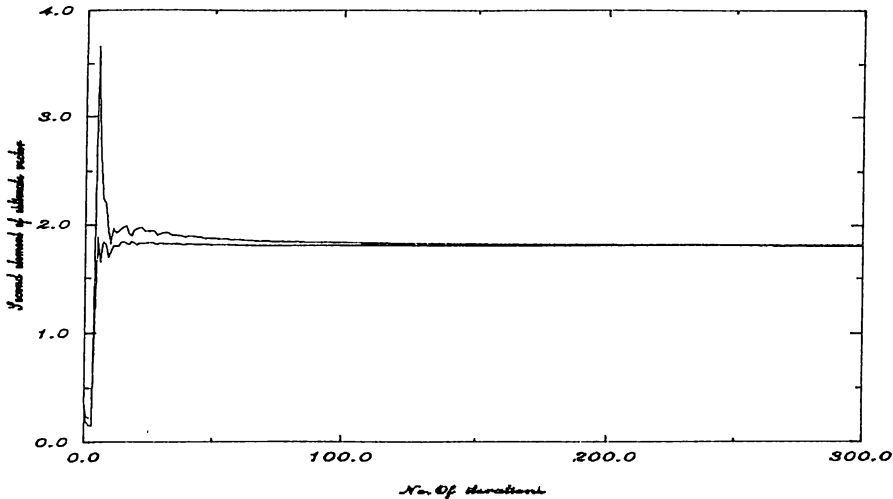


Figure 3.4: Second example results, when Block GS applied

Chapter 4

Investigating RLS using SVD computations

4.1 Introduction

In this chapter we will investigate the RLS algorithm and parameter error behavior in some detail. Of particular interest is the behavior of the algorithm when the excitation is not complete or very weak so that the convergence of the estimated parameters is very slow. It can be shown that when excitation is not complete, the identification error stays bounded and some part of the error goes to zero, [11]. While this might seem to be a serious problem, that the identification error stays bounded may be sufficient for some model reference adaptive control applications, i.e., the tracking error between the plant and the model may still converge to zero, [11]. In [11] defining an unexcitation subspace by the regressor vectors, results on parameter and tracking errors are derived. Since update of the covariance plays an important role for the convergence of the algorithm, this update equation is also modified. In [3], after making some modifications in this equation, results show that exponentially fast convergence could be obtained, by an efficient usage of data. In [9], [10] some modifications of the RLS algorithm which avoid the turning off of the standard RLS algorithm are proposed. The turning off of the standard algorithm can be explained as follows. Assuming that the excitation condition (2.7) is satisfied, it follows that the following holds;

$$\lim_{t \rightarrow \infty} \lambda_{max}(P(t)) = 0 \quad (4.1)$$

Since the gain $K(t)$ of the RLS algorithm is proportional to $P(t)$, see (2.5), it follows that this gain asymptotically approaches to zero, hence the convergence of the algorithm also slows down.

If we investigate how the modes of $P^{-1}(t)$ increase, we can obtain some intuition on the excitation conditions. In this chapter we propose a modification of the algorithm which is based on the excitation of modes at each iteration step and the corresponding parameters are updated correspondingly. Also, simulations suggest that even if all the parameters in a sweep are not updated, asymptotically the parameter error decreases. Main motivations of proposing the above algorithms are the large time consumptions and low convergence rates of the standard RLS algorithm. (i.e. if unnecessary processing does not take place, time consumption decreases). If each step, instead of updating all parameters, we update the ones that are suitable in some sense at average, time consumption can be decreased. For this purpose SVD is used in covariance update equations. Since $P(t)$ is symmetric, and positive definite, SVD is equivalent to eigenvalue-eigenvector computation. The main motivation for the use of SVD can be explained as follows. Once the singular values of $P(t)$ are found, we can compare these values with the previous ones, and find the directions along which the rate of change of the singular values are relatively small. Therefore we may expect small change in the parameters along these directions and do not update the parameters along them. It should be noted that the standard eigenvalue calculation algorithms are not efficient and machine precision dependent. This point should be taken into account when the above algorithm is implemented. Also the calculation of singular values can be reduced to finding roots of a polynomial, which has some interesting root-interlacing property, see section 4.2. This property could be exploited further and a parallelization of the algorithm can be obtained.

4.2 Application of SVD to RLS

Since we are interested in the modes of $P^{-1}(t)$, which can be taken as the singular values of $P^{-1}(t)$, we apply SVD to obtain them. As $P^{-1}(t)$ is symmetric and positive definite, all singular values are positive (not zero) and decomposition is as follows:

$$P^{-1}(t) = U_t D_t^{-1} U_t^T \quad D_t^{-1} = \begin{bmatrix} d_t^1 & & \\ & \ddots & \\ & & d_t^n \end{bmatrix} \quad (4.2)$$

where $U_t \in \mathcal{R}^{m \times m}$ is an orthogonal matrix and $d_t^1 \geq d_t^2 \geq \dots \geq d_t^n > 0$, $t \in \mathcal{N}$ are the singular values. Then the covariance update equation (2.6) becomes

$$P^{-1}(t) = U_{t-1} D_{t-1}^{-1} U_{t-1}^T + \varphi(t) \varphi^T(t) \quad (4.3)$$

By rearranging (4.3) we obtain

$$P^{-1}(t) = U_{t-1}(D_{t-1}^{-1} + U_{t-1}^T \varphi(t) \varphi^T(t) U_{t-1}) U_{t-1}^T \quad (4.4)$$

if we need to find SVD of $P^{-1}(t)$ having SVD of $P^{-1}(t-1)$ at hand, the above equation suggests that we need to find the SVD of $D_{t-1}^{-1} + U_{t-1}^T \varphi(t) \varphi^T(t) U_{t-1}$ if this matrix is decomposed as

$$D_{t-1}^{-1} + U_{t-1}^T \varphi(t) \varphi^T(t) U_{t-1} = Q_t D_t Q_t^T \quad (4.5)$$

then putting (4.5) in (4.4) we obtain

$$P^{-1}(t) = U_{t-1} Q_t D_t Q_t^T U_{t-1}^T = U_t D_t^{-1} U_t^T \quad (4.6)$$

where we set

$$U_t = U_{t-1} Q_t \quad (4.7)$$

The update form introduced by SVD seems easy since a dyad is added to a diagonal matrix. Now let us look what changes did SVD bring to parameter update equation (2.4). Before going further, let us define

$$\phi_t = U_{t-1}^T \varphi(t), \quad \alpha_t = U_{t-1}^T \hat{\theta}(t), \quad \gamma_t = \frac{1}{1 + \phi_t^T D_{t-1} \phi_t} \quad (4.8)$$

Now the gain could be rewritten as

$$K(t) = \frac{P(t-1) \varphi(t)}{1 + \varphi^T(t) P(t-1) \varphi(t)} = \gamma_t U_{t-1} D_{t-1} \phi_t \quad (4.9)$$

Putting (4.9) in the parameter update equation (2.4) and using the definitions (4.8) in (2.4) we obtain

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + \gamma_t U_{t-1} D_{t-1} \phi_t (y(t) - \phi_t^T U_{t-1}^T \hat{\theta}(t-1)) \\ &= U_{t-1} (\alpha_{t-1} + \gamma_t D_{t-1} \phi_t (y(t) - \phi_t^T \alpha_{t-1})) \end{aligned} \quad (4.10)$$

Then an update on α_t will become as

$$\alpha_t = U_t^T U_{t-1} (\alpha_{t-1} + \gamma_t D_{t-1} \phi_t (y(t) - \phi_t^T \alpha_{t-1})) \quad (4.11)$$

and since $U_t = U_{t-1} Q_t$, α_t becomes:

$$\alpha_t = Q_t^T (\alpha_{t-1} + \gamma_t D_{t-1} \phi_t (y(t) - \phi_t^T \alpha_{t-1})) \quad (4.12)$$

The vector α_t is the representation of $\hat{\theta}(t)$ with respect to the basis obtained by the columns of U_t .

In the sequel, we will show that, under some conditions, $\lim_{t \rightarrow \infty} \|Q_t - I\| = 0$. Now assume that this is the case. Now (4.12) implies that, if a certain

component of ϕ_t is zero, then the corresponding components of α_t and α_{t-1} are the same hence the parameters are not updated along this direction. In the sequel, we will show that if a certain component of ϕ_t is zero, then the corresponding singular value of $P(t)$ does not change. Hence, by monitoring the singular values and by not updating the parameter equations along certain directions along which the singular values do not change very much, one may decrease the time consumption of the RLS algorithm.

To summarize, the use of the SVD in RLS results in the algorithm:

Algorithm1:

1) Given the SVD of $P_{t-1}^{-1} = U_{t-1}D_{t-1}U_{t-1}^T$ and the regressors $\varphi(t)$, find the SVD of

$$D_{t-1}^{-1} + U_{t-1}^T \varphi(t) \varphi^T(t) U_{t-1} \stackrel{\text{def}}{=} Q_t D_t^{-1} Q_t^T \quad (A)$$

2) Update U_t as:

$$U_t = U_{t-1} Q_t \quad (B)$$

hence

$$P_t^{-1} = U_t D_t^{-1} U_t^T$$

3) The parameter update equation is given as:

$$\alpha_t = Q_t^T (\alpha_{t-1} + \gamma_t D_{t-1} \phi_t (y(t) - \phi_t^T \alpha_{t-1}))$$

4) go to step 1.

4.3 A Parallel Implementation of SVD

In this part we will give an update scheme for covariance update (4.5). In view of (A), in implementing SVD in RLS algorithm, the crucial step is to find the SVD of the following matrix:

$$D_t + \phi_{t+1} \phi_{t+1}^T \quad (4.13)$$

where D_t is a diagonal matrix with entries $d_t^1 \geq d_t^2 \geq \dots \geq d_t^n > 0$, and $\phi_{t+1} \in \mathcal{R}^n$ is a vector. Since the matrix in (4.13) symmetric, positive definite, finding the SVD is equivalent to an eigenvalue-eigenvector decomposition.

To find the singular values of (4.13), we first note the following:

$$\begin{aligned} p(\lambda) &\stackrel{\text{def}}{=} \det(\lambda I - D_t - \phi_{t+1}\phi_{t+1}^T) = \det((\lambda I - D_t)(I - (\lambda I - D_t)^{-1}\phi_{t+1}\phi_{t+1}^T)) \\ &= \det(\lambda I - D_t)\det(I - (\lambda I - D_t)^{-1}\phi_{t+1}\phi_{t+1}^T) \end{aligned} \quad (4.14)$$

But, since

$$\det(I - (\lambda I - D_t)^{-1}\phi_{t+1}\phi_{t+1}^T) = \det(I - \phi_{t+1}^T(\lambda I - D_t)^{-1}\phi_{t+1})$$

then (4.14) can be written as

$$p(\lambda) = \left(1 - \frac{(w_{t+1}^1)^2}{(\lambda - d_t^1)} - \frac{(w_{t+1}^2)^2}{(\lambda - d_t^2)} - \dots - \frac{(w_{t+1}^n)^2}{(\lambda - d_t^n)}\right) \prod_{j=1}^n (\lambda - d_t^j)$$

or equivalently

$$p(\lambda) = \prod_{j=1}^n (\lambda - d_t^j) - \sum_{i=1}^n ((w_{t+1}^i)^2) \prod_{j \neq i} (\lambda - d_t^j) \quad (4.15)$$

where $\phi_{t+1} = (w_{t+1}^1, w_{t+1}^2, \dots, w_{t+1}^n)^T$ and $D_t = \text{diag}(d_t^1, d_t^2, \dots, d_t^n)$

Remark1: Now, assume that $w_{t+1}^i = 0$ for some $i = 1, 2, \dots, n$. We see then from (4.15) that $p(d_t^i) = 0$, hence $d_{t+1}^i = d_t^i$, i.e., the i^{th} singular value of D_{t+1} and D_t are the same. \square

Since $d_t^1 \geq d_t^2 \geq \dots \geq d_t^n > 0$, we have

$$p(d_t^1) = -(w_{t+1}^1)^2 (d_t^1 - d_t^2)(d_t^1 - d_t^3) \dots (d_t^1 - d_t^n) \leq 0 \quad (4.16)$$

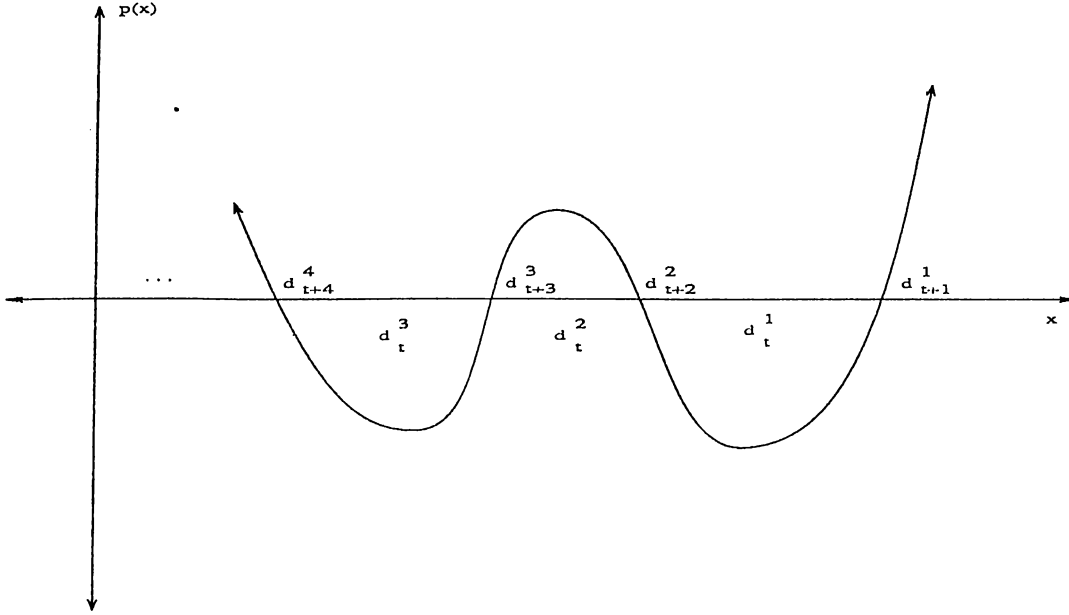
$$p(d_t^2) = -(w_{t+1}^2)^2 (d_t^2 - d_t^1)(d_t^2 - d_t^3) \dots (d_t^2 - d_t^n) \geq 0$$

$$p(d_t^j) = -(w_{t+1}^j)^2 (d_t^j - d_t^1) \dots (d_t^j - d_t^{j-1})(d_t^j - d_t^{j+1}) \dots (d_t^j - d_t^n) \begin{cases} \leq 0 & \text{if } j = 2k + 1 \\ \geq 0 & \text{if } j = 2k \end{cases}, k \in \mathcal{N}$$

Since $p(\lambda)$ is continuous in λ a typical figure for $p(\lambda)$ is as shown in the Figure(4.1). Then, it is not hard to see that an interlacing property between the new roots and the previous ones, as given below, holds:

$$d_{t+1}^1 \geq d_t^1 \geq d_{t+1}^2 \geq d_t^2 \geq \dots \geq d_{t+1}^n \geq d_t^n > 0 \quad (4.17)$$

Furthermore, if $w_t^j \neq 0$ for $j = 1, 2, \dots, n$, and if $d_t^1 > d_t^2 > \dots > d_t^n > 0$, then the inequalities in (4.17) can be replaced by strict inequalities. This suggest that, to find the j^{th} root of $p(\lambda)$, we may choose d_t^j as initial condition and apply the well-known Newton-Raphson (NR) method. A problem with these initial


 Figure 4.1: A typical $p(\lambda)$ plot

conditions is that two of them may be an initial condition of a same root. If we choose d_t^i , $i = 1, \dots, n$ as the initial conditions in NR, to guarantee appropriate convergence behaviors, $p(\lambda)$ should satisfy

$$p(d_t^i) \frac{d}{d\lambda}(p(d_t^i)) < 0, i = 1, 2, \dots, n \quad (4.18)$$

To put condition on the regressors we will write the explicit form of (4.18), using the product rule in differentiation, as

$$p'(\lambda) = \left(1 - \sum_{j=1}^n \frac{(w_{t+1}^j)^2}{(\lambda - d_t^j)}\right) \frac{d}{d\lambda} \left(\prod_{j=1}^n (\lambda - d_t^j) + \prod_{j=1}^n (\lambda - d_t^j) \left(\sum_{j=1}^n \frac{(w_{t+1}^j)^2}{(\lambda - d_t^j)^2}\right)\right) \quad (4.19)$$

If we examine the behavior of $p'(\lambda)$ at the point, d_t^k , we obtain

$$p'(d_t^k) = \left[1 - \sum_{i \neq k}^n \frac{((w_{t+1}^i)^2 - (w_{t+1}^k)^2)}{(d_t^k - d_t^i)}\right] \prod_{j \neq k}^n (d_t^k - d_t^j) \quad k = 1, \dots, n \quad (4.20)$$

if $k = 1$ then since $p(d_t^1) < 0$, we need $p'(d_t^1) > 0$ or, if $k = 2$ then since $p(d_t^2) > 0$, we need $p'(d_t^2) < 0$ for a general result let us partition $p'(d_t^k)$ as:

$$p'(d_t^k) = q(d_t^k) \prod_{j \neq k}^n (d_t^k - d_t^j) \quad (4.21)$$

where

$$q(d_t^k) = \left[1 - \sum_{i \neq k}^n \frac{((w_{t+1}^i)^2 - (w_{t+1}^k)^2)}{(d_t^k - d_t^i)}\right]$$

we can easily see that $\prod_{j \neq 1}^n (d_t^1 - d_t^j) > 0$, and $\prod_{j \neq 2}^n (d_t^2 - d_t^j) < 0$. Then, from (4.15) it can easily be concluded that the second term in the right hand side of (4.20) and $p(d_t^k)$ have opposite signs, for $k = 1, 2, \dots, n$. Hence (4.17) is equivalent to the following.

$$q(d_t^k) > 0 \quad k = 1, 2, \dots, n \quad (4.22)$$

Hence, from the above calculations it can be concluded that if we choose d_t^i , $i = 1, 2, \dots, n$, as the initial conditions, then (4.21) gives a sufficient condition for the proper convergence in Newton-Raphson method. Then we can propose a parallel method for calculating roots of $p(\lambda)$. In this method each processor calculates roots by NR method given the initial conditions which are previously defined. (i.e. d_t^i) We do not only need the eigenvalues but also the eigenvectors to find the matrix Q_{t+1} which contains eigenvectors. (i.e. $Q_{t+1} = [q_{t+1}^1 \ q_{t+1}^2 \ \dots \ q_{t+1}^n]$). Eigenvectors q_{t+1}^i , are given by the following formula

$$q_{t+1}^i = \tau(d_{t+1}^i I - D_t)^{-1} \phi_{t+1} \quad i = 1, 2, \dots, n \quad (4.23)$$

Where, τ is chosen to make $(q_{t+1}^i)^T (q_{t+1}^i) = 1$. To verify that above is an eigenvector of $D_t + \phi_{t+1} \phi_{t+1}^T$ associated with the eigenvalue d_{t+1}^i , let us multiply the matrix with q_{t+1}^i :

$$\begin{aligned} & (D_t + \phi_{t+1} \phi_{t+1}^T) \tau(d_{t+1}^i I - D_t)^{-1} \phi_{t+1} \\ &= \tau D_t (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} + \tau \phi_{t+1} \phi_{t+1}^T (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} \end{aligned} \quad (4.24)$$

Assume that $d_{t+1}^i \neq d_t^j$ $i = 1, 2, \dots, n$ $j = 1, 2, \dots, n$ (note that this is the case if $w_t^i \neq 0$ $i = 1, 2, \dots, n$ and $d_t^1 > d_t^2 > \dots > d_t^n > 0$). Since $p(d_{t+1}^i) = 0$ we will have:

$$1 - \phi_{t+1}^T (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} = 0 \quad (4.25)$$

Using (4.25) in (4.24) we get

$$\begin{aligned} & \tau D_t (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} + \tau \phi_{t+1} \phi_{t+1}^T (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} \\ &= \tau D_t (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} + \tau \phi_{t+1} \\ &= \tau (D_t (d_{t+1}^i I - D_t)^{-1} + I) \phi_{t+1} \\ &= \tau (D_t + (d_{t+1}^i I - D_t)) (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} \\ &= d_{t+1}^i \tau (d_{t+1}^i I - D_t)^{-1} \phi_{t+1} \\ &= d_{t+1}^i q_{t+1}^1 \quad \square \end{aligned}$$

These calculations hold if $d_t^i \neq d_t^j$ $i \neq j$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$. If we set $P(0) = \text{diag}(d_0^1 d_0^2 \dots d_0^n)$ with $d_0^1 > d_0^2 > \dots > d_0^n > 0$ and if $w_t^i \neq 0$ $i = 1, 2, \dots, n, \forall t$, then (4.16) holds with strict inequalities and (4.22) gives all eigenvectors, i.e., the columns of Q_{t+1} . Another set up for calculation of roots of $p(\lambda)$ could be given under some conditions which we will give below. These results will give us the approximate roots of $p(\lambda)$. If we set

$$d_{t+1}^i = d_t^i + \Delta_i \quad (4.26)$$

then for d_{t+1}^i to be an eigenvalue the following must hold:

$$p(d_{t+1}^i) = 0$$

$$\begin{aligned} & (d_t^i + \Delta_i - d_t^i) \dots \Delta_i (d_t^i + \Delta_i - d_t^{i+1}) \dots (d_t^i + \Delta_i - d_t^n) \\ & - (w_{t+1}^1)^2 (d_t^i + \Delta_i - d_t^2) \dots \Delta_i \dots (d_t^i + \Delta_i - d_t^n) \\ & - (w_{t+1}^i)^2 (d_t^i + \Delta_i - d_t^2) \dots (d_t^i + \Delta_i - d_t^{i-1}) (d_t^i + \Delta_i - d_t^{i+1}) \dots (d_t^i + \Delta_i - d_t^n) \\ & - (w_{t+1}^n)^2 (d_t^i + \Delta_i - d_t^1) \dots \Delta_i \dots (d_t^i + \Delta_i \dots (d_t^i + \Delta_i - d_t^n) = 0 \end{aligned}$$

Dividing both sides by $(d_t^i + \Delta_i - d_t^1)(d_t^i + \Delta_i - d_t^2) \dots (d_t^i + \Delta_i - d_t^n)$ we obtain

$$\begin{aligned} & \Delta_i - (w_{t+1}^1)^2 \frac{\Delta_i}{(d_t^i - d_t^1 + \Delta_i)} - \dots - (w_{t+1}^{i-1})^2 \frac{\Delta_i}{(d_t^i - d_t^{i-1} + \Delta_i)} - \\ & (w_{t+1}^i)^2 - (w_{t+1}^{i+1})^2 \frac{\Delta_i}{(d_t^i - d_t^{i+1} + \Delta_i)} - \dots - (w_{t+1}^n)^2 \frac{\Delta_i}{(d_t^i - d_t^n + \Delta_i)} = 0. \quad (4.27) \end{aligned}$$

Now, assume that for $j > i + 1$ and $j < i - 1$, we have the following inequality

$$|d_t^i - d_t^j + \Delta_i| > M \quad (4.28)$$

where M is a large number. The above assumption holds if the eigenvalues are well-separated. Simulations show and the above calculations suggest that this is the case if regressors are sufficiently rich in all directions (i.e. $|w_t^i| > 0, \forall t$). If (4.28) is satisfied then (4.27) can be approximated as

$$\Delta_i - (w_{t+1}^{i-1})^2 \frac{\Delta_i}{(d_t^i - d_t^{i-1} + \Delta_i)} - (w_{t+1}^i)^2 - (w_{t+1}^{i+1})^2 \frac{\Delta_i}{(d_t^i - d_t^{i+1} + \Delta_i)} = 0. \quad (4.29)$$

This equation will give us a polynomial of degree 3 in terms of Δ_i , which has analytical solutions depending on coefficients of polynomial in many books

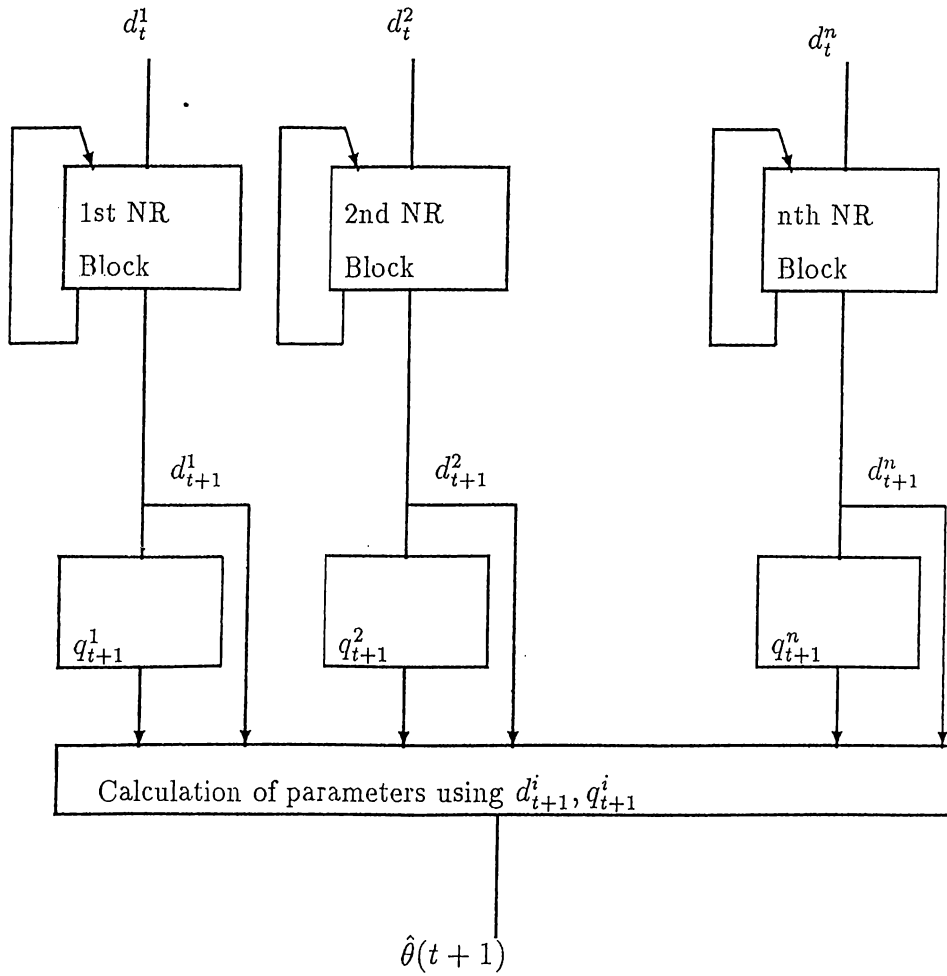


Figure 4.2: A Parallel architecture For Algorithm 1 using NR

of algebra. Hence, gives the roots d_i^i , $i = 1, 2, \dots, n$, the solutions Δ_i of (4.28), together with (4.26) give us the (approximate) roots of $p(\lambda)$. As before, note that if $w_i^i \neq 0$, then $d_{t+1}^i \neq d_t^i$.

A parallel architecture proposed using NR method which could be embedded in the Algorithm 1 given in Figure(4.2).

4.4 Asymptotic Behavior of Q_t

In this section we study the asymptotic behavior of Q_t when the regressors are persistently exciting. Note that Q_t is the resultant of SVD of $D_{t-1}^{-1} + \phi_t \phi_t^T$ i.e.,

$$D_{t-1}^{-1} + \phi_t \phi_t^T = Q_t D_t^{-1} Q_t^T \quad (4.30)$$

Dividing each sides by the maximum diagonal element of D_{t-1}^{-1} , which is positive, as $P(t)$ is positive definite $\forall t \in \mathcal{N}$, we get

$$\begin{bmatrix} 1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_n \end{bmatrix} + \frac{1}{\lambda_{\max}(D_{t-1}^{-1})} \phi_t \phi_t^T = Q_t \begin{bmatrix} \bar{r}_1 & & & \\ & \bar{r}_2 & & \\ & & \ddots & \\ & & & \bar{r}_n \end{bmatrix} Q_t \quad (4.31)$$

where $r_i = \frac{\lambda_i(D_{t-1}^{-1})}{\lambda_{\max}(D_{t-1}^{-1})}$, $i = 1, 2, \dots, n$, $\bar{r}_j = \frac{\lambda_j(D_{t-1}^{-1})}{\lambda_{\max}(D_{t-1}^{-1})}$, $j = 1, 2, \dots, n$.

We can use perturbation theory results stated in section 2.4. If we define

$$T = \begin{bmatrix} 1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_n \end{bmatrix} \quad T^{(1)} = \phi_t \phi_t^T \quad (4.32)$$

and assuming $\frac{1}{\lambda_{\max}(D_{t-1}^{-1})} \stackrel{def}{=} x$ a small number approaching zero, our problem fits the problem of finding eigenvalues and eigenvectors of $T + xT^{(1)}$. It is easy to see that the columns of Q_t are in fact eigenvectors of $T + xT^{(1)}$, so finding the columns of Q_t with the help of eigenvectors of T , we can guess the behavior of Q_t (i.e. $Q_t \rightarrow I$). To do this, we simply use the bound given in section 2.4. If $Q_t = [\underline{e}_1 \ \underline{e}_2 \ \dots \ \underline{e}_n]$ and the eigenvectors of T are as (e_1, e_2, \dots, e_n) , the standard basis, since T is diagonal. We can interpret the bound as

$$\|\underline{e}_i - e_i\| \leq |x| \frac{s_0}{(psq)^{\frac{1}{2}}} (ps^{\frac{1}{2}} + q^{\frac{1}{2}})^2 \|T^1 \phi_{i+1}^i\| \quad (4.33)$$

where to find s, p, q we need P, S , and they are defined in this example as

$$P : \mathcal{R}^n \rightarrow \text{span}(e_i), S = \frac{1}{2\pi j} \int_{\Gamma} \zeta^{-1} \begin{bmatrix} \frac{1}{1-\zeta} & & & \\ & \frac{1}{r_2-\zeta} & & \\ & & \ddots & \\ & & & \frac{1}{r_n-\zeta} \end{bmatrix} d\zeta$$

where Γ is a closed curve in complex plane containing r_i only. We can choose

$$P = \begin{bmatrix} 0 & 0 & \dots & 0 \\ & \vdots & & \\ 1 & 1 & & 1 \\ & \vdots & & \\ 0 & 0 & \dots & 0 \end{bmatrix} \leftarrow i^{th} \text{ row} \quad S = \begin{bmatrix} 0 & & & \\ & & & \\ & & \frac{1}{r_i} & \\ & & & 0 \end{bmatrix}$$

So, the norms $\|\phi_{t+1}\phi_{t+1}^T P\| < M_1$ and $\|\phi_{t+1}\phi_{t+1}^T S\| < M_2$ and $\|\phi_{t+1}\phi_{t+1}^T e_i\| < M_3$ are all bounded for bounded input, bounded output systems. So we have;

$$\|\underline{e}_i - e_i\| \leq \frac{1}{\lambda_{\max}(D_{t-1}^{-1})} M \quad (4.34)$$

where M is a combination of $M_1, M_2, M_3, \|S\|$. When the persistence of excitation condition is satisfied, we have $\lim_{t \rightarrow \infty} \lambda_{\max}(D_t) = 0$, hence (4.34) implies that we have $\lim_{t \rightarrow \infty} \|Q_t - I\| = 0$. Summarizing this result we have.

Theorem(4.10: Consider the RLS algorithm based on SVD as given in section 4.1. Assume that the regressors are bounded (i.e. for some $M > 0$, we have $\|\varphi(t)\| \leq M \forall t \in \mathcal{N}$). Under these conditions, if the persistence of excitation condition(2.7) is satisfied then we have

$$\lim_{t \rightarrow \infty} \|Q_t - I\| = 0 \quad \square$$

Remark 2: The estimate (4.34) implies that we have

$$\|Q_t - I\| \leq \frac{1}{\lambda_{\max}(D_{t-1}^{-1})} M \quad (4.35)$$

where M is a constant. Since, for a standard RLS the convergence is normally of order $(\frac{1}{t})$, we expect from (4.35) that $\|Q_t - I\|$ is of order $O(\frac{1}{t})$, for large t .

Now consider the parameter update law given by (4.12), by using the Theorem(4.1), from (4.12) that if a certain component of ϕ_t is zero, or small, then the difference of the corresponding components of α_t and α_{t-1} is zero, or small, respectively. In this case, in the calculations of section 3.2. the corresponding singular values of $P^{-1}(t)$ and $P^{-1}(t-1)$ are the same or very close to each other. Hence, by monitoring the singular values of $P^{-1}(t)$, which are calculated iteratively by the NR algorithm presented in this section. we may decide on whether a corresponding component of the parameter vector should be updated or not. This results in the following algorithm:

Algorithm2

1) Calculate the SVD of

$$D_{t-1}^{-1} + \phi_t \phi_t^T = Q_t D_t^{-1} Q_t^T \quad (A)$$

2) Update U_t

$$U_t = U_{t-1} Q_t \quad (B)$$

3) Update the parameters in the following fashion: if the difference between d_{t-1}^i and d_t^i is small do not update α_{t-1}^i in the equation;

$$\alpha_t = Q_t(\alpha_{t-1} + \gamma_t D_{t-1} \phi_t(y(t) - \phi_t \alpha_{t-1}))$$

4) Update regressors by measurement

5) Go to step 1

Using Algorithm 1 observations on the matrix Q_t shows the convergence of it to I . A simulation using the system and the input in Chapter 3 shows this behavior. At $t = 1$ the matrix is

$$Q_1 = \begin{pmatrix} 0.4082 & -0.9129 & -0.0000 & 0.0000 & -0.0000 & -0.0000 \\ 0.4082 & 0.1826 & -0.3456 & -0.7420 & -0.1448 & -0.3301 \\ 0.4082 & 0.1826 & 0.8415 & 0.0243 & -0.1213 & -0.2767 \\ 0.4082 & 0.1826 & -0.0425 & 0.0243 & 0.8667 & 0.2156 \\ 0.4082 & 0.1826 & -0.0425 & 0.0243 & -0.4285 & 0.7836 \\ 0.4082 & 0.1826 & -0.4108 & 0.6690 & -0.1721 & -0.3924 \end{pmatrix}$$

at $t = 20$ it is

$$Q_{20} = \begin{pmatrix} 0.9958 & -0.0458 & 0.0782 & -0.0045 & 0.0105 & -0.0009 \\ -0.0430 & -0.9984 & -0.0369 & 0.0019 & -0.0040 & 0.0003 \\ -0.0790 & -0.0331 & 0.9943 & 0.0433 & -0.0455 & 0.0026 \\ 0.0079 & 0.0031 & -0.0424 & 0.9990 & 0.0083 & -0.0004 \\ 0.0143 & 0.0050 & -0.0447 & 0.0063 & -0.9989 & -0.0014 \\ 0.0011 & 0.0004 & -0.0026 & 0.0003 & -0.0013 & 1.0000 \end{pmatrix}$$

at $t = 1000$ it is

$$Q_{1000} = \begin{pmatrix} 1.0000 & -0.0003 & 0.0000 & -0.0001 & 0.0001 & -0.0000 \\ 0.0003 & 1.0000 & 0.0002 & -0.0005 & 0.0005 & -0.0000 \\ -0.0000 & -0.0002 & 1.0000 & 0.0001 & -0.0001 & 0.0000 \\ -0.0001 & -0.0005 & 0.0001 & -1.0000 & -0.0008 & 0.0000 \\ -0.0001 & -0.0005 & 0.0001 & -0.0008 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 1.0000 \end{pmatrix}$$

You can see that $\|Q_{1000} - I\|$ is a small quantity which verifies the assertion made in Remark 2.

Chapter 5

Conclusion

5.1 Conclusions

In this thesis RLS, which is one of the most popular system identification schemes are derived by error least squares cost function, is analyzed in some aspects. Gauss-Seidel method and singular value decomposition applied to RLS.

The main contributions of the thesis is in the Chapter 3 and in the section 3 and section 4 of Chapter 4.

In Chapter 3, we first give a parametrization of MIMO system, based on ARMA model. We then apply Gauss-Seidel method to standard RLS algorithm, both for SISO and MIMO systems. The resultant theorems Theorem(3.1) and Theorem(3.2) give sufficient conditions for the convergence of iterative algorithms introduced, in this chapter. In simulations done in Chapter 3 we see that different choices of blocks in Gauss-Seidel method will lead to different convergence rates. Especially this observation may lead to further analysis in this area. For example finding the most convergent algorithm by varying the blocks will be a good result.

In Chapter 4 after applying SVD on the covariance matrix P_t we derive an algorithm which has also the ability of monitoring the eigenvalues (singular values) of P_t either standard SVD calculators or the parallel method we propose in section 4.3 could be used for this purpose. We also give a sufficient condition which guarantees the convergence of the parallel algorithm proposed in this chapter. The proposed parallel method is based on the well-known Newton-Raphson algorithm for finding the roots of a polynomial, whose roots are all

real and satisfy an interesting interlacing property. In section 4.4 asymptotic behavior of Q_t is investigated. A perturbation theory based analysis gives us a bound on the norm $\|Q_t - I\|$. This norm goes to zero if the excitation is persistent. This behavior gives us a way to modify Algorithm 1. Again monitoring singular values of P_t and updating the most excited modes, a new algorithm is derived, Algorithm 2. We also present a simulation result showing the behavior of Q_t .

As a concluding remark we believe that RLS is a modifiable material in system identification in many ways.

References

- [1] D. Bertsekas, J. N. Tsitsiklis, *Parallel and Distributed Computations: Numerical Methods*, MA: Prentice-Hall INC. , Englewood Cliffs, New Jersey, 1989
- [2] F. M. F. Gaston, Prof. G. W. Irwin, "Systolic Kalman Filtering: an overview", *IEE PROCEEDINGS*, Vol. 137, Pt.D , No. 4, 1990
- [3] G. Kreisselmeier, "Stabilized Least-Squares Type Adaptive Identifiers", *IEEE Trans.on A.C.*, Vol.35, No.3, 1990
- [4] G. C. Goodwin, K. S. Sin, *Adaptive Filtering Prediction And Control*, MA: Prentice-Hall, Inc. , Englewood Cliffs, New Jersey, 1984
- [5] G. H. Golub, C. Reinsch , "Singular value decomposition and least squares solutions, " *Numer. Math.* , vol. 14, pp. 403-420, 1970
- [6] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, MA: Academic Press, New York, 1977
- [7] J. M. Jover, T. Kailath, "A Parallel Architecture for Kalman Filter MEasurement Update and Parameter Estimation," *Automatica*, vol. 22, No. 1, pp 43-57 ,1986.
- [8] K. J. Aström, B. Wittenmark, *Adaptive Control*, MA: Addison-Wesley, USA, 1989
- [9] M. E. Salgado, G. C. Goodwin, R. H. Middleton, "Modified least squares algorithm incorporating exponential resetting and forgetting", *INT. J. CONTROL*, Vol. 47, No. 2, pp 477-491, 1988.
- [10] N. R. Sropada, D. G. Fisher, "Improved least squares identification", *INT. J. CONTROL*, Vol. 46, No. 6, pp 1889-1913, 1987.
- [11] S. Bittanti, P. Bolzern, M. Campi, "Recursive Least-Squares Identification Algorithms with Incomplete Excitation: Convergence Analysis and

- Application to Adaptive Control”, *IEEE Trans. Automat. Contr.* Vol. 35, No. 12. 1990
- [12] Tosio Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, 2nd edition MA: Springer-Verlag, Berlin, 1982.
- [13] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C*, MA: Cambridge University Press, USA, 1988
- [14] Virginia C. Klema, Alan J. Laub, “The Singular Value Decomposition: Its Computation and Some Applications,” *IEEE Trans.on AC.*, vol. AC-25, No. 2 pp. 164-176, 1980.
- [15] V. Lakshimikantham, D. Trigiante, *Theory of Difference Equations: Numerical Methods and Applications*, MA: Academic Press, San Diego. USA, 1988