System Identification via Nuclear Norm Regularization

A Thesis

SUBMITTED TO THE FACULTY OF THE

UNIVERSITY OF MINNESOTA

BY

Harshad Deshmane

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Professor Mihailo R. Jovanović

July, 2014

Acknowledgments

I would like to thank my adviser Professor Mihailo Jovanović, for his constant support and guidance, and for pushing me to always be better. I truly admire his attitude and work ethic, and constantly strived to emulate the same. I am fortunate to have had the opportunity to work with him.

I am grateful to Professors Gary Balas and Jarvis Haupt for serving on my thesis committee. I wish to thank Professors Tryphon Georgiou and William Robbins for their understanding and encouragement during my time working for them. I also want to thank Professors Peter Seiler Jr., Zhi-Quan Luo and Stergios Roumeliotis, for teaching me various classes, and providing valuable criticism and appreciation.

I want to thank Professor Steven P. Schneider and Dennis Berridge at Purdue University, for their help with this project.

I wish to thank Armin Zare, with whom I had many useful discussions. I would also like to thank Fu, Xiaofan, Neil, Binh, Hamza, Yongxin, Lipeng, Francesca and Sepideh, for helping me in various ways, not least of which was to make me feel comfortable, and a part of the group.

My time in Minnesota was made memorable by the company of my wonderful friends: Virendra, Mayur, Shreyas, Abhijeet, Ganesh, Udayan, Tejas, Prashant and Kewal.

Finally, but most importantly, I want to express my deepest gratitude toward my parents, and my parents in the US: my brother Chinmay, and my sister-in-law Shivani. I want to take this opportunity to acknowledge the many sacrifices my mother has made over the years; and to tell her that I love her.

I do want to thank Freddie as well.

Dedication

To my grandparents Annasaheb and Sushila,
and my mother Jayashri

ABSTRACT

System Identification via Nuclear Norm Regularization

We study the subspace method for system identification, and look at algorithms that rely on nuclear norm regularization for solving this problem. We introduce our own algorithm for solving the problem, based on the alternating direction method of multipliers (ADMM). Our algorithm involves an iterative minimization step, which is solved using line search methods. We demonstrate the effectiveness of our algorithm on a particular real world example, as well as two benchmark examples. In addition, we compare the computational efficiency of our algorithm to that of other existing algorithms for solving the nuclear norm system identification problem, observing that for single-input single-output systems, our algorithm is faster than an existing interior-point method. We also note that our algorithm converges the fastest when we use a gradient descent direction in the iterative minimization step of the ADMM.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Linear time-invariant (LTI) systems find applications in modeling dynamical phenomena in a wide range of areas. The problem of fitting an LTI model to observed data, to characterize a system about which little is known, is therefore important if we want to have any useful discussion about the behavior of the system. This thesis looks at such an input-output system identification problem, wherein the goal is to fit a discrete-time linear time-invariant system model to measured input-output data.

We restrict our attention to subspace methods for system identification [Ljung, 1999; Verhaegen, 1994], which have encouraged the formulation of a convex optimization problem consisting of a nuclear norm[1] cost function [Liu and Vandenberghe, 2009; Mohan and Fazel, 2010; Fazel *et al.*, 2013; Liu *et al.*, 2013]. This optimization step serves to pre-condition the output sequence, before it is used to obtain a system realization. This idea arises form the fact that, minimizing the rank of a (block) Hankel-type matrix results in a low order model for the system; and the nuclear norm term serves as a proxy for the rank function. That, nuclear norm optimization can be used as a convex heuristic for minimizing the rank of a matrix, was proposed in [Fazel *et al.*, 2001; Fazel, 2002], wherein the authors showed that, the nuclear norm of a matrix is the convex envelope[2] of the rank function over the

[1] Sum of singular values of a matrix.

[2] The convex envelope of a function $f : \mathcal{C} \to \mathbb{R}$, on the set $\mathcal{C} \subseteq \mathbb{R}^n$ is the largest convex function that lower bounds f at each point in $\mathcal{C}$.

unit spectral norm ball[3]. This heuristic is further motivated by the observation that, the solution of the nuclear norm optimization problem tends to be low-rank.

Nuclear norm optimization, in itself, has a range of applications. For symmetric positive semidefinite matrices, which arise in reduced order controller synthesis [Pare, 2000; Mesbahi, 1999], the nuclear norm heuristic reduces to the trace minimization heuristic, since for a symmetric positive semidefinite matrix, the nuclear norm equals its trace. For diagonal matrices, the rank is simply the number of nonzero elements on the diagonal, i.e., the cardinality of the vector formed from the diagonal elements; and the rank minimization problem then reduces to that of finding the sparsest vector in a convex set. Or correspondingly, the nuclear norm minimization problem reduces to the problem of minimizing the ℓ_1 norm of a vector over a convex set, which is a well-known heuristic for cardinality minimization [Chen and Donoho, 1994; Hassibi *et al.*, 1999]. The ℓ_1 norm techniques have been highly successful in practice, backed up with theoretical results guaranteeing exact recovery of sparse vectors [Candès and Tao, 2005; Candès *et al.*, 2006]. Along similar lines, statistical guarantees for exact recovery of low-rank matrices via nuclear norm optimization have also been established recently [Recht *et al.*, 2007].

Other applications of nuclear norm optimization include the matrix completion problem [Candès and Recht, 2009; Candès and Plan, 2010] arising in machine learning and computer vision, minimal system realization and stochastic realization problems encountered in control and signal processing, and of course, input-output system identification, which is the focus of this thesis.

In this work, we propose an algorithm for solving the system identification problem via nuclear norm regularization. Our algorithm is based on the alternating direction method of multipliers (ADMM), and it is different from the other existing ADMM algorithms in that, it involves an iterative minimization step, which is solved using line search methods. We use this algorithm to solve a specific real world problem, that of identifying a model for the PCB-132 pressure sensor. In addition, we demonstrate the performance of our algorithm on two more examples from DaISy (Database for the Identification of Systems) [DeMoor

[3]The set of all matrices whose spectral norm, or largest singular value, is upper bounded by one. The nuclear norm is the dual of the spectral norm.

et al., 1997], and compare it to existing algorithms.

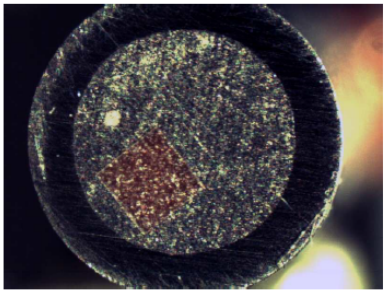
Outline. In section 1.1, we provide a brief description of the pressure sensor application that motivated this work, while section 1.2 introduces notation used in this thesis. In chapter 2, we provide some background on the classical subspace method for system identification, and introduce the nuclear norm regularized problem considered in this thesis. In chapter 3, we discuss existing algorithms, and propose our own *Iterative ADMM* algorithm, for solving the optimization problem. We present computational results, and compare the performance of our proposed algorithm to that of the existing algorithms, in chapter 4. Chapter 5 concludes this thesis.

1.1 PCB-132 pressure sensor example

Our interest in nuclear norm system identification was stimulated when we were presented with the problem of identifying a model for the PCB-132 pressure sensors, being used by the research group of Prof. Steven P. Schneider in the Aerospace Engineering department at Purdue University. The PCB-132 sensors are piezoelectric pressure transducers, high-pass filtered at 11 kHz, with a resonant frequency of above 1 MHz. The group has had some success in using the PCB-132 sensors for measuring instability waves in a hypersonic wind tunnel. However, they are faced with the challenge of the calibration provided by the manufacturer, not being relevant or accurate enough for the purposes of instability measurements. This is because, the shock strength used by the manufacturer for calibration generates a response that is not necessarily similar to that generated by an instability wave.

For calibrating the sensor more accurately, experiments were performed, which involved providing a step input to the sensor in the form of a shock wave, generated using a shock tube. The rationale behind using a step input is that it would excite the higher frequencies of the sensor, which are of interest. The sensor output is normalized, and averaged over several individual responses. More details on the calibration experiments can be found in [Berridge and Schneider, 2012; Abney *et al.*, 2013].

Figure 1.1: PCB-132 pressure sensor



The epoxy covering is removed here. The brown patch is the sensing element.

1.2 Notation

In this thesis, the m-dimensional Euclidean space is denoted by $\mathbb{R}^m$, while the set of all $p \times q$ real matrices is denoted by $\mathbb{R}^{p \times q}$. $\| v \|$ denotes the Euclidean norm of the vector $v \in \mathbb{R}^m$, while for any $p \times q$ real matrix M, $\| M \|$, $\| M \|_F$ and $\| M \|_*$ denote its spectral, Fröbenius and nuclear norms respectively. The notation vec(M) stands for the vector formed by stacking the columns of M on top of each other, while $\sigma_{max}(M)$ and $\sigma_{min}(M)$ stand for the largest and smallest singular value of M respectively. $\mathbb{S}^n$ stands for the set of all symmtric $n \times n$ matrices, while the expression $M \succeq 0$ imples that M is a symmetric positive semidefinite matrix. The identity matrix and identity map are denoted by I and $\mathcal{I}$ respectively.

Chapter 2

System Identification via Nuclear Norm Regularization

We begin this chapter with a discussion on how input-output system identification can be thought of as a rank minimization problem, or in other words, how solving a nuclear norm optimization problem can lead to a suitable estimate for the order of the system we wish to realize. We also provide a brief description of the classical subspace method for system identification. Finally, we introduce the nuclear norm regularized optimization problem.

2.1 Subspace method for system identification

Having been given input and output sequences, respectively $u(t) \in \mathbb{R}^m$ and $\tilde{y}(t) \in \mathbb{R}^p, t = 0, ..., N-1$, our goal is to fit a discrete-time linear time-invariant state-space model of the following form, to these problem data:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ \tilde{y}(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ denotes the n-dimensional state vector of the system at time t. That is, we want to obtain estimates for the model order n, the matrices A, B, C, D, and the initial state $x(0)$, to completely describe the identified state-space model.

We first rewrite the system (2.1) in matrix form, as follows,

$$\widetilde{Y} = GX + FU, \tag{2.2}$$

where

$$G = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{r-1} \end{bmatrix}, \quad F = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{r-2}B & CA^{r-3}B & CA^{r-4}B & \cdots & D \end{bmatrix}, \tag{2.3}$$

$$X = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-r) \end{bmatrix},$$

and U and $\widetilde{Y}$ are the block Hankel matrices constructed as follows, from the given input and output sequences respectively,

$$U = \begin{bmatrix} u(0) & u(1) & u(2) & \cdots & u(N-r) \\ u(1) & u(2) & u(3) & \cdots & u(N-r+1) \\ \vdots & \vdots & \vdots & & \vdots \\ u(r-1) & u(r) & u(r+1) & \cdots & u(N-1) \end{bmatrix},$$

$$\widetilde{Y} = \begin{bmatrix} \tilde{y}(0) & \tilde{y}(1) & \tilde{y}(2) & \cdots & \tilde{y}(N-r) \\ \tilde{y}(1) & \tilde{y}(2) & \tilde{y}(3) & \cdots & \tilde{y}(N-r+1) \\ \vdots & \vdots & \vdots & & \vdots \\ \tilde{y}(r-1) & \tilde{y}(r) & \tilde{y}(r+1) & \cdots & \tilde{y}(N-1) \end{bmatrix}.$$

Here, r is the number of block rows of the output (or input) Hankel matrix. The value for r is chosen such that pr is an upper bound for the order of the estimated system.

Multiplying (2.2) on the right with $U^{\perp}$, the orthogonal matrix whose columns span the nullspace of U, we get

$$\widetilde{Y}U^{\perp} = GXU^{\perp}. \tag{2.4}$$

We observe that if there is no rank cancellation in the product $GXU^{\perp}$ (a property that holds generically for randomly chosen inputs), the rank of $\widetilde{Y}U^{\perp}$ equals the rank of the *extended observability matrix* G - which is equal to the true order of the system - and that the range of $GXU^{\perp}$, or in turn, the range of $\widetilde{Y}U^{\perp}$ equals the range of G.

In the following, we describe a procedure to realize a state-space model starting with the matrix $\widetilde{Y}U^{\perp}$, in the case when the given output, y, is noisy, which is the case we consider in our experiments; readers interested in further details can refer to [Liu *et al.*, 2013, §2], or more classically, to [Ljung, 1999, §10.6]. We start by computing a singular value decomposition (SVD) of the matrix $\widetilde{Y}U^{\perp}$. We then set a threshold n, for the rank of $\widetilde{Y}U^{\perp}$ (which is also the approximate system order), and accordingly truncate its SVD to obtain a matrix $\hat{G}$, whose columns span the range of the rank n approximation of $\widetilde{Y}U^{\perp}$, where

$$\hat{G} = \begin{bmatrix} \hat{G}_0 \\ \hat{G}_1 \\ \vdots \\ \hat{G}_{r-1} \end{bmatrix}.$$

From (2.3), we set C to be the first block of $\hat{G}$, i.e., $C = \hat{G}_0$; and we solve the following system of linear equations in the least-squares sense to obtain A:

$$\begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \\ \vdots \\ \hat{G}_{r-1} \end{bmatrix} = \begin{bmatrix} \hat{G}_0 \\ \hat{G}_1 \\ \vdots \\ \hat{G}_{r-2} \end{bmatrix} A.$$

We then solve for matrices B and D, and the initial state $x(0)$ in the least-squares sense using the following overdetermined system of linear equations, which follows from (2.1):

$$CA^t x(0) + \sum_{k=0}^{t-1} CA^{t-k-1}Bu(k) + Du(t) = \tilde{y}(t), \qquad t = 0, ..., N-1.$$

2.2 System identification via nuclear norm regularization

Nuclear norm optimzation can be incorporated into the standard subspace system identification algorithm, as a preprocessing step that provides an optimal output sequence, which can then be used to obtain the system realization. Ideally, we would want this optimal output to be as close to the measured output as possible. Our cost function, therefore, consists of a measure of the fidelity of our optimization variable to the true data. It was es-

tablished in the previous section that the rank of the matrix[1] $YU^{\perp}$ equals the true order of the system. Ideally, we would like the identified model to be of low order, since it simplifies analysis. Therefore, it would make sense to add another term to the cost function, one that penalizes the rank, or in turn, the nuclear norm of $YU^{\perp}$. We now formalize these ideas by presenting the general nuclear norm regularized problem that is at the center of this work:

$$\underset{y}{\text{minimize}} \quad \gamma \parallel H(y)U^{\perp} \parallel_{*} + \frac{1}{2} \parallel \mathcal{A}(y) - b \parallel^{2} . \tag{2.5}$$

Here $b \in \mathbb{R}^{pN}$ is the vector obtained by vectorizing $\tilde{y} = \big(\tilde{y}(0) \;\; \tilde{y}(1) \ldots \tilde{y}(N-1)\big)$, the sequence of output measurements, and $U^{\perp} \in \mathbb{R}^{N-r+1 \times c}$ is as defined in the previous section. The matrix $y = \big(y(0) \;\; y(1) \ldots y(N-1)\big)$ is the optimization variable, $\mathcal{A} : \mathbb{R}^{p \times N} \rightarrow \mathbb{R}^{pN}$ is a linear map, and γ is a positive regularization parameter. The matrix $H(y) \in \mathbb{R}^{pr \times N-r+1}$ is nothing but the matrix Y above; we introduce the Hankel operator, $H(\cdot)$, for the purpose of generalizing the matrix Y in order to make it more amenable to the reformulations considered later. We formally define $H : \mathbb{R}^{p \times N} \rightarrow \mathbb{R}^{pr \times N-r+1}$ as a linear map that admits y and constructs the block Hankel matrix $H(y)$. As described above, the objective is to match y to the measured outputs, $\tilde{y}$, while minimizing the nuclear norm of $H(y)U^{\perp}$. The optimization is solved for a range of values of γ, that yields a curve characterizing the tradeoff between the nuclear norm of $H(y)U^{\perp}$ and the fitting error. The optimal solution, y^{*}, is selected depending on the desired tradeoff between the system order and fitting error. A system realization is then obtained from the singular value decomposition of $H(y^{*})U^{\perp}$, as described in section 2.1.

[1]Here, Y is the block Hankel matrix formed from the optimization variable y, instead of the measured outputs, $\tilde{y}$.

Chapter 3

Solving the Optimization Problem

In this chapter, we discuss existing algorithms for solving the nuclear norm regularized problem (2.5), and propose our own version of the alternating direction method of multipliers. We also compare the cost per iteration of the different algorithms.

3.1 Interior-point method

Consider the following general nuclear norm optimization problem:

$$\underset{x}{\text{minimize}} \quad \| \mathcal{B}(x) - B \|_*,$$

where $x \in \mathbb{R}^{pN}, B \in \mathbb{R}^{pr \times c}$, and $\mathcal{B}$ is linear map. This problem can be expressed as a semidefinite program (SDP) [Vandenberghe and Boyd, 1996; Fazel *et al.*, 2001] as follows:

$$\begin{aligned} \underset{x,U,V}{\text{minimize}} \quad & \frac{1}{2}(\operatorname{tr} U + \operatorname{tr} V) \\ \text{subject to} \quad & \begin{bmatrix} U & (\mathcal{B}(x) - B)^T \\ (\mathcal{B}(x) - B) & V \end{bmatrix} \succeq 0, \end{aligned}$$

where $U \in \mathbb{S}^q$ and $V \in \mathbb{S}^{pr}$. This idea was explored in [Liu and Vandenberghe, 2009]; the problem (2.5) - with $\mathcal{A}(y) = \text{vec}(y)$ - was reformulated as a SDP, and an interior-point method, tailored to exploit the problem structure, was developed, that solves the Newton system more efficiently than general-purpose SDP solvers. More specifically, the cost per iteration of the custom interior-point method grows roughly as $\mathcal{O}(p^3 r N^2 c)$, or as $\mathcal{O}(p^4 N^4)$,

if for simplicity $r = \mathcal{O}(N)$ and $c = \mathcal{O}(pN)$, as against $\mathcal{O}(p^6N^6)$ for general-purpose solvers. The interior-point method converges to a solution of high accuracy within a few tens of iterations; however, it has a large memory requirement and high computational cost per iteration.

3.2 Alternating direction method of multipliers

The alternating direction method of multipliers (ADMM) is a popular first order method for dealing with distributed and large-scale convex optimization problems [Boyd *et al.*, 2011; Lin *et al.*, 2013a; Wu and Jovanović, 2014a]. As a consequence, it has been employed in a wide range of applications; see e.g., [Zhu *et al.*, 2009; Goldstein and Osher, 2009; Forero *et al.*, 2010; Lin *et al.*, 2012b,a; Dhingra *et al.*, 2012; Lin *et al.*, 2013b; Jovanović and Lin, 2013; Dörfler *et al.*, 2013; Lin *et al.*, 2013c; Zare *et al.*, 2014; Zoltowski *et al.*, 2014; Wu and Jovanović, 2014b; Wu *et al.*, 2014; Dörfler *et al.*, 2014; Lin *et al.*, 2014; Dhingra *et al.*, 2014]. With a simple reformulation, the problem (2.5) can be split into two variables, whereby it is very well-suited to be solved using ADMM, which offers fast convergence to a solution of modest accuracy. This approach was considered in [Fazel *et al.*, 2013; Liu *et al.*, 2013]. Here we briefly discuss these two algorithms.

3.2.1 The *Primal ADMM* of Fazel *et al.*

Again, we consider the problem (2.5). In this case, $\mathcal{A}(y) = \mathrm{vec}(y)$ and $b = \mathrm{vec}(\tilde{y})$, $\tilde{y}$ being the sequence of measured outputs: $\tilde{y} = \big(\tilde{y}(0)\ \ \tilde{y}(1) \dots \tilde{y}(N-1)\big)$. Here we define the adjoint of the operator H as,

$$H^*(Q) = H^* \begin{pmatrix} q_{00} & q_{01} & \cdots & q_{0,N-r} \\ q_{10} & q_{11} & \cdots & q_{1,N-r} \\ \vdots & \vdots & & \vdots \\ q_{r-1,0} & q_{r-1,1} & \cdots & q_{r-1,N-r} \end{pmatrix}$$
$$= \begin{pmatrix} q_{00} & q_{01}+q_{10} & q_{02}+q_{11}+q_{20} & \cdots & q_{r-1,N-r} \end{pmatrix} \in \mathbb{R}^{p\times N}.$$

Introducing the substitution $Z = -H(y)U^{\perp}$, we can reformulate (2.5) as a constrained optimization problem,

$$\begin{aligned} \underset{Z,y}{\text{minimize}} \quad & \gamma \parallel Z \parallel_* + \frac{1}{2} \parallel y - \tilde{y} \parallel_F^2 \\ \text{subject to} \quad & Z + H(y)U^{\perp} = 0. \end{aligned} \tag{3.1}$$

We then form the augmented Lagrangian associated with (3.1),

$$\mathcal{L}_\rho(Z, y, \Lambda) = \gamma \parallel Z \parallel_* + \frac{1}{2} \parallel y - \tilde{y} \parallel_F^2 - \operatorname{tr}\left(\Lambda^T \left(Z + H(y)U^{\perp}\right)\right) + \frac{\rho}{2} \parallel Z + H(y)U^{\perp} \parallel_F^2,$$

where Λ is the Lagrange multiplier (or the dual variable), and ρ is a positive scalar. To find a minimizer of the constrained problem (3.1), ADMM iteratively minimizes $\mathcal{L}_\rho$ with respect to Z and then y, followed by an update of the dual variable. The Z-minimization step has a closed form solution, but this is not the case for the y-minimization step, because of the complicated quadratic terms. The authors of [Fazel *et al.*, 2013] got around this issue by adding a suitable proximal term to "cancel" out the complicated terms. In this case the y-update is given by

$$\begin{aligned} y^{k+1} = \arg\min_y \; g(y) :=& \mathcal{L}_\rho(Z^{k+1}, y, \Lambda^k) \\ & + \frac{\rho}{2}\left[r \parallel y - y^k \parallel^2 - \left(y - y^k\right)^T H^*\left(H\left(y - y^k\right)U^{\perp}\left(U^{\perp}\right)^T\right)\right], \end{aligned} \tag{3.2}$$

where r is as defined in section 2.1. For the interested reader, the convergence analysis of this approach is contained in [Fazel *et al.*, 2013]. We now describe the algorithm.

Primal ADMM

Initialize y^0, Λ^0. Select $\rho > 0, \sigma = \frac{\rho}{\rho r + 1}$ and $\tau \in \left(0, \frac{\sqrt{5}+1}{2}\right)$.

1. Compute the SVD

$$\begin{aligned} W\Sigma V^T &= -H(y^k)U^{\perp} + \frac{1}{\rho}\Lambda^k, \quad \text{and set} \\ Z^{k+1} &= W \max\left\{\Sigma - \frac{\gamma}{\rho} I, 0\right\} V^T. \end{aligned}$$

2. $y^{k+1} = y^k - \sigma\left(H^*\left(\left(-\frac{1}{\rho}\Lambda^k + H(y^k)U^{\perp} + Z^{k+1}\right)\left(U^{\perp}\right)^T\right) + \frac{1}{\rho}\left(y^k - \tilde{y}\right)\right)$.

3. $\Lambda^{k+1} = \Lambda^k - \tau\rho\left(Y^{k+1} + H(y^{k+1})U^\perp\right)$.

If a termination criterion is not satisfied, go to 1.

We define the primal and dual residuals, respectively, as,

$$r^{k+1} = \| Z^{k+1} + H(y^{k+1})U^\perp \|_F, \quad s^{k+1} = \rho \| Z^{k+1} - Z^k \|_F,$$

and check to see if they satisfy the following termination criteria:

$$r^{k+1} < \sqrt{prc} \cdot \epsilon_{abs} + \epsilon_{rel} \cdot max\left\{ \| H(y^{k+1})U^\perp \|_F, \| Z^{k+1} \|_F \right\}$$
$$s^{k+1} < \sqrt{prc} \cdot \epsilon_{abs} + \epsilon_{rel} \cdot \| \Lambda^{k+1} \|_F,$$

where $\epsilon_{abs} = 10^{-5}$ and $\epsilon_{rel} = 10^{-3}$. Note that these are slightly different from the termination criteria used in [Fazel *et al.*, 2013].

3.2.2 Liu *et al.*'s ADMM algorithm

The ADMM implementation considered in [Liu *et al.*, 2013] is similar to that of [Fazel *et al.*, 2013]. The first of two main differences is that, they include instrumental variables and weight matrices in their problem formulation, that are commonly used in state-of-the-art subspace methods. The other difference is in the y-minimization step, wherein the gradient $\nabla_y \mathcal{L}_\rho$ is set to zero, and the resulting system of linear equations is solved to obtain the minimizer y^{k+1}. The authors also provide two techniques that improve the efficiency of the algorithm.

3.3 The proposed *Iterative ADMM* algorithm

We propose an *Iterative ADMM* algorithm for solving the nuclear norm system identification problem, that differs from the ADMM algorithms discussed in the previous section in the y-minimization step. Notice from (3.2) that y^{k+1} in step 2 of the *Primal ADMM* algorithm is not the minimizer of $\mathcal{L}_\rho$, but of $g(y)$, which is the sum of $\mathcal{L}_\rho$ and some quadratic terms. Our proposed algorithm uses a line search method at every ADMM iteration to solve, to a

reasonable accuracy, for the y that minimizes $\mathcal{L}_\rho$. The line search method makes use of a gradient descent direction, a Newton and a quasi-Newton direction to solve for y.

An accelerated version of the first order proximal gradient method was applied in [Toh and Yun, 2009] to the nuclear norm regularized linear least squares problem arising in matrix completion. This prompted us to try to apply the proximal gradient method to nuclear norm system identification. Indeed in our proposed algorithm, the $\mathbf{y}$-minimization step, with a gradient descent direction, is a special case of the proximal gradient approach.

We start with a formulation similar to (2.5):

$$\underset{\mathbf{y}}{\text{minimize}} \quad \gamma \parallel \mathbf{H}(\mathbf{y})U^\perp \parallel_* + \frac{1}{2} \parallel \mathbf{y} - b \parallel^2 . \tag{3.3}$$

There are two subtle differences, however. The optimization variable, in this case, is $\mathbf{y} \in \mathbb{R}^{pN}$, and is related to $y \in \mathbb{R}^{p\times N}$ in (2.5) as $\mathbf{y} = \text{vec}(y)$. Also, the operator $\mathbf{H} : \mathbb{R}^{pN} \to \mathbb{R}^{pr\times N-r+1}$ in this case, admits $\mathbf{y}$ to construct the block Hankel matrix $\mathbf{H}(\mathbf{y})$. The adjoint of $\mathbf{H}$ is given by,

$$\begin{aligned}\mathbf{H}^*(Q) &= \mathbf{H}^* \begin{pmatrix} q_{00} & q_{01} & \cdots & q_{0,N-r} \\ q_{10} & q_{11} & \cdots & q_{1,N-r} \\ \vdots & \vdots & & \vdots \\ q_{r-1,0} & q_{r-1,1} & \cdots & q_{r-1,N-r} \end{pmatrix} \\ &= \text{vec} \begin{pmatrix} q_{00} & q_{01}+q_{10} & q_{02}+q_{11}+q_{20} & \cdots & q_{r-1,N-r} \end{pmatrix} \in \mathbb{R}^{pN}.\end{aligned}$$

Again, we reformulate (3.3) as a constrained optimization problem by introducing the substitution $\mathbf{H}(\mathbf{y})U^\perp = Z$,

$$\begin{aligned} \underset{\mathbf{y},Z}{\text{minimize}} \quad & \gamma \parallel Z \parallel_* + \frac{1}{2} \parallel \mathbf{y} - b \parallel^2 \\ \text{subject to} \quad & \mathbf{H}(\mathbf{y})U^\perp - Z = 0. \end{aligned} \tag{3.4}$$

We then form the augmented Lagrangian associated with (3.4),

$$\mathcal{L}_\rho(\mathbf{y}, Z, \Lambda) = \gamma \parallel Z \parallel_* + \frac{1}{2} \parallel \mathbf{y} - b \parallel^2 + \text{tr}\left(\Lambda^T \left(\mathbf{H}(\mathbf{y})U^\perp - Z\right)\right) + \frac{\rho}{2} \parallel \mathbf{H}(\mathbf{y})U^\perp - Z \parallel_F^2 . \tag{3.5}$$

Next we describe the iterative $\mathbf{y}$-minimization step of the algorithm.

3.3.1 The iterative minimization step

For minimizing $\mathcal{L}_\rho$ with respect to $\mathbf{y}$, we make use of line search methods, which are very popular for unconstrained optimization. In this approach, a suitable search direction is chosen, and a new iterate of the optimization variable, that gives a lower function value than the previous one, is searched for in this direction, using an appropriate step-size rule. For details on most of the discussion in this subsection, readers can refer to [Nocedal and Wright, 2006].

Consider the following general unconstrained minimization problem:

$$\underset{x}{\text{minimize}} \quad f(x). \tag{3.6}$$

For this problem, an iteration of a line search method is given by

$$x^{\ell+1} = x^\ell + \alpha^\ell d^\ell,$$

where $\alpha^\ell > 0$ is called the step-size, and d^ℓ is the search direction, which is usually a direction of descent, i.e., for which $\left(d^\ell\right)^T \nabla f^\ell < 0$.

We now briefly discuss the different choices of the descent direction and the step-size, that have been considered in this work.

Descent direction. The descent direction is usually of the form

$$d^\ell = -\left(D^\ell\right)^{-1} \nabla f^\ell,$$

where D^ℓ is a symmetric and nonsingular matrix. In our case, the problem can be expressed as[1]

$$\underset{\mathbf{y}}{\text{minimize}} \quad \mathcal{L}_\rho\left(\mathbf{y}\right),$$

where $\mathcal{L}_\rho$ is given by (3.5). The gradient of $\mathcal{L}_\rho$ with respect to $\mathbf{y}$ is given by

$$\nabla_{\mathbf{y}} \mathcal{L}_\rho\left(\mathbf{y}^{k+1,\ell}\right) := \rho \mathbf{H}^* \left(\left(\mathbf{H}\left(\mathbf{y}^{k+1,\ell}\right) U^\perp - \left(Z^k - \left(\rho^{-1}\right)\Lambda^k\right)\right)\left(U^\perp\right)^T\right) + \left(\mathbf{y}^{k+1,\ell} - b\right). \tag{3.7}$$

Here, $k+1$ stands for the outer (ADMM) iteration, which remains constant over the inner (line search) iterations. So for our problem, the line search iteration is given by

$$\mathbf{y}^{k+1,\ell+1} = \mathbf{y}^{k+1,\ell} - \alpha^\ell \left(D^\ell\right)^{-1} \nabla_{\mathbf{y}} \mathcal{L}_\rho\left(\mathbf{y}^{k+1,\ell}\right).$$

[1]Notice the argument of $\mathcal{L}_\rho$: when we minimize with respect to $\mathbf{y}$, Z and Λ are constant.

We consider the gradient descent, Newton and quasi-Newton methods. In the gradient descent method, the matrix D^ℓ is simply identity. In Newton's method, it exactly equals the Hessian $\nabla^2 \mathcal{L}_\rho(\mathbf{y})$. The gradient $\nabla_{\mathbf{y}} \mathcal{L}_\rho(\mathbf{y})$ given in (3.7) is linear in $\mathbf{y}$, and hence the Hessian is a constant matrix that depends only on the entries of the matrix $U^\perp \left(U^\perp\right)^T$. In quasi-Newton methods, D^ℓ is an approximation to the Hessian that is obtained using information about the gradient evaluated at the present and past iterates. In particular, we use the *BFGS formula* for approximating the inverse of the Hessian, which is named after its inventors, Broyden, Fletcher, Goldfarb, and Shanno. Let us first define:

$$\Delta \mathbf{y}^{k+1,\ell-1} = \mathbf{y}^{k+1,\ell} - \mathbf{y}^{k+1,\ell-1}, \quad \Delta \mathbf{g}^{k+1,\ell-1} = \nabla_{\mathbf{y}} \mathcal{L}_\rho \left(\mathbf{y}^{k+1,\ell}\right) - \nabla_{\mathbf{y}} \mathcal{L}_\rho \left(\mathbf{y}^{k+1,\ell-1}\right).$$

Now, if we let $\Pi^\ell := \left(D^\ell\right)^{-1}$, the BFGS formula for approximating the Hessian inverse is given by

$$\Pi^\ell = \left(I - \frac{\Delta \mathbf{y}^{k+1,\ell-1} \left(\Delta \mathbf{g}^{k+1,\ell-1}\right)^T}{\left(\Delta \mathbf{g}^{k+1,\ell-1}\right)^T \Delta \mathbf{y}^{k+1,\ell-1}}\right) \Pi^{\ell-1} \left(I - \frac{\Delta \mathbf{g}^{k+1,\ell-1} \left(\Delta \mathbf{y}^{k+1,\ell-1}\right)^T}{\left(\Delta \mathbf{g}^{k+1,\ell-1}\right)^T \Delta \mathbf{y}^{k+1,\ell-1}}\right) + \frac{\Delta \mathbf{y}^{k+1,\ell-1} \left(\Delta \mathbf{y}^{k+1,\ell-1}\right)^T}{\left(\Delta \mathbf{g}^{k+1,\ell-1}\right)^T \Delta \mathbf{y}^{k+1,\ell-1}}.$$

Step-size. We now discuss the different step-size rules considered, in the context of the simple example (3.6). Backtracking line search is a popular method, which requires that the step-size α^ℓ satisfy the *sufficient decrease* condition, also called the *Armijo condition*, given by

$$f\left(x^\ell + \alpha^\ell d^\ell\right) \leq f\left(x^\ell\right) + c_1 \alpha^\ell \left(\nabla f^\ell\right)^T d^\ell, \tag{3.8}$$

for some constant $c_1 \in (0,1)$. Another step-size rule is the Wolfe line search, which requires α^ℓ to satisfy another condition, called the *curvature condition*, in addition to the sufficient decrease condition (3.8). The curvature condition is given by

$$\left(\nabla f\left(x^\ell + \alpha^\ell d^\ell\right)\right)^T d^\ell \geq c_2 \left(\nabla f^\ell\right)^T d^\ell,$$

where $c_2 \in (c_1, 1)$. We also implement a step-size selection rule based on the Barzilai-Borwein methods [Barzilai and Borwein, 1988], given by

$$\alpha^\ell = \frac{\langle x^\ell - x^{\ell-1}, \nabla f^\ell - \nabla f^{\ell-1} \rangle}{\langle \nabla f^\ell - \nabla f^{\ell-1}, \nabla f^\ell - \nabla f^{\ell-1} \rangle}.$$

We now describe the proposed algorithm.

Iterative ADMM

Initialize $\mathbf{y}^0, Z^0, \Lambda^0$. Select $\rho > 0$.

1. Set $\mathbf{y}^{(k+1,0)} = \mathbf{y}^k$, and $\mathbf{y}^{(k+1,-1)} = \mathbf{y}^k$ (if applicable).

   While a termination criterion is not satisfied, set

$$\mathbf{y}^{(k+1,\ell+1)} = \mathbf{y}^{(k+1,\ell)} - \alpha^\ell \left(D^\ell\right)^{-1} \nabla_{\mathbf{y}} \mathcal{L}_\rho \left(\mathbf{y}^{k+1,\ell}\right), \tag{3.9}$$

   where α^ℓ and d^ℓ are selected in one of the many ways described above.

   $\mathbf{y}^{k+1}$ is then given by the value of $\mathbf{y}^{(k+1,\ell+1)}$ at termination.

2. Compute the SVD

$$W\Sigma V^T = \mathbf{H}\left(\mathbf{y}^{k+1}\right) U^\perp + (\rho)^{-1} \Lambda^k, \quad \text{and set}$$
$$Z^{k+1} = W \max\left\{\Sigma - \frac{\gamma}{\rho} I, 0\right\} V^T.$$

3. $\Lambda^{k+1} = \Lambda^k + \rho\left(\mathbf{H}\left(\mathbf{y}^{k+1}\right) U^\perp - Z^{k+1}\right)$.

If a termination criterion is not satisfied, go to 1.

We use a similar termination criterion as in section 3.2.1. For the iterative minimzation step, we terminate when the norm of the gradient of $\mathcal{L}_\rho$ decreases below 10^{-3}. It is worth noting that in the case of the gradient descent direction with a constant step-size given by $\alpha = \frac{\rho}{\rho r+1}$, where $\rho = \frac{\gamma r}{2\sigma_{max}(\tilde{y})}$ [Fazel *et al.*, 2013], the iteration (3.9) is equivalent to step 2 of the Primal ADMM algorithm.

We now discuss the computational requirements of each of these algorithms. As mentioned in section 3.1, the interior-point method of [Liu and Vandenberghe, 2009] has a cost per iteration of $\mathcal{O}(p^3 r N^2 c)$, with $pN \geq c \geq pr$. The ADMM algorithm of [Liu *et al.*, 2013] involves an SVD of cost $\mathcal{O}(p^2 r^2 c)$ at each iteration, in addition to the cost, $\mathcal{O}(p^3 N^3)$, of solving a linear system of equations once at the start of the algorithm. For the *Primal ADMM* of [Fazel *et al.*, 2013], each iteration involves an SVD of cost $\mathcal{O}(p^2 r^2 c)$, and several matrix multiplications each of cost $\mathcal{O}(pNrc)$. For our proposed *Iterative ADMM* algorithm, using a gradient descent direction requires the computation of an SVD of cost $\mathcal{O}(p^2 r^2 c)$ and

several matrix multiplications each of cost $\mathcal{O}(pNrc)$, at every ADMM iteration. These are common to both, the BFGS and Newton methods as well. In addition, the BFGS method involves several matrix multiplications each of cost $\mathcal{O}(p^3N^3)$ at every ADMM iteration. The Newton method involves the additional cost, $\mathcal{O}(p^3N^3)$, of computing the inverse of the Hessian matrix once at the start of the algorithm, apart from the basic $\mathcal{O}(p^2r^2c)$ SVD and $\mathcal{O}(pNrc)$ matrix multiplications at every ADMM iteration..

Chapter 4

Results and Discussion

We now compare the performance of the different algorithms for solving the nuclear norm system identification problem, on the PCB-132 data, as well as on two benchmark problems from DaISy (Database for the Identification of Systems) [DeMoor *et al.*, 1997]: hair dryer (single input single output), and CD player arm (two inputs and outputs). We use the Matlab code provided by the authors of [Liu *et al.*, 2013], and the Python code of [Liu and Vandenberghe, 2009], which is compiled using Python 2.7.3 and cvxopt 1.1.3. All other codes are written in Matlab 8.2.0.348 (R2013b). All experiments are performed on a 3.40 GHz Intel Core i7-2600 desktop computer with 8 GB of memory.

Table 4.1 compares all the algorithms in terms of the number of iterations and the time taken to converge. For every dataset, a value of the regularization parameter γ is chosen corresponding to the smallest identification error for a range of values of γ, obtained from Iterative ADMM. The identification error is defined as,

$$e_I = 100 \times \frac{\| \hat{y} - \tilde{y} \|_F}{\| \tilde{y} \|_F}, \tag{4.1}$$

where $\hat{y} \in \mathbb{R}^{p \times N}$, is the output of the estimated state-space model, starting at the estimated initial state. The LTI System Identification Toolbox [Verhaegen and Verdult, 2007] is used for estimating the state-space system once the optimal output sequence is obtained by solving the optimization problem. We also define the validation error, similar to (4.1) as,

$$e_V = 100 \times \frac{\| \hat{y}_V - \tilde{y}_V \|_F}{\| \tilde{y}_V \|_F}, \tag{4.2}$$

Table 4.1: Computational results

Dataset		**PCB-132**	**Hair Dryer**	**CD Player arm**
N/r		300/30	250/30	200/30
γ		0.008	0.119	5.298
Method		**iter/cpu**	**iter/cpu**	**iter/cpu**
IP		8/9.17	9/6.67	10/8.33
PADMM		**14/0.11**	**50/0.25**	353/1.37
LHV ADMM		51/0.4	55/0.31	**200/0.67**
IADMM, Gradient Descent	Armijo	22/7.38	23/3.87	max/74.5
	Wolfe	22/0.77	23/0.66	max/63.5
	BB	**22/0.48**	**23/0.35**	**max/60.6**
IADMM, BFGS	Armijo	22/1.18	23/0.83	max/353
	Wolfe	22/2.66	23/1.39	max/196
	BB	22/1.05	23/0.94	max/602
IADMM, Newton	Armijo	22/1.79	23/1.06	max/62.7
	Wolfe	22/4.17	24/1.57	max/79
	BB	22/1.92	23/0.97	max/64.5

Here, IP stands for the interior-point method of [Liu and Vandenberghe, 2009], PADMM denotes the Primal ADMM of [Fazel *et al.*, 2013], LHV denotes the ADMM algorithm of [Liu *et al.*, 2013], and IADMM stands for the proposed Iterative ADMM algorithm, implemented using Armijo linesearch, Wolfe linesearch, and the Barzilai-Borwein (BB) step-size rules. The results are in the form, '**iter/cpu**,' where iter stands for the number of iterations (outer iterations in case of IADMM), and cpu denotes the CPU time, in seconds, taken to perform the optimization. The word "max" denotes the maximum number of iterations, 10,000. N denotes the number of samples used for identification, r, the number of block rows of the Hankel matrices, and γ is the regularization parameter.

where $\hat{y}_V, \tilde{y}_V \in \mathbb{R}^{p \times N_V}$, N_V being the number of samples used for validation, which is greater than, and includes N. The fastest algorithm for each dataset is highlighted in bold. We observe that, while the existing ADMM algorithms are very efficient, the proposed Iterative ADMM algorithm fares well in comparison. Notably, it outperforms the interior-point algorithm in terms of computation time, at least for single-input single-output (SISO) systems. We also note that Iterative ADMM works best with a gradient descent direction, followed by BFGS for SISO, and Newton's method for multi-input multi-output (MIMO) systems. The Barzilai-Borwein (BB) step-size rule looks to be the fastest to converge, with backtracking linesearch also performing well.

Table 4.2 shows some of the postprocessing results for Iterative ADMM, for the same set of experiments as table 4.1. Namely, it shows the estimated system order, n, and the identification and validation errors. The estimated system order is set to be the rank of[1] $H(y^*)U^\perp$ in (2.5), y^* being the optimal solution. Here, we also include identification results of the Matlab command `n4sid`, which is often used for system identification. This command implements the classical subspace identification algorithm discussed in section 2.1. This approach yields rather large values for the identification and validation errors, providing some perspective to the relevance and usefulness of the nuclear norm approach.

We also illustrate, in figures 4.1 and 4.2, the singular values of $H(y^*)U^\perp$ and estimated outputs for two of the examples in table 4.2. To further analyze the proposed Iterative ADMM algorithm, we compare in table 4.3, the number of inner iterations required for convergence of the iterative minimization step, over the first[2] outer (ADMM) iteration.

[1] Or the rank of $\mathbf{H}(\mathbf{y}^*)U^\perp$ in the case of Iterative ADMM. The two are equivalent.

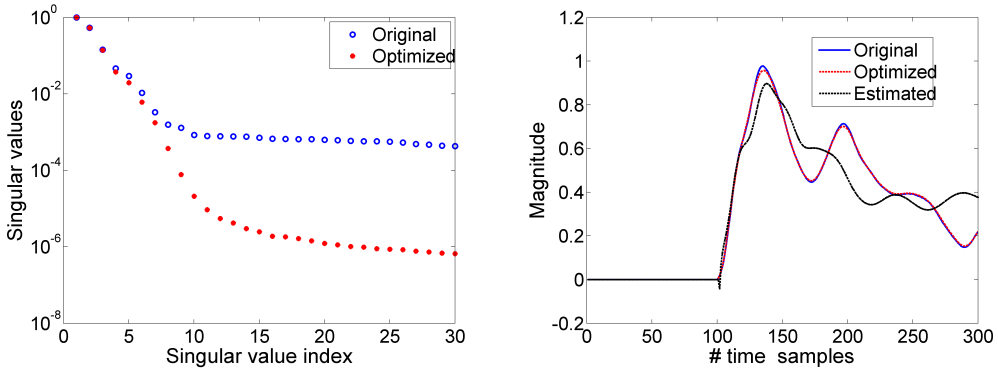
[2] Over subsequent outer iterations, the number of inner iterations required for convergence decreases.

Table 4.2: Estimated system order and identification/validation errors

Dataset	PCB-132 $n/e_I/e_V$	Hair Dryer $n/e_I/e_V$	CD Player arm $n/e_I/e_V$
IADMM	8/22.98/68.98 (GD, Wolfe)	7/1.56/2 (Newton, BB)	2/19.04/19.45 (BFGS, Armijo)
n4sid$_d$	3/30.01/86.87	3/1.53/2.11	2/104/102

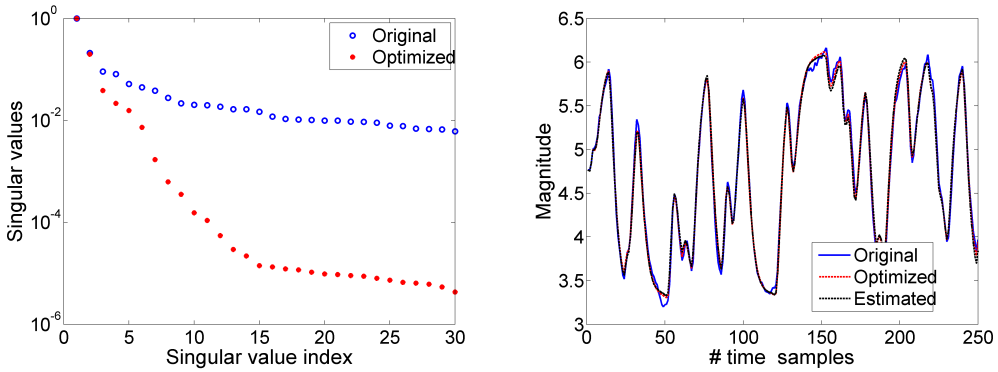
Here, n is the estimated system order, and e_I and e_V are the identification and validation errors defined in 4.1 and 4.2 respectively. For Iterative ADMM, the descent direction and step-size rule used are shown in parenthesis. 'n4sid' denotes the identification results of the Matlab command of the same name. The subscript d stands for the default option, where the 'best' system order is selected.

Figure 4.1: Singular values of $H(y^*)U^\perp$ and estimated outputs for PCB-132



Results for the IADMM algorithm, using gradient descent + Wolfe linesearch. In the left figure, 'Original' stands for the singular values of $H(\tilde{y})U^\perp$, while 'Optimized' denotes the singular values of $H(y^*)U^\perp$. The right figure shows the true data, $\tilde{y}$, the optimal output y^* and the output, $\hat{y}$, of the estimated system.

Figure 4.2: Singular values of $H(y^*)U^\perp$ and estimated outputs for 'Hair dryer'



Results for the IADMM algorithm, using Newton + BB step-size. In the left figure, 'Original' stands for the singular values of $H(\tilde{y})U^\perp$, while 'Optimized' denotes the singular values of $H(y^*)U^\perp$. The right figure shows the true data, $\tilde{y}$, the optimal output y^* and the output, $\hat{y}$, of the estimated system.

Table 4.3: Comparison of descent directions for the iterative minimization step

Dataset	PCB-132 # inner iterations	Hair Dryer # inner iterations	CD Player arm # inner iterations
Gradient Descent	216	208	153
BFGS	14	12	10
Newton	14	8	12

Here we compare the number of inner iterations required for convergence of the iterative minimization step of the proposed algorithm, over the first outer iteration. Only backtracking linesearch is considered here, for step-size selection.

Chapter 5

Conclusions

This thesis looked closely at the problem of nuclear norm system identification. Unlike in the classical subspace method for system identification, the singular values of the *optimal* Hankel-type matrix arising from the nuclear norm optimization step, provide a more clear-cut criterion for model order selection. It is therefore, a very attractive feature of this approach, that it preserves this Hankel structure during the optimization. This approach also allows the user to trade off the model order with the fitting error, as desired.

We developed an *Iterative ADMM* algorithm for solving this nuclear norm system identification problem. This algorithm involves an iterative minimization step, which can employ a gradient descent direction, or Newton or quasi-Newton directions. From the computational results on the PCB-132 sensor dataset, and two additional benchmark datasets from the DaISy database, we conclude that our proposed algorithm, although slightly less efficient than the ADMM algorithms of [Fazel *et al.*, 2013; Liu *et al.*, 2013], outperforms the interior-point method proposed in [Liu and Vandenberghe, 2009], for SISO systems, in terms of computation time. Also, since the computation time and the number of iterations for the interior-point method depends only on the size of the problem, and does not change with the value of the regularization parameter, a comparison of the cost per iteration leads us to conclude that our approach is more suitable for large scale system identification problems, particularly if a modest accuracy is acceptable.

Among the different search directions considered, the gradient descent direction works best, followed by BFGS for SISO, and Newton for MIMO systems. For step-size selection,

the Barzilai-Borwein rule seems to be the fastest to converge

For the particular problem of the PCB-132 sensor, we think that a lot more can be said about the system if it is excited using a richer class of inputs, such as a sinusoidal input.

Bibliography

A. Abney, C. Ward, D. Berridge, R. Greenwood, and S. Schneider. Hypersonic boundary-layer transition experiments in the boeing/afosr mach-6 quiet tunnel. In *Proceedings of 51st AIAA Aerospace Sciences Meeting*, 2013.

J. Barzilai and J. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 1988.

D. Berridge and S. Schneider. Calibration of pcb-132 sensors in a shock tube. 2012.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2011.

E. J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 2010.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009.

E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 2005.

E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 2006.

S. S. Chen and D. Donoho. Basis pursuit. In *Proceedings of the 28th Asilomar Conference*, 1994.

B. DeMoor, P. DeGersem, B. DeSchutter, and W. Favoreel. Daisy: A database for identification of systems. *Journal A 38*, 1997.

N. Dhingra, F. Lin, M. Fardad, and M. R. Jovanović. On identifying sparse representations of consensus networks. In *Preprints of the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 305–310, Santa Barbara, CA, 2012.

N. K. Dhingra, M. R. Jovanović, and Z. Q. Luo. Optimal sensor and actuator selection for large-scale systems. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, Los Angeles, CA, 2014. to appear.

F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo. Sparse and optimal wide-area damping control in power networks. In *Proceedings of the 2013 American Control Conference*, pages 4295–4300, Washington, DC, 2013.

F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo. Sparsity-promoting optimal wide-area control of power networks. *IEEE Trans. Power Syst.*, 2014. doi:10.1109/TPWRS.2014.2304465; also arXiv:1307.4342.

M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the IEEE American Control Conference*, 2001.

M. Fazel, T. K. Pong, D. Sun, and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 2013.

M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

P. A. Forero, A. Cano, and G. B. Giannakis. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11:1663–1707, 2010.

T. Goldstein and S. Osher. The split Bregman method for ℓ_1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

A. Hassibi, J. How, and S. Boyd. Low authority controller design via convex optimization. *AIAA J. Guidance, Control, and Dynamics*, 1999.

M. R. Jovanović and F. Lin. Sparse quadratic regulator. In *Proceedings of the 12th European Control Conference*, pages 1047–1052, Zurich, Switzerland, 2013.

F. Lin, M. Fardad, and M. R. Jovanović. Identification of sparse communication graphs in consensus networks. In *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing*, pages 85–89, Monticello, IL, 2012.

F. Lin, M. Fardad, and M. R. Jovanović. Sparse feedback synthesis via the alternating direction method of multipliers. In *Proceedings of the 2012 American Control Conference*, pages 4765–4770, Montréal, Canada, 2012.

F. Lin, M. Fardad, and M. R. Jovanović. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Transactions in Automation and Control*, 2013.

F. Lin, M. Fardad, and M. R. Jovanović. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Trans. Automat. Control*, 58(9):2426–2431, September 2013.

F. Lin, M. R. Jovanović, and T. T. Georgiou. An ADMM algorithm for matrix completion of partially known state covariances. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, 2013. to appear.

F. Lin, M. Fardad, and M. R. Jovanović. Algorithms for leader selection in stochastically forced consensus networks. *IEEE Trans. Automat. Control*, 59(7):1789–1802, July 2014.

Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 2009.

Z. Liu, A. Hansson, and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. *Systems & Control Letters*, 2013.

L. Ljung. *System Identification - Theory for the User*. Prentice-Hall, 1999.

M. Mesbahi. On the semidefinite programming solution of the least order dynamic output feedback synthesis. In *Proceedings of the IEEE American Control Conference*, 1999.

K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *Proceedings of the IEEE American Control Conference*, 2010.

J. Nocedal and S. Wright. *Numerical Optimization (2nd edition)*. Springer, 2006.

T. E. Pare. *Analysis and control of nonlinear systems*. PhD thesis, Stanford University, 2000.

B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 2007.

K. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 2009.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 1996.

M. Verhaegen and V. Verdult. *Filtering and system identification: a least squares approach*. Cambridge University Press, 2007.

M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 1994.

X. Wu and M. R. Jovanović. Sparsity-promoting optimal control of consensus and synchronization networks. In *Proceedings of the IEEE American Control Conference*, 2014.

X. Wu and M. R. Jovanović. Sparsity-promoting optimal control of consensus and synchronization networks. In *Proceedings of the 2014 American Control Conference*, pages 2948–2953, Portland, OR, 2014.

X. Wu, F. Dörfler, and M. R. Jovanović. Analysis and design trade-offs for power network inter-area oscillations. In *Proceedings of the 21st International Symposium on Mathematical Theory of Network and Systems*, pages 657–663, Groningen, The Netherlands, 2014.

A. Zare, M. R. Jovanović, and T. T. Georgiou. Completion of partially known turbulent flow statistics. In *Proceedings of the 2014 American Control Conference*, pages 1680–1685, Portland, OR, 2014.

H. Zhu, G. B. Giannakis, and A. Cano. Distributed in-network channel decoding. *IEEE Trans. Signal Process.*, 57(10):3970–3983, 2009.

D. M. Zoltowski, N. K. Dhingra, F. Lin, and M. R. Jovanović. Sparsity-promoting optimal control of spatially-invariant systems. In *Proceedings of the 2014 American Control Conference*, pages 1261–1266, Portland, OR, 2014.