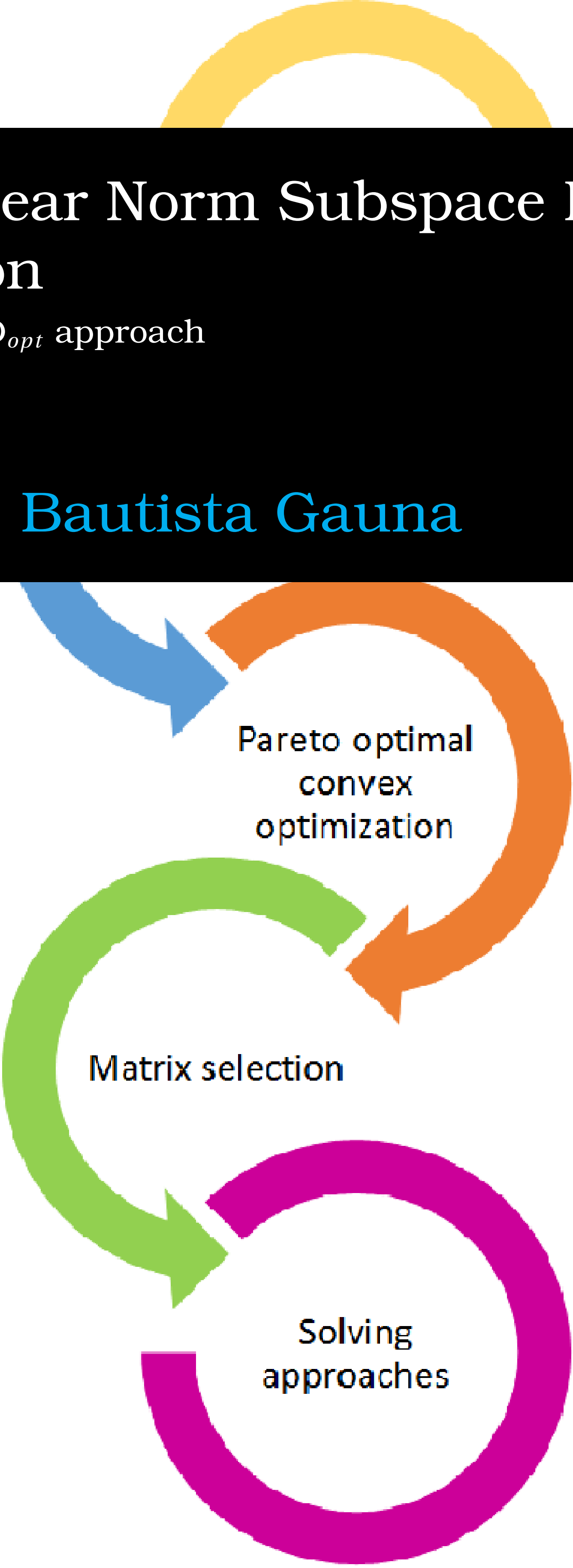
Nuclear Norm Subspace Identification

a PBSID$_{opt}$ approach

J. E. Bautista Gauna

Literature Survey



Delft University of Technology

Delft Center for Systems and Control

Nuclear Norm Subspace Identification

a PBSID$_{opt}$ approach

Literature Survey

J. E. Bautista Gauna

September 14, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

New system identification methods are developing constantly to come up with solutions that can take into account all of the factors that real-time systems have. These factors affect the analysis of the system's behavior, particularly noise, non-linearities, system's complexity, time varying changes, among others.

This thesis is concerned with system identification methods related to a nuclear norm pareto optimization. These algorithms try to join 2 families of system identification methods, the PEM and SID, this with the purpose of trying to find a balance or trade-off between the complexity of the system identified and the goodness of the fitting criterion.

The nuclear norm heuristics are used widely as a convex approximation for the RMP. In this work it will be used for system identification purposes; were a new low rank matrix and fitting criterion formulation is constructed inspired by the PBSID$_{opt}$ algorithm. This algorithm will rename and stack data to construct a matrix that thanks to the properties assumed in the system it can be defined as a upper triangular low rank matrix. The objective is to see if this method brings any benefits compared to the already existent low rank matrix mechanisms that are inspired by structural analysis or the instrumental variables and/or projections methods.

A step further was taken to provide the framework for this method named N2PBSID algorithm by providing the steps necessary for the implementation of this optimization problem on MATLAB.

Lastly, several experiments are performed and a number of conclusions are drawn from them. Making an assessment on the effectiveness of the algorithm, the feasibility of its performance based on the difficulties encountered and final thoughts and recommendations.

Table of Contents

Acknowledgments

This thesis is the achievement that marks the end of my studies here in TUDelft as a Master student, which oddly enough took me a lot of bittersweet situations to complete. From the initial excitement of coming to a new stimulating country, to what I now call the 3 important and essential S in every foreign or international student: struggles, stress and self-discovery from the cultural shock, and ending with the inherent and inevitable rich experiences, I can finally put a end to this play. Because yes, this part of my life felt a lot more like a dramatic Greek play than a standard enriched educational journey.

That's why here I want to thank everyone that was involve in these impromptu scenes:

The characters. Everyone included in the play. Starting with me for not giving up on finishing all the bloody acts, my friends by being there for me when I needed a shoulder to cry or a reason to laugh, my roommate Martine for giving me a refreshing perspective of life and a cozy place that I can call home and my friend Sander that was the sage, patient and secret second protagonist in this whole emotional jungle. Without them carrying me repeatedly half death from one stage to the other, opening and closing curtains and placing the correct props everywhere I could not have finish this.

The plot. Everyone that develop my journey. Everything in TUDelft, all of my professors and the way the classes were developed, the complicated homeworks, the terrifing exams, the not-ending group projects, coupled with the horrible Aula food and the culture differences made this play really interesting for everyone watching.

Special thanks to Prof. Jan-Willem van Wingerden who acted as the wise philosopher of the story, giving me a really clear way to finish my storyline, giving me hints (sometimes more obvious that others) and encouragements to finish, and to Sachin Navalkar for his time and infinite patience.

The audience. Everyone that was watching the play expecting me to succeed. My parents and my grandma, getting really nervous that it took me more acts than necessary to finish the story, the rest of my family to be proud of me for seeking this professional success, and all the friends that are far away from me whom cheer me up from the other side of the globe.

Without the help of every and each person mentioned I am sure this dramatic comedy would have

ended more as a tragic chaos.

Delft, University of Technology J. E. Bautista Gauna
September 14, 2016

Chapter 1

Introduction

1-1 Introduction to System Identification

System identification is about data-based modeling; it is the science of building mathematical models of dynamic systems from observed input-output data. The field uses not only statistical methods for model creation, but also includes model reduction.

When thinking of modeling, we indeed generally think of first-principle modeling and not data-based modeling, that is the modeling using the laws of physics (Newton laws, mass conservation, Kirchhoff's circuit laws, etc.). However, data-based modeling is often as important because first-principle modeling has several disadvantages,

- model contains many unknown (physical) parameters ⇒ high uncertainty
- model generally more complex than with system identification
- missing actuator/sensor dynamics and phenomena can be forgotten
- sometimes the model is impossible to determine
- there is no way to determine a disturbance model

Two widely used methods, or families of methods for system identification are the so called Prediction Error Methods (PEMs) [1] and Subspace IDentification (SID) methods [2].

The core of PEMs is to parametrize the predictor or observer to generate an estimate of the output, and then formulate an optimization problem to minimize a weighted cost function defined in terms of the difference between the measured output and the observer predicted output. This cost function is for the finite data length case generally a summation of the trace of the covariance matrix of the prediction error.

Though the PEM framework provides a vast amount of insights in studying and analyzing the estimated predictor, its main drawback is the non-convexity for general multivariable state space models in innovation form [2]. The lack of convexity can result in the optimization method getting stuck in a local minimum, and thereby complicating the analysis of the numerical results, such as e.g. difficulty to distinguish between a bad model estimate due to a local minimum or due to a bad model parametrization.

SID methods on the contrary derive approximate models rather than models that are "optimal" with respect to a chosen cost function. All existing SID methods aim to derive a low rank matrix from which key subspaces, hence the name subspace identification, are derived. The low rank approximation is in general done using a Singular Value Decomposition (SVD).

The SID methods that had been developed turned out to be very useful in complementing PEM in a way that SID provided the needed initial estimates for PEM to start up the non-covex parameter optimization.

Despite this complementarity between the two families of system identification methods, algorithms that integrate both are numbered. These algorithms aim to merge the subspace determination step of SID with the goodness of fit optimization step of PEM to identifying state space models in innovation form. This integration step enables a trade-off between the complexity of the identified state space models, as expressed e.g. by a rank constraint in the subspace determination step, with the accuracy of fitting as expressed by a goodness of fit criterion.

The lack of algorithms that integrate both methods motivates a quest for an algorithm which not only merge both worlds but have the same or better results that conventional existing PEM methods. The aim of this report is to continue this quest.

1-2 Motivation

The control community is always searching for a better method for system identification analysis, and providing an algorithm than can bring better results that the ones previously obtained is always a pursuing objective by engineers. So, bringing onto the table an algorithm that can consolidate both families, PEM and SID, while obtaining promising results is of interest. That is why this new approach is analyzed, implemented and tested; to see if a multi-criteria convex relaxation using the theory of predictor-based subspace identification (PBSID$_{opt}$) can deliver any new benefits.

1-3 Goals of this thesis

The objective of this thesis is to provide a new formulation on the low rank matrix of the existing nuclear norm approach inspired by the $PBSID_{opt}$ algorithm. This approach will focus on finding Markov parameters instead of the column space of the extended observability matrix.

The first step to achieve this is to develop and implement these $PBSID_{opt}$ approach using the Alternating Direction Method of Multipliers (ADMM) algorithm. The second step is to carry out a study on the performance of the subspace identification algorithm with the new low rank formulation and compare it to several existing PEM and SID algorithms.

In this thesis the following questions are answered:

- Does this new approach provide satisfactory results for system identification?
- Which benefits does the proposed method provide over regular existing algorithms and what conditions for these benefits to occur?

1-4 Outline

This work is organized as follows. In Chapter 2 an overview is presented of the principles of subspace identification. This will give the reader the framework necessary to follow the next chapters. Chapter 3 presents the new $PBSID_{opt}$ approach as a low rank matrix for the nuclear norm algorithm. Additionally the same $PBSID_{opt}$ framework will be used with a reweighted iteration minimization to try to find new benefits out of this analysis. The next chapter, Chapter 4, shows the results of the implemented algorithm in several data cases, including a comparison of the simulations of other methods. Finally, the conclusions of this thesis are described in Chapter 5 with additional recommendations for future research.

Chapter 2

Nucelar Norm Subspace Identification

This chapter provides an overview of the principles of system identification were several SID methods will be presented; this will give the reader the framework necessary to follow the next chapters.

2-1 Problem formulation

System identification methods for Linear Time-Invariant (LTI) systems estimate linear state-space models with process and measurement noise.

These models for LTI systems are generally given in a so-called innovation form [2],

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ke(k) \\ y(k) &= Cx(k) + Du(k) + e(k) \end{aligned} \tag{2-1}$$

with $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$, $u(k) \in \mathbb{R}^m$ and $e(k)$ a zero-mean white noise ergodic sequence.

The problem is to determine approximate system matrices ($\hat{A}_T$, $\hat{B}_T$, $\hat{C}_T$, $\hat{D}_T$, $\hat{K}_T$) that define the $\hat{n}$-th order observer of "low" complexity:

$$\begin{aligned} \hat{x}_T(k+1) &= \hat{A}_T\hat{x}_T(k) + \hat{B}_T u(k) + \hat{K}_T\big(y(k) - \hat{C}_T\hat{x}_T(k) - \hat{D}_T u(k)\big) \\ \hat{y}(k) &= \hat{C}_T\hat{x}_T(k) + \hat{D}_T u(k) \end{aligned} \tag{2-2}$$

such that the approximated output $\hat{y}(k)$ is "close" to the measured output $y(k)$ of the validation pair $\{u(k), y(k)\}_{k=1}^N$ as expressed by a small value of the cost function

$$\frac{1}{N}\sum_{k=1}^{N} \| y(k) - \hat{y}(k) \|_2^2 \tag{2-3}$$

The first step, depicted in Figure 2-1, is given by the available data, that is the input-ouput (i/o) data batches $\{u(k), y(k)\}_{k=1}^N$, with $N > n$. The data is assumed to be retrieved from an identification

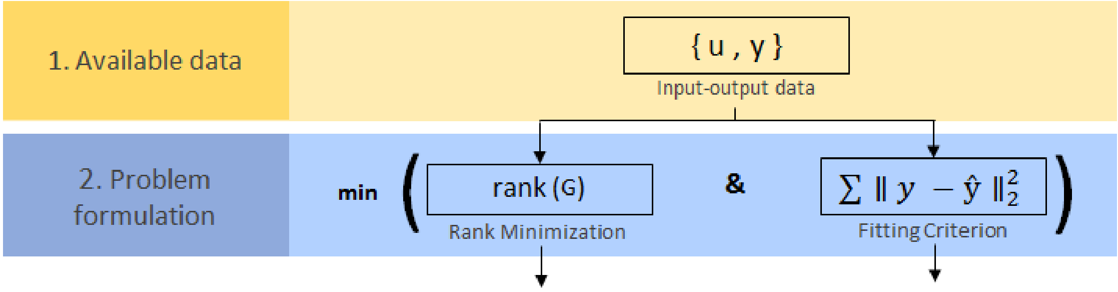


Figure 2-1: First steps of the system identification process by multi-criteria optimization

experiment with a system belonging to the class of LTI systems as represented by Eq. (2-1). The second layer represents the union between the PEM and the SID methods, this frame is presented in the next problem formulation:

$$\text{minimize} \;\; \text{rank}\,(G) \quad \text{and} \quad \text{minimize} \;\; \frac{1}{N}\sum_{k\in T} \| y(k) - \hat{y}(k) \|_2^2 \tag{2-4}$$

This problem seeks the Pareto optimal solution with respect to two cost functions.

2-2 Pareto optimal convex optimization

The optimization problem presented earlier seeks for the Pareto optimal solution with respect to the two cost functions rank$\left(G\right)$ and $\frac{1}{N}\sum \| y(k) - \hat{y}(k) \|_2^2$. This optimization is however NP (Non-deterministic Polynomial-time) hard, making this a not tractable problem, which means that it cannot be solved in polynomial time.

However, thanks to the discovery of the heuristics behind the rank minimization problem in Eq. (2-4) [3] the first term of the optimization problem can be rewritten in two different convex ways that can be seen on the first row in Figure 2-2, were each one seeks to minimize the trace of the matrix in question.

The first of the approaches is the nuclear norm minimization (depicted as the letter B in the diagram). This approach is used in several research papers, mainly [4], [5], [6], [7], and is expressed as

$$\text{minimize} \quad \| G \|_* \tag{2-5}$$

were the nuclear norm of the matrix X is denoted by $\| X \|_*$ and represent the sum of the singular values of the matrix. Hence, the first part of this problem formulation adds relevant structural information available about the unknown matrices in the data equation in SID that relates the Hankel matrices of input and/or output measurements as convex constraints.

The second approach is named reweigthed nuclear norm by [8] (depicted as the letter A in the diagram). In this approach the rank minimization problem mentioned above is replaced by the

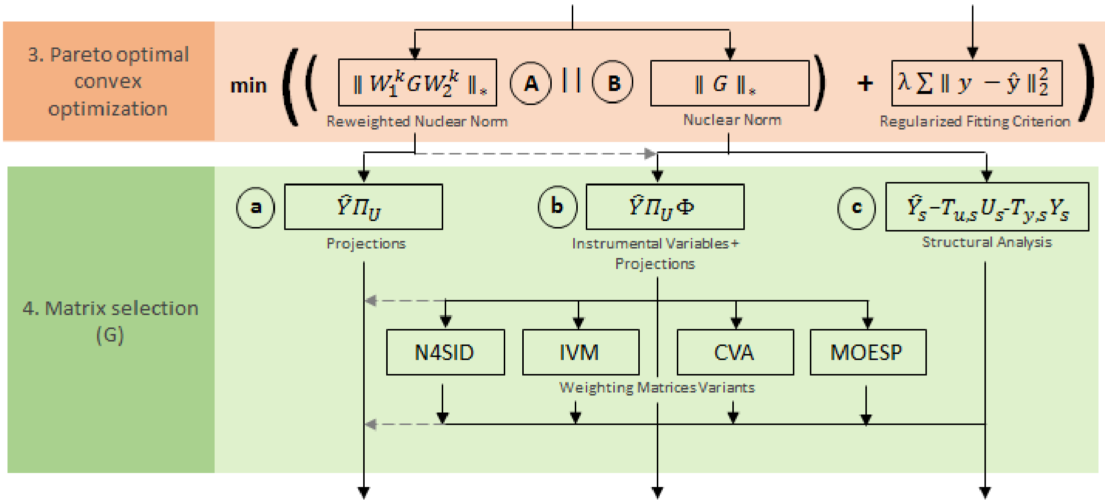

3. Pareto optimal convex optimization
min
$\| W_1^k G W_2^k \|_*$
A
||
B
$\| G \|_*$
+
$\lambda \sum \| y - \hat{y} \|_2^2$
Reweighted Nuclear Norm
Nuclear Norm
Regularized Fitting Criterion
4. Matrix selection (G)
a
$\hat{Y} \Pi_U$
Projections
b
$\hat{Y} \Pi_U \Phi$
Instrumental Variables + Projections
c
$\hat{Y}_s - T_{u,s} U_s - T_{y,s} Y_s$
Structural Analysis
N4SID
IVM
CVA
MOESP
Weighting Matrices Variants

Figure 2-2: Next steps on the diagram: different forms of convex regulatization and low rank matrix selection

reweighted trace heuristics by a surrogate concave $log\text{-}det$ function, which by local iterative linearization (trace heuristics) can help us find a solution to the problem.

Thanks to this trace heuristics linearization we can transform the RMP into a characterization of the nuclear norm expressed as

$$\text{minimize} \quad \| W_1^k G W_2^k \|_* \tag{2-6}$$

Finally, thanks to the addition of the regularization parameter $\lambda \in [0, \infty)$ in the second cost function it is possible to obtain all Pareto optimal solutions via a convex optimization where both fitting and accuracy can be treated in the same convex optimization problem expressed as

$$\text{minimize} \quad \| Z \|_* + \frac{\lambda}{N} \sum_{k \in T} \| y(k) - \hat{y}(k) \|_2^2 \tag{2-7}$$

where Z represents any of the nuclear norm approaches for the convex optimization.

Is important to remark that this regularization parameter λ is the bridge that joins and creates a trade-off between the cost criterion (PEM) and the complexity of the estimated system (SID). Choosing this parameter, however, is not trivial. All authors suggest to create an interval of the parameter λ denoted by $\Lambda = [\lambda_{min}, \lambda_{max}]$ where a grid is made to test different values until the minimum possible value of both cost functions is discovered.

2-3 Low Rank Matrix Selection

The forth step in this process is when the authors choose a low rank matrix G, this matrix is mentioned in Eq. (2-5) and Eq. (2-6), to represent the range of the observability matrix of the real system.

To construct the low rank matrix several authors have taken different approaches, seen clearly in the second segment of Figure 2-2. The main approaches mentioned are:

- The conventional SID methods, which require projection and/or instrumental variables to isolate the range of the observability matrix.
- The N2SID structural analysis approach, which uses structural properties of the $data\ equation$ which give us the range of the observability matrix without the loss of information that the conventional SID methods contain thanks to key structural analysis.

The purpose of this thesis is to expose a new low rank matrix G which applies the PBSID$_{opt}$ theory to create not only a new formulation of the first element of the pareto optimal convex optimization (the reweighted or the standard nuclear norm) but also rewrites the second part (the regularized fitting criterion) with the Hankel matrix of the PBSID$_{opt}$ state space output equation. This approach will be described and explained in the next chapter.

2-4 Solving approaches

In this section we show several methods that can be used to solve the optimization problem presented in the last section and explain the steps that these algorithms require for its computation.

There are 3 general methods depicted in Figure 2-3 that are the ones used to solve this kind of convex optimization problem, these are the Semi-Definite Programming (SDP), a more recent one named Alternating Direction Method of Multipliers (ADMM) and an Iterative Minimization method. Each one will be explained briefly.

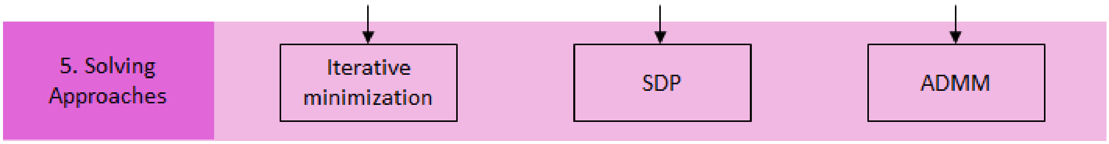


Figure 2-3: Final step of the system identification process

2-5 Semi-Definite Programming (SDP)

A linear programming problem is one in which we wish to maximize or minimize a linear objective function of real variables over a polytope. In semidefinite programming, we wish to maximize or minimize a linear objective function of real-valued vectors (and are allowed to take the dot product of vectors). In other words, nonnegativity constraints on real variables in linear programming (LP) are replaced by semidefiniteness constraints on matrix variables in SDP [9].

The main advantage of this approach is the formulation. Almost any problem can be formulated as an SDP problem and can be easily modified. For example, if you want to add a constraint, the user just needs to input a new constraint into the problem and run the algorithm. The problem or disadvantage relies in the fact that this is not the most efficient algorithm, thus, it takes a larger computational time than the algorithm that will be presented below.

2-6 ADMM algorithm

The ADMM is a relatively new method of optimization which is of growing interest for several reasons, mainly because many large-scale and distributed convex optimization problems can be rewritten or approximated as an ADMM implementation. What this method achieves, is to create an unconstrained problem by applying the Lagrangian method to the constrained problem with an additional penalty term (the augmentation).

The advantage of this algorithm is the fact that it takes any convex optimization problem and breaks it into smaller pieces for an easier and faster solution. So, compared to SDP it takes less computation time, but it has the difficulty that if the optimization problem is changed (the addition of constraints for example) the reformulation in the algorithm needs to be changed.

2-7 Iterative minimization

The nuclear norm minimization has a connection to the terms /vector sparsity/ and /ℓ_1-norm/. A variation of this heuristic terms helps to reduce the rank minimization problem further. This is achieved by using a weighted objective based on using a nonconvex surrogate function for the rank and solving the resulting problem locally via a sequence of convex problems.

This algorithm is easily implemented, but it has only been mentioned when dealing with rank minimization problems, since the heuristic behind it leads to the log-det surrogate function, which is the one that can be solve locally with linearization. It remains uncertain if this algorithm can be used in other convex optimization problems. Then again, the addition of changes (constraints) remains as well uncertain, since it was not found in literature other forms of iterative linearization other than the standard rank minimizaton problem. Despite that, if constraints can be added the reformulation of the problem is needed, thus, making it not suitable for modifications.

On the other hand, as presented in Section 2-2, the reweighted nuclear norm solved with this algorithm seems to converge into lower order models than the standard nuclear norm.

2-8 Conclusions

Here it concludes the overview of the system identification process. The approach taken to complete the diagram in Figure 2-4 was mainly based on the new research in [8], [5] and [4] for the reweighted and standard nuclear norm approach to form a pareto optimal convex optimization.

The process and different approaches of system identification by an PEM and SID combination to estimate the matrices that best describe the systems dynamics with a trade-off of the complexity of the system is completely depicted in Figure 2-4.

As explained above, each option in the process, like the rank minimization heuristic options (standard and reweighted nuclear norm), the low rank matrix selection (instrumental variables and/or projections and structural analysis) and the solving algorithms (ADMM, SDP, iterative weighted minimization) not only present advantages or disadvantages according to the problem that is presented (mainly the formulation of the problem and the amount of data available), but also achieves different results in the form of fitting, rank minimization or singular value gap spacing. and computational time.

Knowing the basics of system identification and how the PEM and SID families can be used together to form this new convex pareto optimization, it is now possible to introduce the new approach on the matrix selection based on the PBSID$_{opt}$ method. In Chapter 3 a new branch of Figure 2-4 will be discussed. This change represents new challenges in the programming and implementation of the algorithm and new opportunities in having advantages over other conventional system identification algorithms.

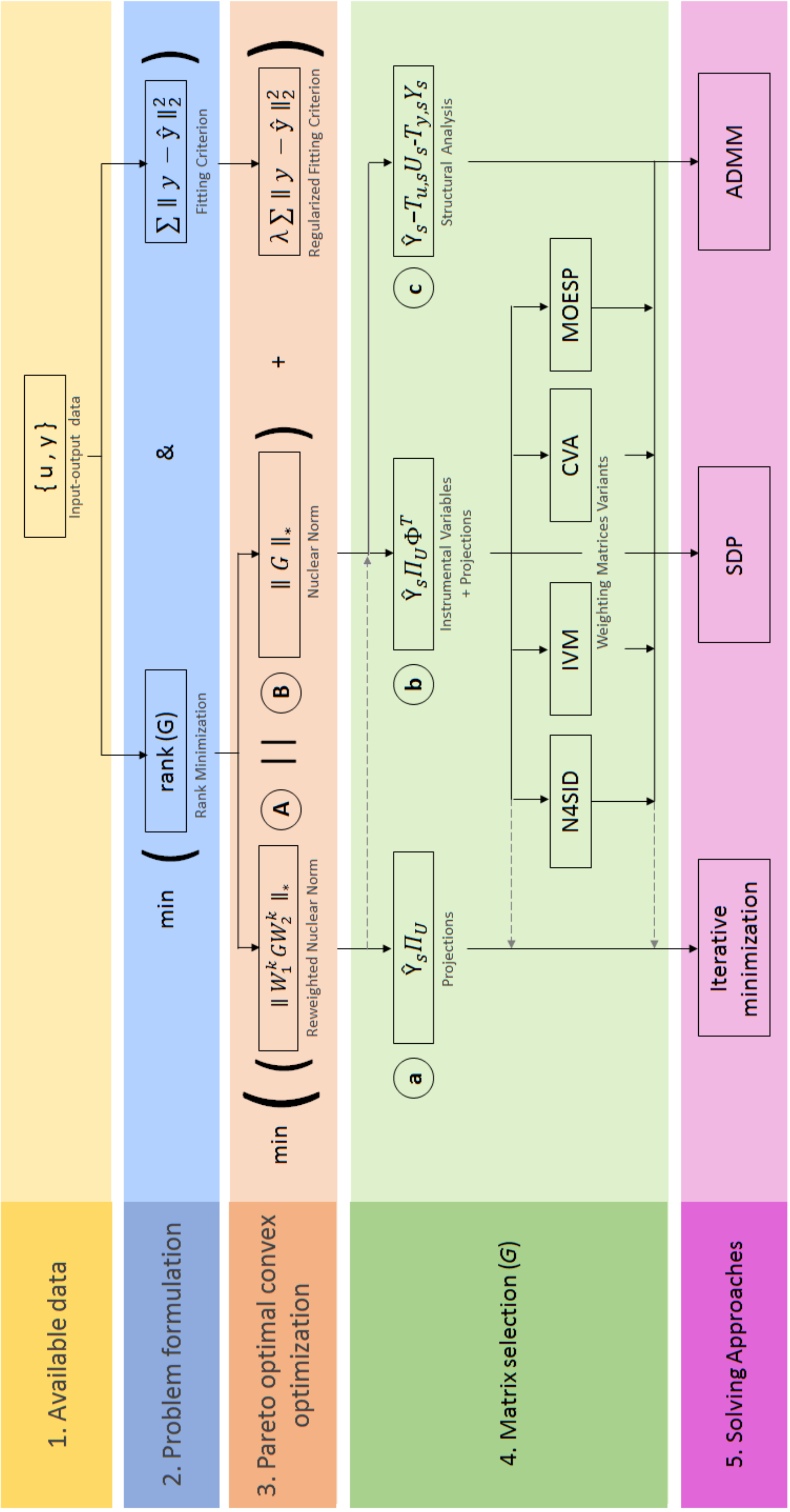


Figure 2-4: Standard flow diagram

Chapter 3

Nuclar Norm PBSID$_{opt}$ form

As explained before, in this chapter a new formulation of the pareto convex optimization will be introduced on the system identification process with the standard and reweighted nuclear norm approach. These new branches in the process will be shown and explained below.

3-1 PBSID$_{opt}$ formulation proposal

What it will be proposed in this thesis is a new way to approach step 3 and 4 of the diagram showed before by including new branches, this will be achieved by establishing a new way to formulate the low rank matrix of the nuclear norm relaxation and a new way to formulate the output estimate present in the second term, which is a cost function.

This new way of reformulate these 2 parameters is inspired by the PBSID$_{opt}$ algorithm for SID ([10]). First, an overview of the algorithm will be given and then it will be explained how this formulation is applied in a pareto convex optimization case.

3-1-1 PBSID$_{opt}$ method

In predictor-based subspace identification (PBSID$_{opt}$), a predictor for the state sequence is constructed, this is done by depicting our state space model in the so-called innovation form in Eq. (2-1) by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ke_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned} \tag{3-1}$$

where the ZMWN e_k can be eliminated from the first equation by substitution, yielding a system description in the one-step-ahead predictor form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + K(y_k - Cx_k - Du_k) = \tilde{A}x_k + \tilde{B}u_k + Ky_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned} \tag{3-2}$$

where $\tilde{A} = A - KC$ and $\tilde{B} = B - KC$.

The state and output equation derived from this notation at any further time p will be given by

$$x_{k+p} = \tilde{A}^p x_k + \begin{bmatrix} \tilde{A}^{p-1}\tilde{B} & \tilde{A}^{p-2}\tilde{B} & \dots & \tilde{B} \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+p-1} \end{bmatrix} + \begin{bmatrix} \tilde{A}^{p-1}\tilde{K} & \tilde{A}^{p-2}\tilde{K} & \dots & \tilde{K} \end{bmatrix} \begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+p-1} \end{bmatrix}$$

$$y_{k+p} = C\tilde{A}^p x_k + C\begin{bmatrix} \tilde{A}^{p-1}\tilde{B} & \tilde{A}^{p}\tilde{B} & \dots & \tilde{B} \end{bmatrix} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+p-1} \end{bmatrix} + C\begin{bmatrix} \tilde{A}^{p-1}\tilde{K} & \tilde{A}^{p}\tilde{K} & \dots & \tilde{K} \end{bmatrix} \begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+p-1} \end{bmatrix} + Du_{k+p} + e_{k+p}$$

By the assumption that $\tilde{A}$ is stable and has all its eigenvalues inside the open unit circle, the term $\tilde{A}^p$ can be made arbitrarily small by choosing p sufficiently large. For that reason the first term on the right-hand side of both state and output equations can be neglected.

Now two stacked variables will be introduced, this will help us to reformulate the state and output equations in a more compact manner.

$$z_k = \begin{bmatrix} u_k \\ y_k \end{bmatrix} = \begin{bmatrix} z_{k-p}^T & z_{k-p+1}^T & \cdots & z_{k-1}^T \end{bmatrix}^T \tag{3-3}$$

$$\hat{B} = \begin{bmatrix} \tilde{B} & K \end{bmatrix} \tag{3-4}$$

Describing our state space model as

$$x_{k+p} = \begin{bmatrix} \tilde{A}^{p-1}\hat{B} & \tilde{A}^{p-2}\hat{B} & \dots & \hat{B} \end{bmatrix} \begin{bmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{k+p-1} \end{bmatrix}$$

$$y_{k+p} = C\begin{bmatrix} \tilde{A}^{p-1}\hat{B} & \tilde{A}^{p-2}\hat{B} & \dots & \hat{B} \end{bmatrix} \begin{bmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{k+p-1} \end{bmatrix} + Du_{k+p} + e_{k+p}$$

Assuming that there is no-sample delay in the loop and that $\mathcal{K} = \begin{bmatrix} \tilde{A}^{p-1}\hat{B} & \tilde{A}^{p-2}\hat{B} & \dots & \hat{B} \end{bmatrix}$ and $z_{k+p} = \begin{bmatrix} z_k^T & z_{k+1}^T & \vdots & z_{k+p-1}^T \end{bmatrix}$, the equations that would describe our state space system at any further time p are

$$\begin{aligned} x_{k+p} &= \mathcal{K} z_{k+p} \\ y_{k+p} &= C\mathcal{K} z_{k+p} + e_{k+p} \end{aligned} \tag{3-5}$$

From here it can be noticed that the *data equation* is described as

$$Y_{k,p,N} = C\mathcal{K} Z_{k,p,N} + E_{k,p,N} \tag{3-6}$$

where the output vector is described as $Y_{k,p,N} = \begin{bmatrix} y_{p+1} & y_{p+2} & \cdots & y_N \end{bmatrix}$ and the Hankel matrix $Z_{k,p,N}$ is structured as

$$Z_{k,p,N} = \begin{bmatrix} z_k & z_{k+1} & z_{k+2} & \cdots & z_{N-p} \\ z_{k+1} & z_{k+2} & z_{k+3} & \cdots & z_{N-p+1} \\ z_{k+2} & z_{k+3} & z_{k+4} & \cdots & z_{N-p+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{k+p-1} & z_{k+p} & z_{k+p+1} & \cdots & z_N \end{bmatrix}$$

If we define a matrix $\tilde{\Gamma}^{(f)} = \begin{bmatrix} C^T & (C\tilde{A})^T & \dots & (C\tilde{A}^{f-1})^T \end{bmatrix}^T$ the observability-times-controllability matrix has the following structure (assuming $f = p$)

$$\tilde{\Gamma}^{(f)}\mathcal{K} = \begin{bmatrix} C\tilde{A}^{p-1}\hat{B} & C\tilde{A}^{p-2}\hat{B} & \dots & C\hat{B} \\ C\tilde{A}^{p}\hat{B} & C\tilde{A}^{p-1}\hat{B} & \dots & C\tilde{A}\hat{B} \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{p+f-2}\hat{B} & C\tilde{A}^{p+f-3}\hat{B} & \dots & C\tilde{A}^{f-1}\hat{B} \end{bmatrix} \tag{3-7}$$

Based on an earlier assumption, the next approximation can be introduced

$$\tilde{\Gamma}^{(f)}\mathcal{K} \approx \begin{bmatrix} C\tilde{A}^{p-1}\hat{B} & C\tilde{A}^{p-2}\hat{B} & \dots & C\hat{B} \\ 0 & C\tilde{A}^{p-1}\hat{B} & \dots & C\tilde{A}\hat{B} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & C\tilde{A}^{f-1}\hat{B} \end{bmatrix} \tag{3-8}$$

3-1-2 PBSID$_{opt}$ nuclear norm inspired formulation

In conventional subspace identification, the main aim of the methods is to derive a low rank matrix from which key subspaces are derived. In this new approach the classic extended observability matrix times state matrix $O_sX_{0,N}$ low rank matrix is substituted by the PBSID$_{opt}$ inspired matrix $\tilde{\Gamma}^{(f)}\mathcal{K}Z_{k,p,N}$, where $\tilde{\Gamma}^{(f)}$ takes the place of the observability matrix O_s and the state matrix $X_{0,N}$ derives from Eq. (3-5).

Since we know that the past window value p (which is assumed to be equal to f) is greater than the order of the system, we can be certain that $\tilde{\Gamma}^{(f)}\mathcal{K}Z_{k,p,N}$ is a low rank matrix.

$$\text{minimize} \quad \| \Gamma^{(f)} \mathcal{K} Z_{k,p,N} \|_* + \frac{\lambda}{N} \| Y_{p,N-p} - C \mathcal{K} Z_{k,p,N-p} \|_2^2 \tag{3-9}$$

This new approach is introduced in the system identification diagram by new branches, which can be seen highlighted in Figure 3-1.

This constitutes the alternative method that will be tested. The first difference can be seen in the 3rd level of the process were the regularized fitting criterion uses the Hankel matrix of the PBSID$_{opt}$ state space output equation. On level 4th the new PBSID$_{opt}$ inspired low rank upper triangular matrix appears, and on the last level it is important to notice that the solving approaches that will be mainly used in this thesis is the ADMM and the iterative minimization. This results will be discussed on the next chapter.

3-2 Implementation

Here it will be presented the calculations and formulas that need to be taken into account to execute the selected solving algorithms.

3-2-1 ADMM

If we want to take the ADMM approach to solve this problem, we need to recall some theory behind the method.

If we want to solve the PBSID$_{opt}$ pareto convex optimization, we need to state the problem in a generic nuclear norm optimization problem with a quadratic regularization term:

$$\text{minimize} \quad \| \mathcal{A}(x) + A_0 \|_* + \frac{1}{2}(x-a)^T H(x-a) \tag{3-10}$$

When this is done then, the ADMM algorithm focuses on solving the optimization problem by breaking it into smaller pieces, each one easier to handle. For this, we will see our problem in the form

$$\min \;\; f(x) \qquad \text{s.t. } Ax = b \tag{3-11}$$

If we define our problem as a dual problem and use a *augmented Lagrangian* function $\mathcal{L}$, we can have the next reformulation

$$\mathcal{L}(x; y, c) = f(x) + \langle y, (Ax-b) \rangle + \frac{c}{2} \|Ax - b\|^2 \tag{3-12}$$

where the primal variable to optimize is x, the dual variable is y (in this standard basic example is not related with the output of the system) and c is the penalty variable. It is also relevant to mention that $\langle , \rangle$ is the inner product between the vectors. Viewed differently, the unconstrained objective is the Lagrangian of the constrained problem, with an additional penalty term (the augmentation).

The objective function is separable in x and y. The dual update requires solving a proximity function in x and y at the same time; the ADMM technique allows this problem to be solved approximately by first solving for x with y fixed, and then solving for y with x fixed, using a singular value thresholding algorithm ([11]). Rather than iterate until convergence, the algorithm proceeds directly to updating the dual variable and then repeating the process. This is not equivalent to the exact minimization, but surprisingly, it can still be shown that this method converges to the right answer (under some assumptions) [7]. Hence, our original problem in Eq. (3-10) reformulates into

$$\mathscr{L}(x;X,Z) = \|X\|_* + \frac{1}{2}(x-a)^T H(x-a) + \mathrm{tr}(Z^T(\mathscr{A}(x)+A_0-X)) + \frac{\rho}{2}\|\mathscr{A}(x)+A_0-X\|_F^2 \tag{3-13}$$

The linear mapping matrix $\mathscr{A}(x)$ that first appear in Eq. (3-10) should be our observability-times-controllability matrix $\tilde{\Gamma}^{(f)}\mathscr{K}$, that will be formed by the optimization variable x (which contains the Markov parameters in a vector form).

If the matrix $\tilde{\Gamma}^{(f)}\mathscr{K}$ and the vector $C\mathscr{K}$ are multiplied by the stacked data matrix $Z_{k,p,N-p}$ we obtain the low rank observability-times-controllability matrix and the output vector $Y_{p,N-p}$ respectively.

It is easy to notice that in the quadratic cost function we need to modify our terms to have only the optimization variables. Leaving in our x term only the Markov parameters, leaving out the stack data matrix.

To do this, we need to follow the next modifications:

Our first step will be to change our variables to vertical vectors. Since our terms inside the quadratic function are vertical vectors, we need to transpose our $Y_{k,p,N}$ vector, which means now that $Z_{k,p,N}$ will be of dimensions $N-p \times p(l+r)$ and $\mathscr{K}$ will have dimensions $p(l+r) \times 1$, thus

$$Y_{k,p,N} = Z_{k,p,N}\mathscr{K}C \tag{3-14}$$

Secondly, we will introduce an extra term that will make our problem significantly smaller if the amount of data is big. This term is $Z^T_{k,p,N-p}$, that multiplied by all of our terms will create the next equation

$$(Z^T_{k,p,N}Y_{k,p,N} - Z^T_{k,p,N}Z_{k,p,N}\mathscr{K}C) \tag{3-15}$$

To simplify this term, we will use from now on the expression $Z^T Z_{k,p,N}$ for $Z^T_{k,p,N}Z_{k,p,N}$, thus

$$(Z^T_{k,p,N}Y_{k,p,N} - Z^T Z_{k,p,N}\mathscr{K}C) \tag{3-16}$$

This means that our vectors, instead of being a length of $N-p$ will be size $p(l+r)$

Next, we need to compare both equations and make sure that $x = C\mathscr{K}$

$$(Z_{k,p,N}^T Y_{k,p,N} - Z^T Z_{k,p,N}\mathscr{K}C)^T H (Z_{k,p,N}^T Y_{k,p,N} - Z^T Z_{k,p,N}\mathscr{K}C)$$

$$=$$

$$(a-x)^T H(a-x)$$

$$=$$

$$(Y_{k,p,N}^T Z_{k,p,N} - C^T\mathscr{K}^T Z^T Z_{k,p,N})\ H\ (Z_{k,p,N}^T Y_{k,p,N} - Z^T Z_{k,p,N}\mathscr{K}C) \qquad H = I$$

$$=$$

$$(Y_{k,p,N}^T Z_{k,p,N}(Z^T Z_{k,p,N})^{-1} - C^T\mathscr{K}^T)\ H\ ((Z^T Z_{k,p,N})^{-1} Z_{k,p,N}^T Y_{k,p,N} - \mathscr{K}C) \qquad H = (Z^T Z_{k,p,N})^2$$

Which means that $a = (Z^T Z_{k,p,N})^{-1} Z_{k,p,N}^T Y_{k,p,N}$

3-2-2 Calculation of the system matrices

In this section 2 options will be explained that can be used to calculate the system matrices once the algorithm finds a suitable low rank matrix.

The convex optimization problem from Eq. (3-9) for each λ yields a $C\mathscr{K}$ vector (which is the variable x), this vector will be used to create the matrix $\tilde{\Gamma}^{(f)}\mathscr{K}$. This matrix will be multiplied by the stacked data matrix $Z_{k,p,N}$ to create the low rank matrix.

Once the low rank matrix is computed (this needs to be done for each λ), the order of the system should be obtained. The order can be acquired two ways. The first can be from the rank of the low rank matrix X obtained from the ADMM algorithm. This matrix rank was defined based on a operation called 'singular value soft-thresholding' (see [11]). The second option is to compute the singular values of $\tilde{\Gamma}^{(f)}\mathscr{K}Z_{k,p,N}$ and select manually or automatically (see [4]) the order based on the gap between the dominant and the weak singular values.

Ones the order is selected, an SVD of the low rank matrix will be done and only the dominant singular values and left/right-singular vectors will be selected to do one of two options: a least square solution or use the MATLAB system identification toolbox `dx2abcd` function to obtain the system matrices.

3-2-3 Other calculations

For the first part of the ADMM optimization its necessary to minimize over x the *augmented Lagrangian* of the generic problem stated in Eq. (3-13). This will lead to the next equation

$$(H + \rho M)\hat{x} = Ha - \mathscr{A}_{adj}(Z + \rho A_0 - \rho X) \tag{3-17}$$

where $\mathscr{A}_{adj}$ is the adjoint of the linear mapping matrix $\mathscr{A}(x)$. Both terms in the terms of the PBSID$_{opt}$ approach are detailed in **Appendix A**

Also, he M matrix is a positive semidefinite matrix defined by the identity

$$Mz = \mathscr{A}_{adj}(\mathscr{A}(z)) \qquad \forall\ z \tag{3-18}$$

which its obtained by summing the diagonals of the original matrix $Z_{k,p,N}Z_{k,p,N}^T \in \mathbb{R}^{p(l+r)\times p(l+r)} = \begin{bmatrix}\xi_{ij}\end{bmatrix}$ as showed below. If $L = \begin{bmatrix}L_1 & L_2 & L_3 & L_4 & \cdots\end{bmatrix} = \begin{bmatrix}0 & (l\times r) & 2(l\times r) & 3(l\times r) & \cdots\end{bmatrix}$ then,

$$M = \begin{bmatrix}m_{ij}\end{bmatrix}$$

$$m_{ij} = \begin{cases} m_{ji}, & i < j \\ \sum\limits_{w=i}^{ceil((p(l\times r)-i+1)/(l\times r))} \xi_{(i+L_w)(j+L_w)}, & i \geq j \end{cases} \tag{3-19}$$

3-3 Reweighted PBSID Nuclear Norm

The reweighted N2PBSID implementation is straightforward performed as indicated in [8].

To start the implementation of the reweighted nuclear norm with our PBSID$_{opt}$ approach we need to use the Eq. (2-6) in combination with Eq. (3-9), which is seen as

$$\text{minimize} \quad \| W_1^k \Gamma^{(f)} \mathcal{K} Z_{k,p,N} W_2^k \|_* + \frac{\lambda}{N} \| Y_{p,N-p} - C\mathcal{K} Z_{k,p,N-p} \|_2^2 \tag{3-20}$$

Once we visualize this problem, the first step to start the algorithm is to initialize the weights W_1^k and W_2^k as identity matrices of adequate dimensions.

Afterwards, we need an initial $low\ rank\ matrix$ that can be obatined by the regular N2PBSID algorithm or any other algorithm that can solve the dual problem in Eq. (3-20) like the gradient projection algorithm. Once this matrix is obtained an SVD of its solution multiplied by the weights needs to be performed. An example of this can be visualized as

$$U\Sigma V^T = W_1^k \Gamma^{(f)} \mathcal{K} Z_{k,p,N} W_2^k \tag{3-21}$$

this SVD will be used to calculate the new components that are later used in the new weights, this new components are matrices calculated as

$$Y^{k+1} = (W_1^k)^{-1} U\Sigma U^T ((W_1^k)^T)^{-1}$$
$$Z^{k+1} = (W_2^k)^{-1} V\Sigma V^T ((W_2^k)^T)^{-1}$$

Now the weights will be updated as

$$W_1^{k+1} = (Y^{k+1} + \delta I)^{-1/2}$$
$$W_2^{k+1} = (Z^{k+1} + \delta I)^{-1/2}$$

were δ is a small regularization constant, in this case it was chosen to be 0.5.

If the termination criterion is reach the iterations stop, else set $k = k+1$ and solve the dual equation in Eq. (3-20) with the new weights and follow the steps until a satisfactory result in found.

Once we know how to apply this new algorithm theoretically and practically, on Chapter 4 a series of tests will be applied to this algorithm to verify if the method performs as a suitable system identification method, and if this does, it will be verified if it brings any benefits compared to the existing ones.

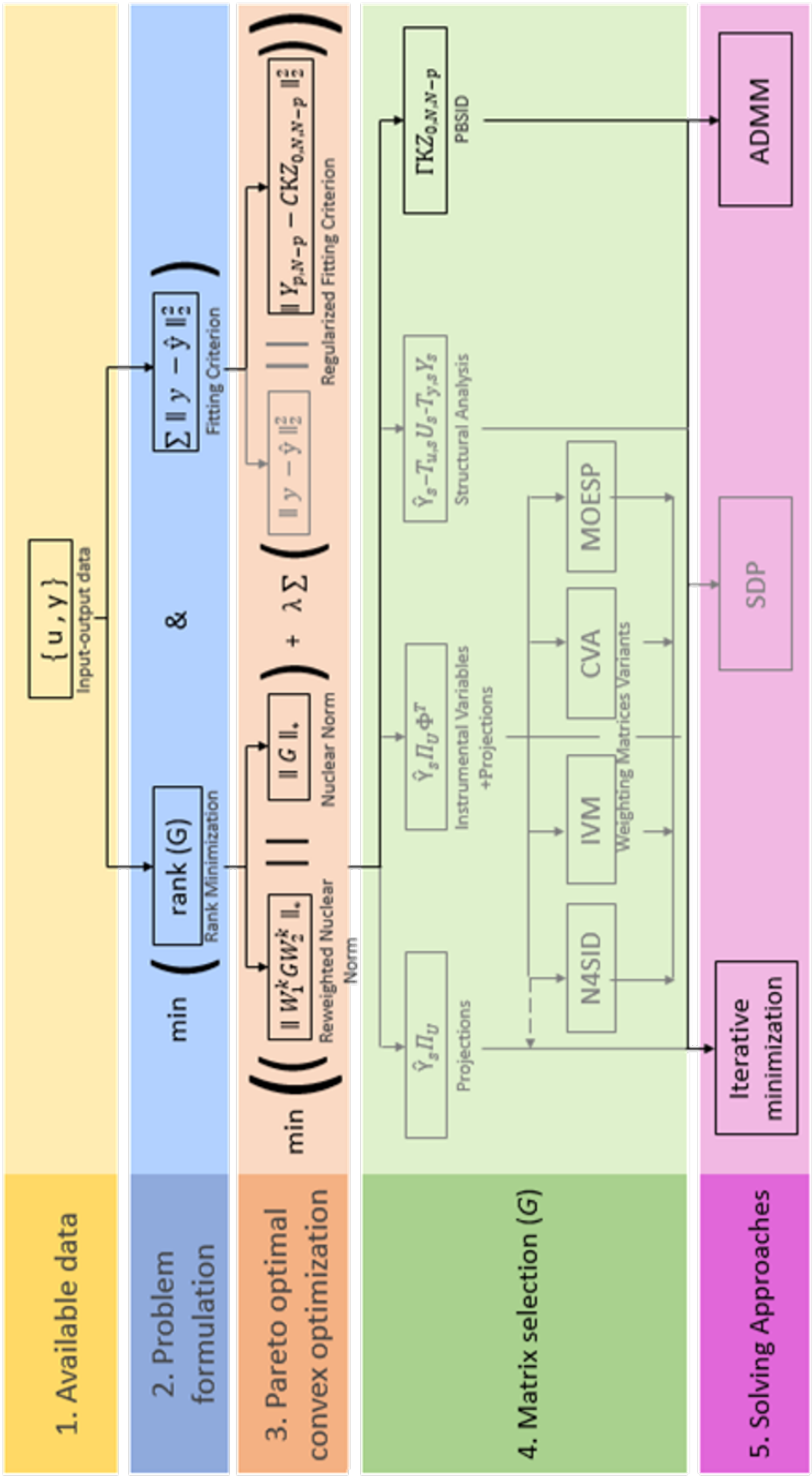


Figure 3-1: Diagram with the new branches inspired in the PBSID$_{opt}$ algorithm

Chapter 4

Results and discussion

In this chapter the results given by this new PBSID$_{opt}$ inspired approach in the ADMM algorithm will be presented, among comparisons with other existing methods, specifically N2SID and PBSID$_{opt}$. In addition, other characteristics like the length of the amount of available data and the reweighted nuclear norm approach will present interesting results that will be mention below.

4-0-1 Validation Study

The numerical results of such experiments will be reported in this section. These experiments were carried out in MATLAB R2013b running on an Intel Core i5-4200U CPU 1.6 GHz with 4 GB of RAM.

An example of the Matlab implementation of this experiments can be seen in **Appendix B**.

4-0-2 Data selection and pre-processing

The Nuclear Norm PBSID$_{opt}$ approach was implemented for SISO systems in several real-life DaISy (Database for the Identification of Systems) data experiments to know if satisfactory results were obtained. Later on, a defined second order transfer function is introduced for further experiments and examination of the algorithm.

The DaISy experiments selected are shown in Table 4-2 and their details are presented in Table 4-3. The $data\ length$ from the individual experiments are indicated by N$_{ide}$. Since many of the data sets contain poorly excited data at their beginning, the first del samples are discarded from each data set. Finally, each identified model is validated for each test case on the same validation data set. These validation data sets contain the N_{val} samples following the sample with index max(N$_{ide}$)+1.

On the other side, the defined transfer function will use different data (mainly randomly created) to compare results and examine on a more detailed level the solutions obtained.

This data details for both, the DaISy and the defined transfer function experiments may be changed for each experiment depending on the circumstances presented in each test, if this is the case, the changes will be defined in each section where this occurs.

4-0-3 Compared Identification methods

In this section the methods that will be involved in this study will be detailed and explained. The reasons why this methods were selected and important details about each one for implementation will be mentioned and described below.

The basis of comparison for the results given by the new PEM-SID PBSID$_{opt}$ approach, which from now own will be named N2PBSID, are compared to the ones found in [5], since this document was the basis of influence for the realization of this thesis. Ans in [5], in [4], which is a source that influenced [5] publication, has some of the the same SID methods for comparison, which makes them suitable to be the basis of comparison for the N2PBSID approach.

The methods that are going to be compared are the N4SID, the N2SID and the PEM. The WNN$_{opt}$ method of [4], included in [5] research, is not included since the DaISy dataset that were selected show a better fitting result for N2SID and N4SID ([5]), which means that the results given by WNN$_{opt}$ are not relevant to compare with the N2PBSID. In addition, the PBSID method was included since this is one of the methods that this new approach is based on, so it is consider interesting if this new method can have any advantage over the classic PBSID$_{opt}$ algorithm.

For N4SID the Matlab System Identification App is used in its default settings, no input delays are estimated and a guaranteed stable simulation model is identified. The model order is determined automatically in the code by using the option `'best'`.

For the N2SID method the ADMM algorithm is implemented with the same specifications mentioned in [5]. In the code that is provided in http://users.isy.liu.se/en/rt/hansson/, it is explained that in the algorithm the user can choose among 3 methods to compute the state-space realization. Since this point was not discussed in the paper, the 3 methods were simulated for each λ and for each N_{ide} experiment, and the one that got the best fit was selected.

The regular PBSID$_{opt}$ algorithm is included since its interesting how this new algorithm compares with the original PBSID$_{opt}$ algorithm. That's why the model order of PBSID$_{opt}$ is the same as the one determined in N2PBSID method.

4-1 Nuclear Norm dual minimization with PBSID theory (N2PBSID) results

DaISy experiments

As stated before, the four SID methods that will be tested in Table 4-1 will be compared for the datasets in Table 4-2. The results of this comparisons are for each experiment summarized in two graphs in the same figure. The left graph of the figure displays the goodness of the fit criterion VAF (Variance-Accounted-For) by simulating the model obtained from each method with the validation data. The fit measure is computed by the function `VAF` in Matlab's System Identification Toolbox. It is defined in percentage as

$$\text{fit} = \left(1 - \frac{\text{variance } (y - y_{pred})}{\text{variance } (y)}\right) * 100\% \tag{4-1}$$

Method	λ	p	Weighting
N4SID, [Ljung [2007]]	/	15	*automatic*
N2SID [Verhaegen]	$[10^{-3}, 10^{3}]$	15	/
N2PBSID Algorithm	$[10^{-3}, 10^{3}]$	15	/
PBSID$_{opt}$ Algorithm	/	15	/

Table 4-1: Methods used for identification and their user selection parameters

for a single output sequence, where y is the validation data output sequence and y_{pred} is the predicted output from the model.

The right graph of the figure displays the model complexity as defined by the model order of the state space model. This two graphs are graphed versus the length of the identification data bath as indicated by the symnol N$_{ide}$ in Table 4-3.

In order to evaluate additional information retrieved from the results, the singular values are computed for the N2PBSID over the N2SID. This methods will be compared because in [5] the N2SID algorithm is compared to WNN$_{\mathrm{opt}}$ from [12] and shows to have an improvement in the gap of the singular values, giving a better estimation of what the real system's order is. Therefore, this analysis will give further information to see if any improvements were obtained from the low rank detection in N2PBSID. The singular values will be compared from the best fit obtain in any of the data length sequences from both algorithms independently.

From these methods is important to mention that the key user selection parameters are listed in Table 4-1. The first user selection is the number of block rows p of the data Hankel matrices. In methods like N4SID, PBSID$_{opt}$ and N2PBSID a distinction is made in the block rows because they are referring to the past and future window of the Hankel matrices, while in N2SID this is not the case. Since picking this parameter is an open research problem, the number of block rows of past and future Hankel matrices was chosen to be equal to the number of block rows of the regular Hankel matrix in N2SID. Usually this number is chosen at least 2 times higher than the real system's order; since we don't know the real order, the number 15 was chosen conservatively because it is suspected that this mechanical SISO systems will not have an order higher than 5.

For N2PBSID the ADMM algorithm presented in last chapter is implemented. The regularization parameter λ spectrum for optimization is given 20 values between 10^{-3} and 10^{3} using the `logspace` Matlab function. The maximum number of iterations in the ADMM algorithm has been set to 200, since in the majority of the cases the final result doesn't change after this number in the algorithm; and the solution accuracy tolerances have been set to 10^{-3} and 10^{-6} for relative and absolute accuracy respectively. The parameters used to update the penalty parameter have been set to $\tau = 2$ and $\mu = 10$. The final model order is selected by the user, and this will be selected depending on the best fit on the validation data taking into account the model's order, that will be selected to be taking into account the one obtained by N2SID.

Experiment 1: Hair dryer (data set 96-006)

The results of this experiment can be seen in Figure 4-1. The goodness of the fitting criterion and the estimated model order are seen on the left and right of the figure respectively.

The N2PBSID algorithm is effective in finding an appropriate model that describes the system's behavior with a VAF mean of approximately 99%, were the best result was given when the data

Nr	Data set	Description	N_{tot}
1	96-006	Hair dryer	1000
2	96-002	Steam heat exchanger	4000
3	96-009	Flexible arm robot	1024

Table 4-2: Data sets form the DaISy collection, [13]; N_{tot} is the total number of data samples available

Nr	N_{ide}										del	N_{val}
1	80	100	120	140	160	180	200	250	300	400	120	450
2	150	200	300	500	750	1000	1250	1500	1750	2000	200	1500
3	80	100	120	140	160	180	200	250	300	400	120	450

Table 4-3: N_{ide} indicates the increasing length of the data sets used for identification starting with samples index del; N_{val} indicates the length of the validation data starting with samples index $max(N_{ide})+1$)

available was 400. However, it doesn't show a better fitting criterion that the other methods besides N4SID. The method that outperforms the others in this experiment is the PBSID_{opt} since its the one that has the highest fitting criterion in 6 out of the 10 data simulations. Subsequently, N2PBSID and N2SID are next in their performances. Even though both present approximately the same VAF mean value, N2SID goth better results between N_{ide} = 250 and 400 and its highest VAF surpass the highest VAF of N2PBSID. Additionally, N2PBSID and PBSID provide a more stable behavior than other methods, when N_{ide} increases the fitting value seem to increase too; this is not the case of N4SID and N2SID.

In this case N2SID and N2PBSID present the same order's model which is $n = 3$, that can be easily seen in the biggest gap of the singular values presented in Figure 4-3 (furthermore, the N2PBSID method seem to improve the singular value gap spacing over N2SID).

Experiment 2: Steam heat exchanger (data set 96-002)

Once again, the goodness of the fitting criterion and the estimated model order are shown in Figure 4-3.

In this experiment the data was detrended before analysing. Usually detrending is used to remove a feature thought to distort or obscure the relationships of interest [14]. In this case the trend presented in the data lead to undesired results. This was the only dataset were the data was detrended since the other two experiments presented approximately the same results without applying the detrending pre-step, if not slightly worse.

In this method is clear that the N2PBSID outperforms all the other methods. Not only has 100% of the cases the highest fitting value, is the one that tends to have the smallest model order estimation.

In this experiment it can also be seen that the accuracy of the model is higher when more data is present for identification purposes, unlike N2SID. This can be seen from N_{ide} =1000 to 2000, when the N2PBSID algorithm chooses a first order model for its result. Besides that, the highest fitting accuracy (N_{ide}=750) is where the N2PBSID algorithm chooses the highest order model, making this a consistent algorithm in the trade off between fitting accuracy and model order estimation.

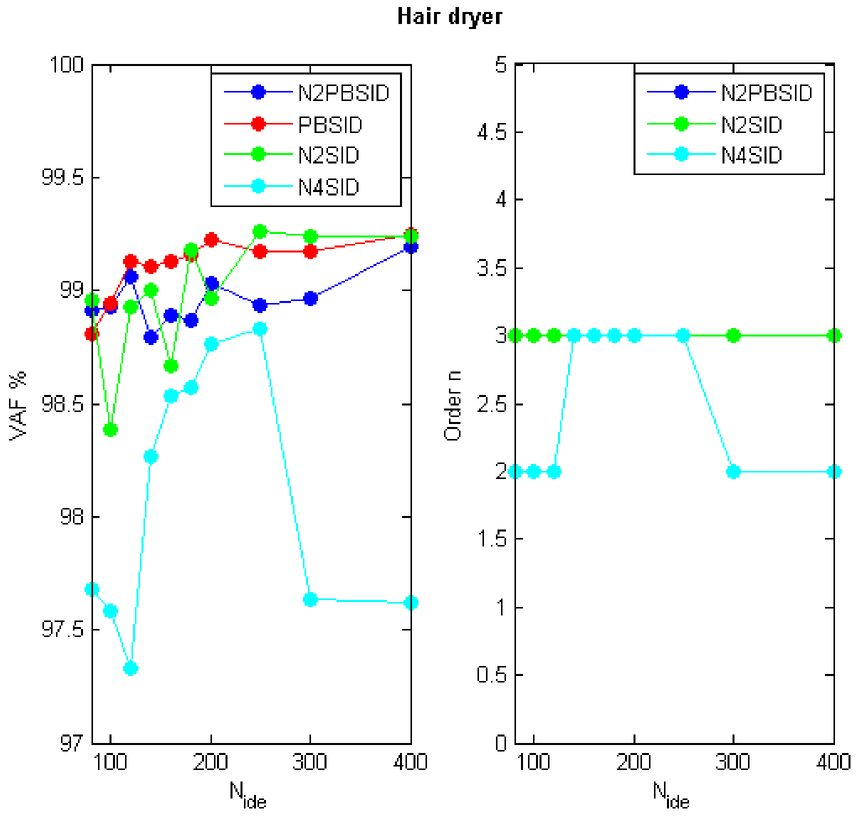


Figure 4-1: VAF and Order DalSy #1 - Hair dryer

The N4SID and PBSID$_{opt}$ methods cannot compete with the last two algorithms. PBSID$_{opt}$ seem to get a good value on the first experiment, but as the amount of data available for analysis increases the method seem to deteriorate in its efficiency, which means that it is not reliable for this kind of experiments. The same can be said about N4SID. In detail, the VAF values in this experiment were lower than in the first experiment, it seems that is due to lower signal to noise ration or system nonlinearities; which seem to be a problem to handle for the PBSID algorithm.

Also, the singular values of the best fits of both experiments are presented in Figure 4-4. In this plot it is seen again that the N2PBSID method has an advantage over N2SID to detect easier the order of the model by making larger the gap between the significant singular values from the other ones. In N2SID the model order chosen for N$_{ide}$ = 750 was $n = 4$, which is consistent with what its seen on the plot, while with the N2PBSID method is its clear that the model order selected should be $n = 3$ for the N$_{ide}$ = 750 data batch. This is not the case since in the experiments it can be that for N$_{ide}$ = 750 the order of the estimated model was chosen to be $n = 4$ just as in N2SID, this happened because the number of λ experiments is limited, and the chosen interval values of the regularization parameter were to wide, thus, the λ values available didn't identify a model $n = 3$ that could be chosen.

Experiment 3: Robot arm (data set 96-009)

The goodness of the fitting criterion and the estimated model order are shown in Figure 4-5.

In this experiment a new change was made. The vector in Eq. (3-10) named a is reformulated.

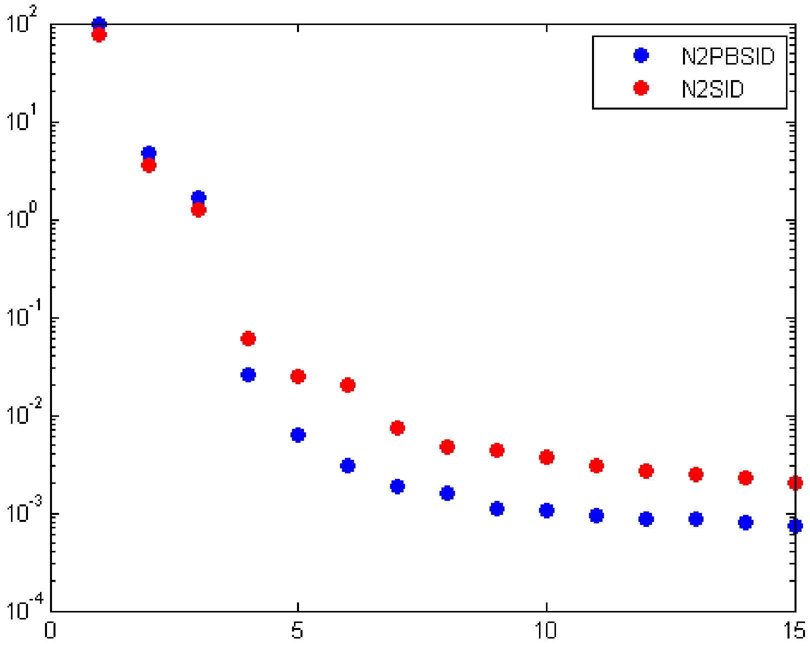


Figure 4-2: Singular values of DaISy #1 - Hair dryer

$$\text{minimize} \quad \| \mathscr{A}(x) + A_0 \|_* + \frac{1}{2}(x-a)^T H(x-a)$$

In the original equation this vector's value was given by $a = (Z^T Z_{k,p,N})^{-1} Z^T_{k,p,N} Y_{k,p,N}$, while in this experiment the vector utilizes the MATLAB function `pinv` to create a pseudo-inverse matrix. This matrix allows a more stable formulation because, when dealing with this specific experiment, the inverse of the multiplication of two matrices that have small values (input and output) can create a singular or almost singular matrix, which creates really inaccurate further formulations. The new a vector's formula reads as $a = (Z^T_{k,p,N})^\dagger Y^T_{k,p,N}$.

It was tested in the other experiments if this vector change could improve the results obtained. The conclusion was that this formulation only affected significantly this experiment, that's why the original formulation was kept for the last couple of experiments since it was theoretically proven how to get to that formulation in Chapter 3 and therefore easier to follow for future references.

The N2SID method seem to be the leading one in fitting values compared to all the others. In the first plot it can be seen that the N4SID and PBSID algorithms fail to get a good fitting criterion for this system, were N4SID displayed values around 20% and PBSID didn't show any result.

Even though the N2SID algorithm has a better VAF in 90% of the cases, the N2PBSID algorithm seems more stable and shows clearly the trade-off between model complexity and model accuracy. The blues lines representing the N2PBSID in both plots show the same pattern, the higher the order of the model, the accuracy follows as well; nonetheless just in one case the N2PBSID shows the same fitting value as N2SID when they have the same high order (N$_{ide}$ = 180), in other instances even when both model orders are high (like in N$_{ide}$ = 120), the fitting criterion of N2PBSID doesn't match up with the model found with N2SID.

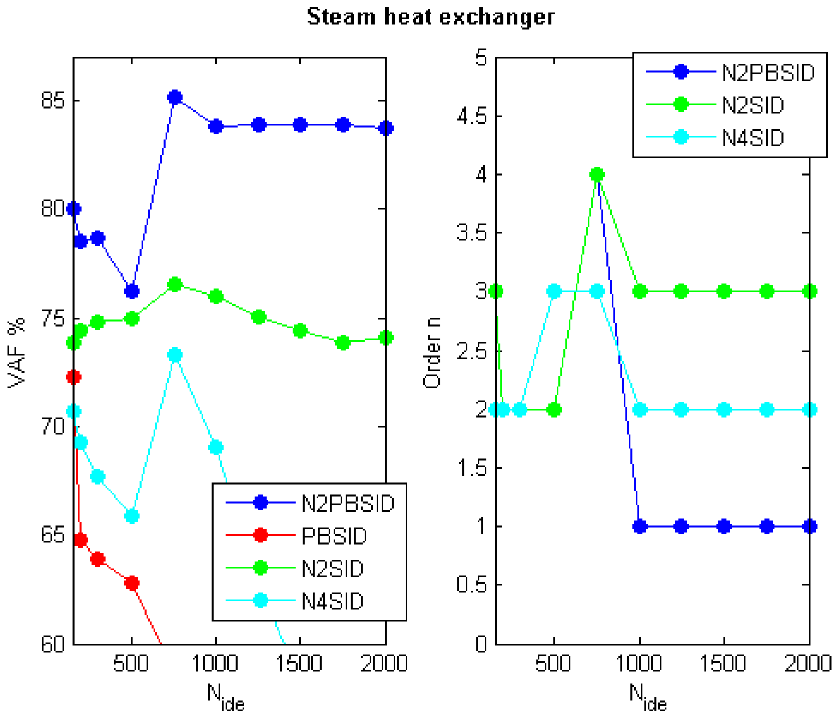


Figure 4-3: VAF and Order DalSy #2 - Steam heat exchanger

Experiment	N_{ide}	N2SID (sec)	N2PBSID (sec)
1 (96-006)	400	24	18
2 (96-002)	2000	260	40
3 (96-009)	400	20	12

Table 4-4: Time taken by each iterative algorithm to find a suitable solution with the maximum amount of data available

It is interesting to mention too that the singular values presented in Figure 4-6 are interesting. Both models present a different visualization of the singular values. N2SID shows a bigger gap in the 4th singular value, while N2PBSID has two major gaps, one in the 2nd and the other one in the 7th singular value. Nonetheless both model orders seen in the singular values differ from the model order that in selected for the estimated system. Both methods chose a 8th order, which doesn't coincide with the gaps in Figure 4-6 or the missing λ values available for the model. Which makes this problem an open research question that can be examined furthermore.

Execution time

It was of interest to test the speed of the algorithms that require iterations for its solution. This was tested for each of the DalSy experiments with the largest identification data available $max(N_{ide})$. The times are presented in Table 4-4.

It can be seen that the N2PBSID algorithm is faster than the N2SID algorithm. It is hypothesize that this occurs because the N2PBSID algorithm doesn't require to calculate any Fourier transformation matrix which N2SID does. This speeds the algorithm up to 5 times.

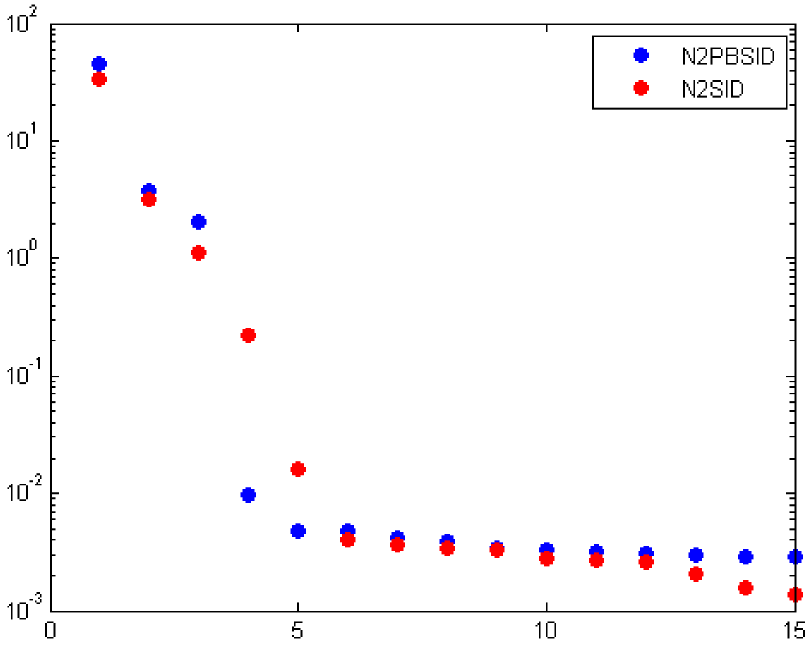


Figure 4-4: Singular values of DaISy #2 - Steam heat exchanger

4-1-1 Small number of samples

As is known, SID methods perform better when more information is used. This is due to the irregularities that short data batches can contain, some examples can be noise trends, non-exciting input signals, nonlinearities, etc.; which makes them not representative of the behavior of the whole system, altogether affecting the results from the estimation model. Equally important, some SID methods use projections to derive the low rank matrix that lead to the approximate model, and this can lead to a loss of information, which again, can result in poor estimation models.

On the other hand, in cases were there is a limited amount of available data a trade-off occurs between the number of block rows (the user parameter p) and the amount of columns in the Hankel matrices of stack data. As it is known, there is an structural property that states that the number of block rows must be higher than the order of the system to have a low rank matrix approximation ($p > n$), but when there is a short amount of data available, the number of block rows also must be low enough to create a Hankel matrix of stack data with enough columns. After all, the amount of columns can affect the real rank of the matrix and future calculations can be influenced negatively by this case.

Therefore, the motivation behind testing the N2PBSID algorithm with batches of really small amount of samples, is to see if in this specific case the method can handle this situation, and if it does, notice if it shows any advantage over the other methods already available.

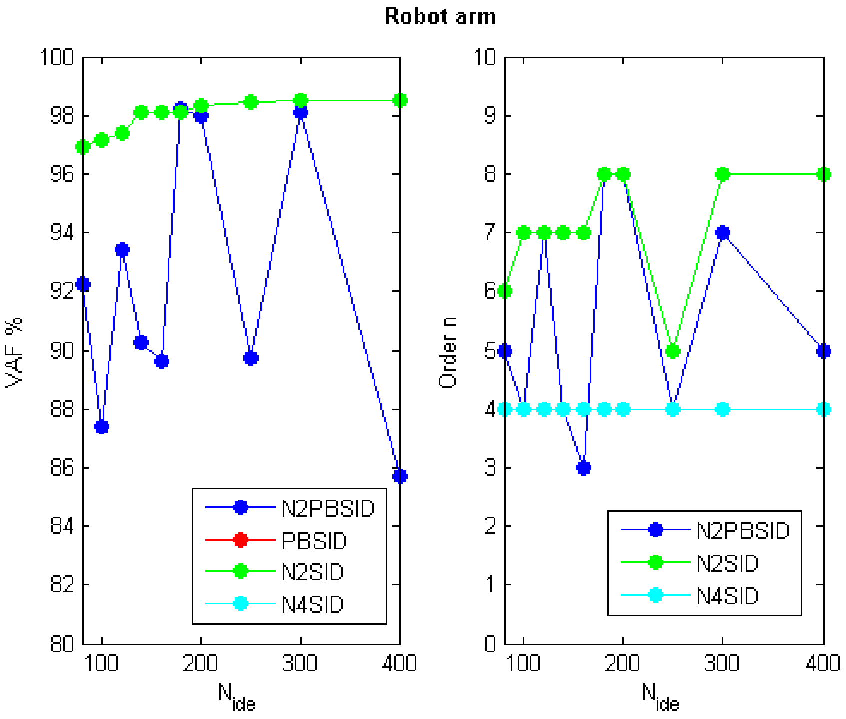


Figure 4-5: VAF and Order DaISy # 3 - Robot arm

Defined transfer function

The first experiments were performed with a specifically chosen second order transfer function. Inputs for identification and validation were created randomly for this purpose. The past window block rows user parameter was chosen to be $p = 9$ and the amount of data available for identification was chosen to be 18 samples (2 times the block row parameter), while the validation was chosen to be 3 times the identification samples this amount with 54 samples, and the relative accuracy tolerance was set to 10^{-1}. The rest of the parameters, like the absolute accuracy tolerance, penalty parameters, the maximum amount of iterations, etc. were chosen with the same values as in the DaISy experiments.

Certainly, it is important to mention that for small data batches the change in the a vector is crucial. As mentioned before, the noise behavior can affect the identified model, in this case it can make the formula $\mathrm{inv}(ZZ^T_{k,p,N})$ singular or badly scaled, which means that when multiplying $ZZ^T_{k,p,N} \times \mathrm{inv}(ZZ^T_{k,p,N})$ the result will not result in identity, which makes the algorithm find wrong solutions when handling this equation for the fitting criterion. Thus, the a vector was again changed for $a = (Z^T_{k,p,N})^{\dagger} Y^T_{k,p,N}$.

Also, the N4SID method is not shown in the results, because the amount of values for identification was not enough for analysis.

The results obtained from this experiment are summarized in Figure 4-8. It was noticed that while repeating random experiments the results obtained were not always the same. It seem that depending on the noise presented on the system the algorithms sometimes could and sometimes could not detect a model. That's why it was decided to make 100 consecutive experiments to see what kind of results the methods tend to exhibit.

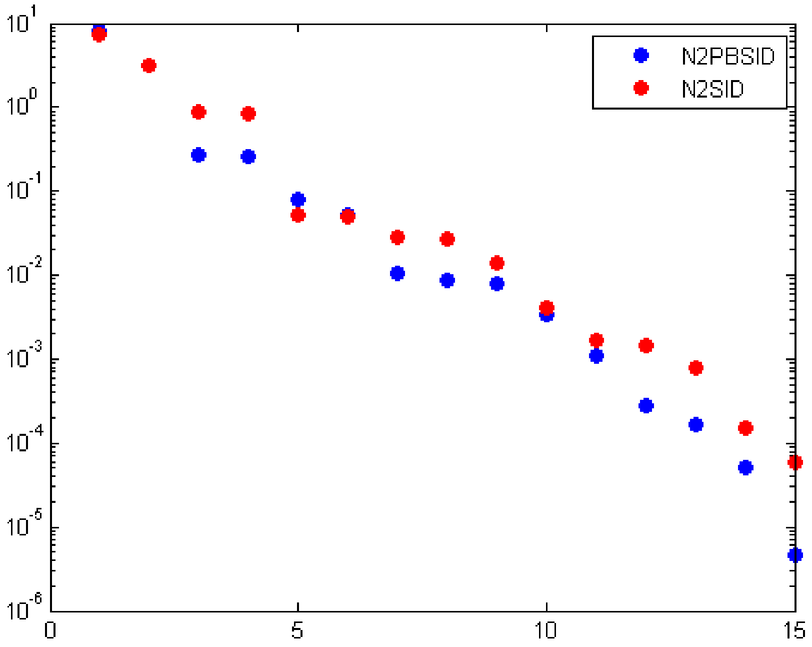


Figure 4-6: Singular values of DaISy #3 - Robot arm

What we can see in Figure 4-8 is that on the left plot each circle represent one experiment and its height represents the fitting obtained from the estimated model. On the right plot a representation of the amount of instances the experiments had, in percentage, in a specific range of VAF fitting obtained by the models

For example, it is shown in Figure 4-8 that the PBSID$_{opt}$ method couldn't find suitable solutions due to the bad scaling of inv($ZZ^T_{k,p,N}$), all results tend to have a fitting of 0%. On the other hand, the N2SID seem to be suitable for most cases, it can be seen in Figure 4-8 that in approximately 75% of the cases the model selected has between 95% and 100% fitting against the validation output. From the other 25% is important to mention that almost all of the models had a fitting VAF higher than 60%, that show that in most cases a good model is guaranteed. While in N2PBSID this is not the case, even though almost 50% of the cases have a 95%-100% fitting against the validation output, and other 20% present a good model estimation with a VAF between from 80% to 95%, there are other cases were the fitting was not satisfactory; approximately 10% didn't show any result at all, and the other (approximately) 20% had a VAF between 20% and 80% fitting.

It was verified that for the N2PBSID those bad model cases were related solely to the noise, because some inputs and outputs were tested with different noise vectors, and the algorithm showed that for some specific noise sequences the results of the estimated model were not adequate. When analysis visually this specific noises sequences nothing particular was observed, it is hypothesize/-believed that in this specific sequences the noise is colored or has one or several outliers/spikes that affect how the system interprets the overall noise in the outputs.

Nonetheless another experiment with a slight increase in the identification samples gives a different perspective. This time the same experiment (with the same amount of data for validation N$_{val}$ = 54) was performed with N$_{ide}$ = 27 samples for identification, which makes the Hankel ma-

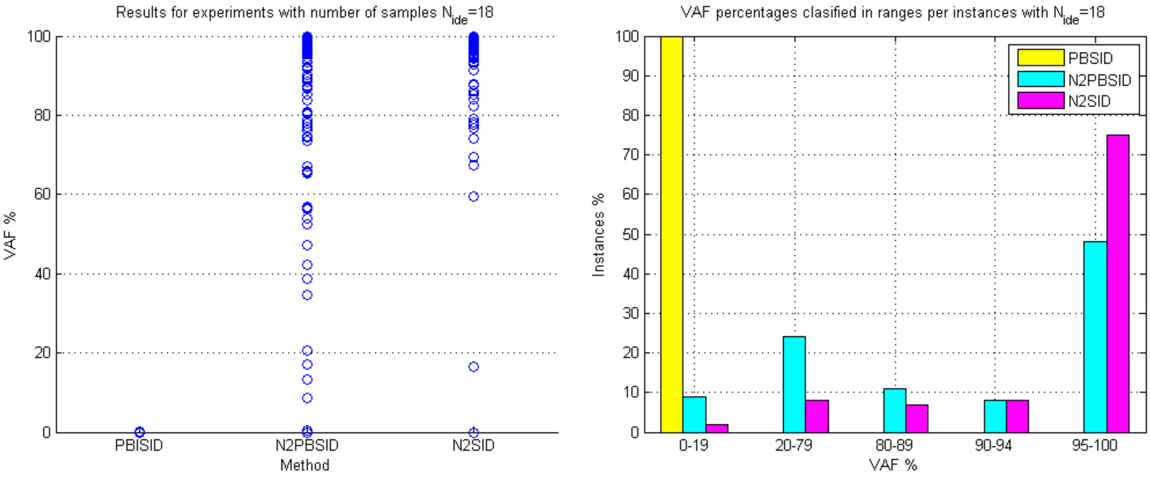


Figure 4-8: Left: VAF representation of 100 random experiments with N$_{ide} = 18.Right:PercentageofinstancesoftheVAFvaluesobtainedwithN_{ide} = 18$

trix of stack data a squared matrix.

The results of these experiments can be seen in Figure 4-10. Then again, on the left each circle represent one experiment and its height represents the fitting obtained from the estimated model. On the right a representation of the amount of instances the experiments had, in percentage, in a specific range of VAF fitting obtained by the models.

In Figure 4-10 you can now see that the PBSID method is able to get better results than in the last experiments, yet again the noise batches affected at least 30% of the samples obtaining fitting models of with less than 60% fitting, nevertheless approximately 50% of the models obtained had a fitting between 95% and 100%, but this results are not comparable with the ones obtained with N2PBSID. In N2PBSID more than 95% of the experiments could obtain a model, and almost all of them (more than 90%) had a model with a fitting between 95% and 100% VAF. Only a small percentage of the results landed in the range of 90% to 95% fitting and less than 10% didn't obtain a model whatsoever. This makes a N2PBSID a better method than N2SID. On the other hand, the N2SID method seem to show the same trend as the previous experiments, since approximately 80% of the models have the 95% to 100% accuracy compared with the 75% of the last experiment; and all the other experiments showed almost the same values of fitting of less than 10%.

DaISy experiments

Thanks to the baffle results obtained in these experiments with random noise sequences and inputs, another experiment was designed. It was decided that this new experiment will use some of the DaISy datasets and validation data but with a much smaller amount of identification samples, this with the purpose of testing real systems to know if the results given resemble the ones of the previous 2 experiments.

The same algorithm parameters were chosen as the ones detailed in section 4-2, the only difference is that for identification purposes the data batches were selected to be of 30 samples long (twice the amount of past window block rows of the Hankel matrices), and 10 different batches were selected from the data available, to see if this batches had differences in the results presented.

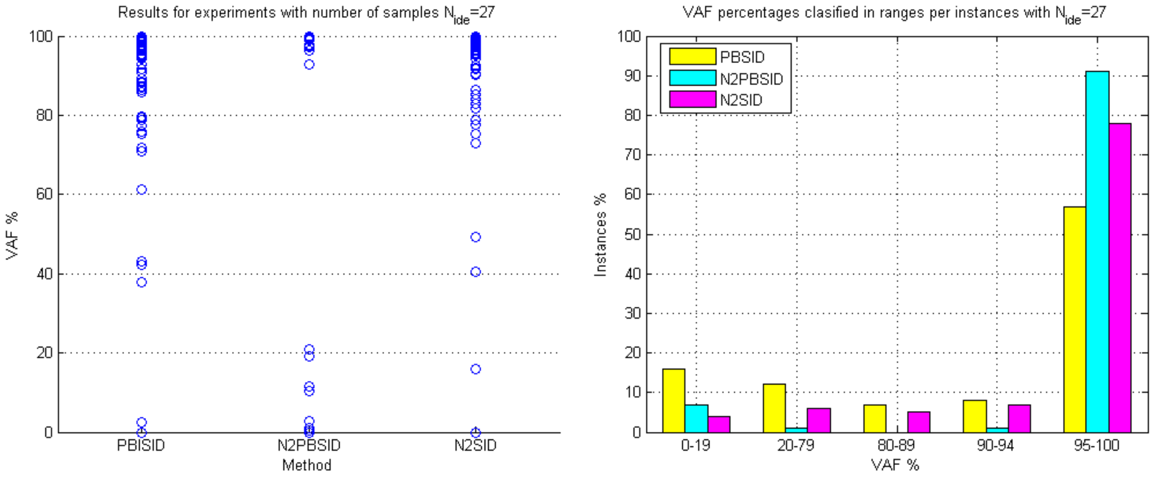


Figure 4-10: Left: VAF representation of 100 random experiments with N$_{ide}$ = 27.$Right:$ $Percentageofinstancesof theVAFvaluesobtainedwithN_{ide}$ = 27

These batches had been selected by first discarding the poorly excited data (del), and then grabbing intervals of 30 samples until 10 batches are collected. Furthermore, each batch is validated on the same validation data set. This validation data sets contain the same N_{val} samples following the sample with index max(N_{ide}) from Table 4-3.

Then again, the N4SID method results will not be presented since the amount of data available for analysis is not enough for identification.

Experiment 1: Hair dryer (data set 96-006)

Results of this experiment can be seen in Figure 4-11. On the left plot it can be seen that in 8 out of the 10 experiments the N2PBSID method outperforms the fitting of the results obtained with N2SID method, while the results on the right plot shows a equivalence in the order selected by both methods (mainly on order $n = 2$).

It can also be seen that in the first sample batch the fitting of both methods sets around the 50% and 60% VAF, which means that the first sample was not representative of the system, meaning that it is highly likely that the input was not exciting enough or the noise present at that particular stage was high for the signal to noise ratio be small.

Nonetheless, the N2PBSID seem to have increased in accuracy on comparison with the N2SID method, since in the first experiment with more data available, shown in Figure 4-1, both methods had comparable results, now it seems that the N2PBSID method is better.

Likewise, it can be noticed that even when the results of the last experiment always round in the 99%, the fitting values presented in the small batch experiments are around the 95%. Furthermore, a reduction on the order of the system identified was present from $n = 3$ to $n = 2$. This means that the results obtained when a really small amount of data is present, even when the accuracy is lower, are acceptable and satisfactory with a more simple model order.

Experiment 2: Steam heat exchanger (data set 96-002)

The results of the second experiment can be seen in Figure 4-12. Now it can be seen on the left plot that the N2PBSID methods outperforms the fitting results of the N2PBSID method. This coincides with the results obtained with the first experiment in 70% of the times.

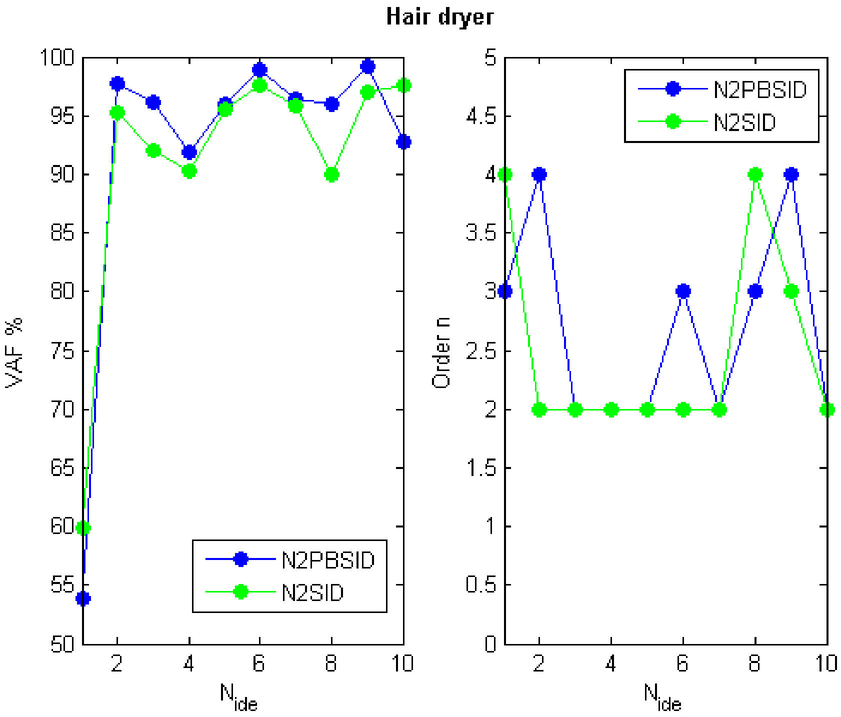


Figure 4-11: VAF and Order for small number of samples DalSy #1 - Hair dryer

Despite that, it can be seen that the N2PBSID method couldn't managed to analyse correctly the last experiments, giving on batch 7, 9 and 10 results below 70% fitting while N2SID was able to give suitable results for all samples.

On the first experiment with a bigger amount of data the right plot shows a model order $n = 1$, while on this experiment the order of the model, on the majority of the cases, seems to be $n = 1$, while the N2SID shows a model order in both experiments at $n = 3$.

It can be concluded that, again, the fitting values of N2PBSID with the limited amount of data are close to the values obtained by the experiments done with all the samples, making this method suitable for this kind of cases.

Experiment 3: Robot arm (data set 96-009)

It was decided that the results of this experiment were not going to be included in this report. The results obtained for both methods, N2PBSID and N2SID, were not only inconsistent (each experiment gave a vast range of different results), but the fitting values obtained were on the majority below the 60%. These poor results are presumed to be explained by the complexity of the system, a low noise to signal ratio and/or a non-representative or non well excited sample of data of the system.

4-2 Reweighted nuclear norm application

The reweighted nuclear norm implementation of the already described algorithm is tested here. The purpose of this implementation is to see if further developments can be made to the singular

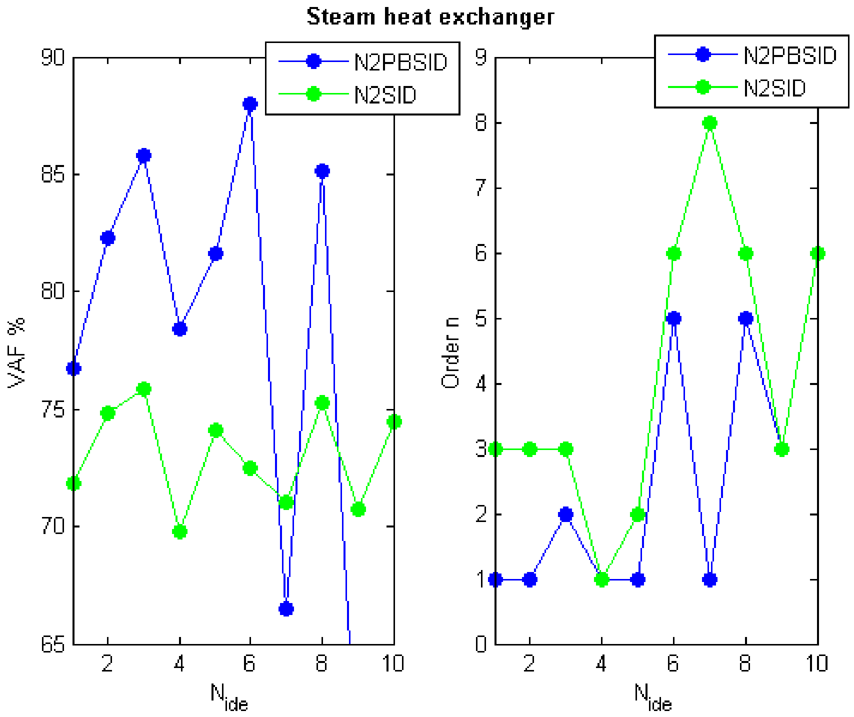


Figure 4-12: VAF and Order for small number of samples DaISy #2 - Steam hear exchanger

values, also it will be checked if the fitting accuracy and the order of the system picked is changed thanks to the iterations made in the algorithm.

This exercise main goal is to see a change in the singular values, were the gap between the dominant singular values and the weak singular values increases, by means of getting higher dominant singular values, lower the weak singular values or a combination of both.

It is indispensable to mention that the first initial results are given by the original N2PBSID algorithms. Then for the weighted iterations the SDP solver cvx program is used to find a solution since the ADMM implementation with the weights require some changes requiring linear algebra calculations. Due to time limits, it was decided to use the cvx solver, so this modification is open for further research to see if the ADMM algorithm can get different results from the cvx solver.

Defined transfer function

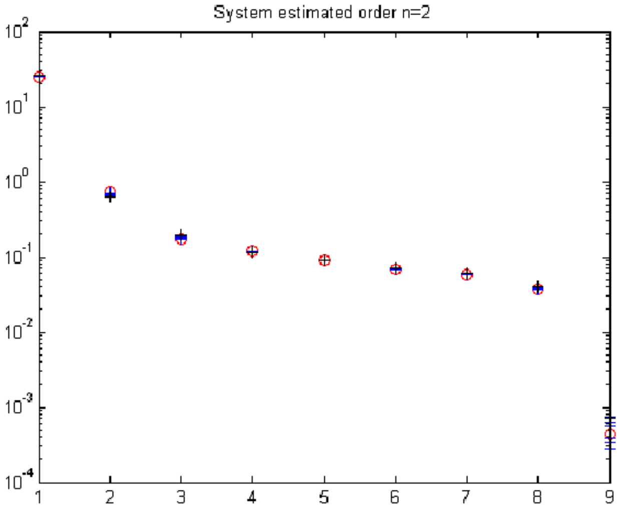
The first experiment was set to be a given transfer function. Again the same second order transfer function as in the experiment with the small batches of data was tested. The same user choice's and parameters are used except for the λ range values, it was changed from 10^{-3} and 10^{3} to 10^{-1} to 10^{5} because it seem that better results were observed if the range was increased towards a high λ. The amount of data for identification is $N_{ide} = 90$ samples (a number that was tested with regular N2PBSID as a choice that always gave good results) and the validation data is the same amount with $N_{val} = 90$ as suggested in [15].

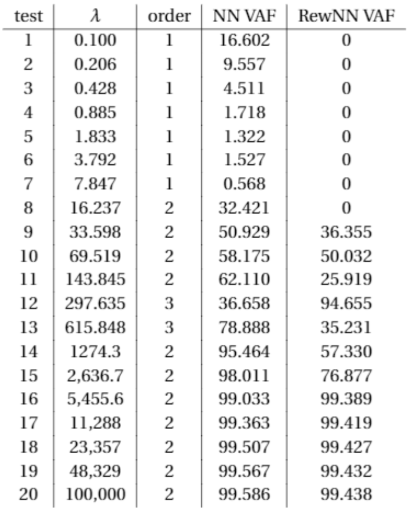
Also, the number of iterations used to see if the weights were effective in moving the singular values was 20, since more iterations didn't change the results and also took a bigger computation time to perform.

The plots that will be shown representing the singular values will have two types of markers, a circle and a plus, the red plus represents the singular values found by the regular N2PBSID algorithm, the blue pluses represent the changes made by the weighted iterations, were the lighter blue pluses are the first iterations and the dark blue or black ones are the latest iterations.

Examples of the results obtained are presented here.

The majority of the results presented with random inputs and noise sequences are (representative) depicted in Figure 4-13.





test	λ	order	NN VAF	RewNN VAF
1	0.100	1	16.602	0
2	0.206	1	9.557	0
3	0.428	1	4.511	0
4	0.885	1	1.718	0
5	1.833	1	1.322	0
6	3.792	1	1.527	0
7	7.847	1	0.568	0
8	16.237	2	32.421	0
9	33.598	2	50.929	36.355
10	69.519	2	58.175	50.032
11	143.845	2	62.110	25.919
12	297.635	3	36.658	94.655
13	615.848	3	78.888	35.231
14	1274.3	2	95.464	57.330
15	2,636.7	2	98.011	76.877
16	5,455.6	2	99.033	99.389
17	11,288	2	99.363	99.419
18	23,357	2	99.507	99.427
19	48,329	2	99.567	99.432
20	100,000	2	99.586	99.438

Figure 4-13: Arbitrary experiment - Left plot: Singular values from both, the system found by N2PBSID and its reweighted iterations. Right table: VAF values obtain by N2PBSID and its reweighted iterations.

As it can be seen, the singular values are not moved significantly to observe a notably change in the gap between the dominant and the weak singular values. Furthermore, the iterations don't seem to give any advantage in the fitting accuracy. In the table found in Figure 4-13, several details from the N2PBSID algorithm and the reweighted iterations are presented. From left to right the columns presented are: the number of the test, the λ used for the optimization, the order that the algorithm picked as suitable for the system, the fitting accuracy form the N2PBSID algorithm and the fitting accuracy once the reweighted iterations are performed. In this table it can be seen that the iterations present three cases, sometimes the iteration help to get a more accurate result in the fitting value, clearly evident in test number 12, secondly, most of the times the iterations decrease the accuracy found first by the N2PBSID algorithm, clearly seen in test 13, 14 and 15, and in the third case the fitting value doesn't change significantly from the originally found, seen in test number 16 to 20.

test	λ	order	NN VAF	RewNN VAF
1	0.100	1	16.602	0
2	0.206	1	9.557	0
3	0.428	1	4.511	0
4	0.885	1	1.718	0
5	1.833	1	1.322	0
6	3.792	1	1.527	0
7	7.847	1	0.568	0
8	16.237	2	32.421	0
9	33.598	2	50.929	36.355
10	69.519	2	58.175	50.032
11	143.845	2	62.110	25.919
12	297.635	3	36.658	94.655
13	615.848	3	78.888	35.231
14	1274.3	2	95.464	57.330
15	2,636.7	2	98.011	76.877
16	5,455.6	2	99.033	99.389
17	11,288	2	99.363	99.419
18	23,357	2	99.507	99.427
19	48,329	2	99.567	99.432
20	100,000	2	99.586	99.438

In some particular cases the reweighted nuclear norm optimization seem to have desirable outcomes as shown in Figure 4-14. Were a more significant gap can be seen thanks to both, a higher position of the dominant singular values and a lower position of the weak singular values. Then again, the fitting accuracy behavior was the same as last example, were if the accuracy didn't decrease with the iterations, it didn't change significantly.

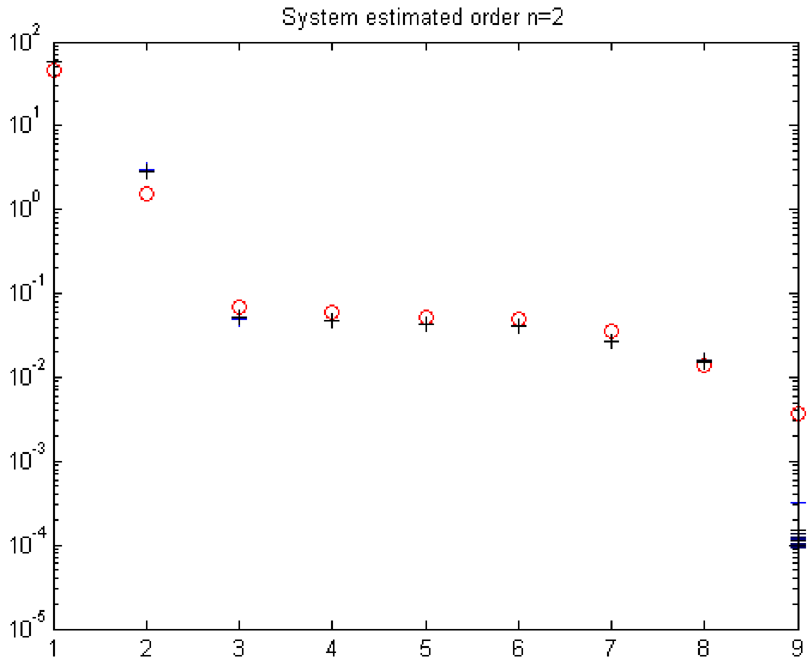
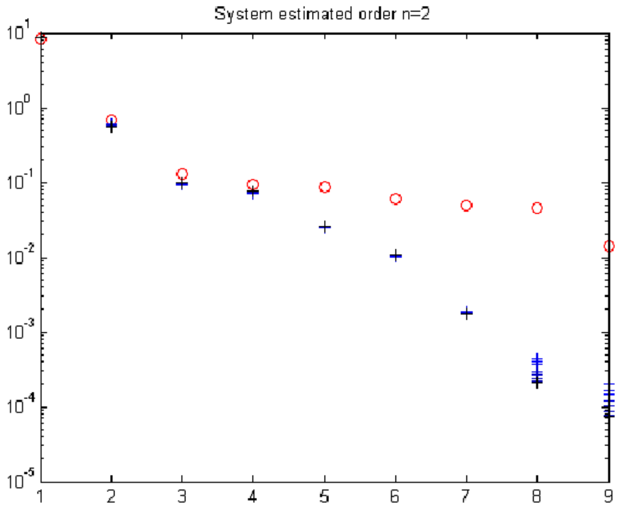


Figure 4-14: Arbitrary experiment - Singular values from both, the system found by N2PBSID and its reweighted iterations.

On the third case of results presented the singular values show a modification depicted in Figure 4-15, were the weak singular values decrease, but the gap between the dominant and the weak ones remains the same.



test	λ	order	NN VAF	RewNN VAF
1	0.100	0	0	0
2	0.206	0	0	0
3	0.428	0	0	0
4	0.885	1	0	0
5	1.833	1	0	0
6	3.792	1	0	0
7	7.847	1	0	0
8	16.237	2	0	0
9	33.598	2	0	1.316
10	69.519	3	3.129	4.968
11	143.845	3	3.7108	32.751
12	297.635	3	6.700	97.548
13	615.848	3	18.912	29.105
14	1274.3	3	66.284	99.425
15	2,636.7	2	95.999	99.842
16	5,455.6	2	99.463	99.786
17	11,288	2	99.907	99.783
18	23,357	2	99.988	99.782
19	48,329	2	99.973	99.782
20	100,000	2	99.955	99.784

Figure 4-15: Arbitrary experiment - Left plot: Singular values from both, the system found by N2PBSID and its reweighted iterations. Right table: VAF values obtain by N2PBSID and its reweighted iterations.

In this case the fitting accuracy seems to have in most of the cases better values in the iterations than in the original N2PBSID, seen from test 11, 12 14 and 15.

test	λ	order	NN VAF	RewNN VAF
1	0.100	0	0	0
2	0.206	0	0	0
3	0.428	0	0	0
4	0.885	1	0	0
5	1.833	1	0	0
6	3.792	1	0	0
7	7.847	1	0	0
8	16.237	2	0	0
9	33.598	2	0	1.316
10	69.519	3	3.129	4.968
11	143.845	3	3.7108	32.751
12	297.635	3	6.700	97.548
13	615.848	3	18.912	29.105
14	1274.3	3	66.284	99.425
15	2,636.7	2	95.999	99.842
16	5,455.6	2	99.463	99.786
17	11,288	2	99.907	99.783
18	23,357	2	99.988	99.782
19	48,329	2	99.973	99.782
20	100,000	2	99.955	99.784

The conclusion from this experiments was the same as in the original N2PBSID algorithm, the results seem to be perplexing. There is not clear advantage or disadvantage of the weighted iterations done of this system, probably due to the noise variations presented in the output.

DaISy experiments

From the last conclusions it was decided that testing this method on real-life data was going to clarify if this algorithm modification can bring an advantage from the original N2PBSID algorithm.

On this experiments it was decided to pick the best results (fitting accuracy) given from the last section to see if the weighted iterations could improve them. For this reason the first experiment (hair dryer #) is done with N$_{ide}$ =400 samples, the second experiment (steam heat exchange #) the experiment is done with N$_{ide}$ =750 samples and the third experiment is performed with N$_{ide}$ =300. For all the experiments the validation data used is the same described in the first section of this chapter from Table 4-3 and the λ ranges return to the original ones with 10^{-3} to 10^{3}.

Since in these experiments the focus is on the benefit that the reweighted iterations have on the singular values, the singular values plots that will be show display several markers. The red circle represents the original singular values found by the N2PBSID algorithm with the λ chosen, the plus sign (+) are the singular values after every iteration, were the more blue ones are the first iterations and the more black ones are the last iterations. Since the blue and black colors are sometimes difficult to distinguish and to follow, a red plus was placed in the last iteration.

Experiment 1: Hair dryer (data set 96-006)

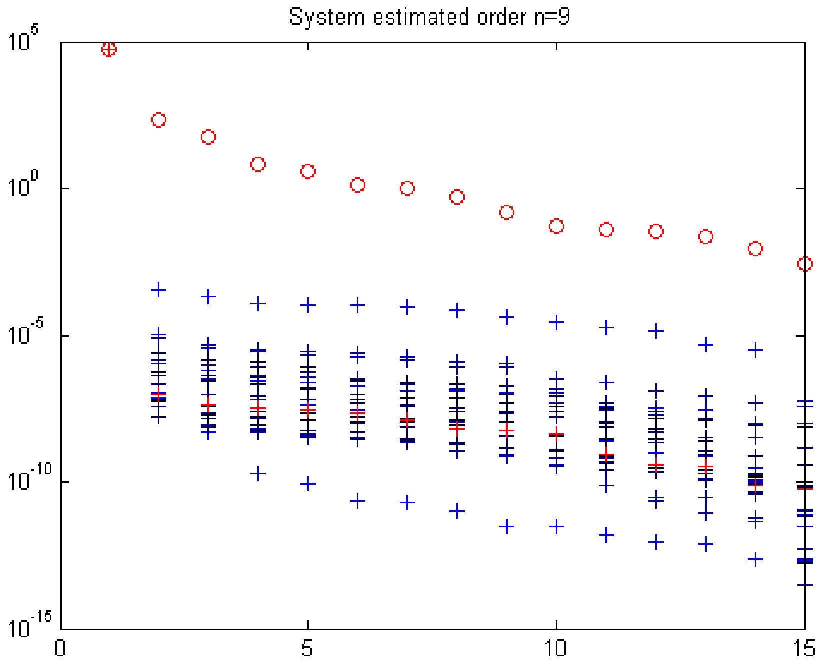


Figure 4-16: VAF and Order for small number of samples DaISy #2 - Steam hear exchanger

We can observe the singular values of the first experiment in Figure 4-16. Here we can see that the singular values change drastically thanks to the weighted iterations. It can be seen that the iterations made all but the first singular value to decrease. This result is not consistent with the one obtained by the N2PBSID, since it was detected a third order system form this data, while this reweighted iteration shows that a better selection would be a first order system. Not only that, it is interesting to see in Table 4-5 that the order selected by the N2PBSID that could get a stable system after the iterations started at order $n = 9$, which is not consistent with the singular values gap in Figure 4-16. Nonetheless the solution found by N2PBSID for a system of order $n = 9$ was used to calculate the matrices of a first order system to see if this system reflected a better or similar fitting accuracy than the third order system found by N2PBSID; but this was valueless since the fitting accuracy decrease to 66%. Not only this, the system with the $n = 9$ order was computed and the fitting from this system reached the 99%, which is the same accuracy reached with the original N2PBSID system with a much lower order ($n = 3$).

Experiment 2: Steam heat exchanger (data set 96-002)

The singular values of this experiment can be observe in Figure 4-17. Here it can be seen that all singular values move to a lower vertical position, but the gap between the dominant and weak singular values didn't increase, but decreased. What it seem to have been a third order system (which is the order found by the N2SID with the best fitting values in the first DaISy experiments) now is a system with really ambiguous singular values. Nonetheless, if the solution found is tested with the original fifth order ($n = 5$) selected by the N2PBSID the fitting value reaches 89% accuracy. Which is the highest value found by any of the methods for any N$_{ide}$ test, since the highest fitting value was found by N2SID with nearly 77% accuracy, which is a difference of more than 10 percentile points for a fifth order system (the one computed by N2SID was $n = 4$).

test	λ	order	NN VAF	RewNN VAF
1	0.001	1	66.699	0
2	0.002	2	77.488	0
3	0.004	3	90.543	0
4	0.008	4	98.315	0
5	0.018	4	98.418	0
6	0.037	3	99.017	0
7	0.078	3	99.147	0
8	0.162	4	99.197	0
9	0.335	5	99.165	0
10	0.695	6	99.191	0
11	1.438	7	99.227	0
12	2.976	9	99.227	99.056
13	6.158	10	99.215	99.044
14	12.743	13	99.211	99.421
15	26.367	13	99.195	99.166
16	54.556	13	99.193	99.137
17	112.883	13	99.198	99.114
18	233.572	13	99.218	99.229
19	483.293	13	99.229	99.226
20	1000	13	99.231	99.231

Table 4-5: VAF results for each λ for N2PBSID and then applying reweighted iterations # 1 - Hair dryer N$_{ide}$ =400

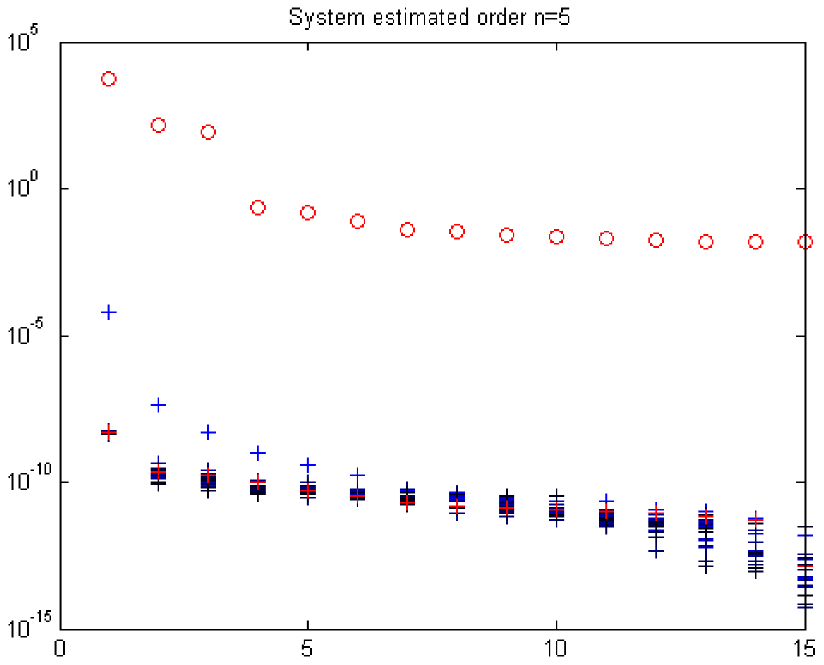


Figure 4-17: VAF and Order for small number of samples DaISy #2 - Steam hear exchanger

On Table 4-6 it can be seen that we can draw the same conclusion about the accuracy. The iterations not necessary improve the accuracy of the system, but in some cases it is possible, that is the case of the test number 5, which is the one chosen for the results presented.

Experiment 3: Robot arm (data set 96-009)

The singular values of the third experiment can be noticed in Figure 4-18. Here again a significant change can be observed in the singular values, from a system that seem to be between a second and fourth order system, the weighted iterations created a big gap between the second and third singular value (even when the N2PBSID selected a fourth order system); making the ambiguity in the original singular values disappear, making it suitable for a second order system. Then again this gap created by the weights doesn't mean that the second order system creates a better model, this was tested and the fitting accuracy decrease to 23% compared to the 75% obtained with the fifth order model (seen in Table 4-7). In spite of this, it is important to recalled from the first DaISy experiment that this particular example was already atypical. In this experiment the model order selection of the original experiment with the best VAF was 8, which didn't coincide with the gap of both the system from N2SID and N2PBSID methods. Making this an abnormal result.

test	λ	order	NN VAF	RewNN VAF
1	0.001	1	76.925	50.494
2	0.002	1	82.312	75.855
3	0.004	1	82.471	80.456
4	0.008	4	82.306	54.683
5	0.018	5	82.898	89.129
6	0.037	4	85.096	78.575
7	0.078	5	86.632	81.828
8	0.162	6	86.481	76.672
9	0.336	6	87.796	82.073
10	0.695	6	88.220	75.160
11	1.438	7	88.299	83.480
12	2.976	7	88.714	89.493
13	6.158	8	89.265	65.218
14	12.743	9	89.306	69.175
15	26.367	10	89.212	83.054
16	54.556	10	88.968	87.890
17	112.883	11	88.475	89.680
18	233.572	11	88.517	87.141
19	483.293	12	88.497	89.781
20	1000	14	89.413	90.019

Table 4-6: VAF results for each λ for N2PBSID and then applying reweighted iterations # 2 - Steam heat exchanger N$_{ide}$ =750

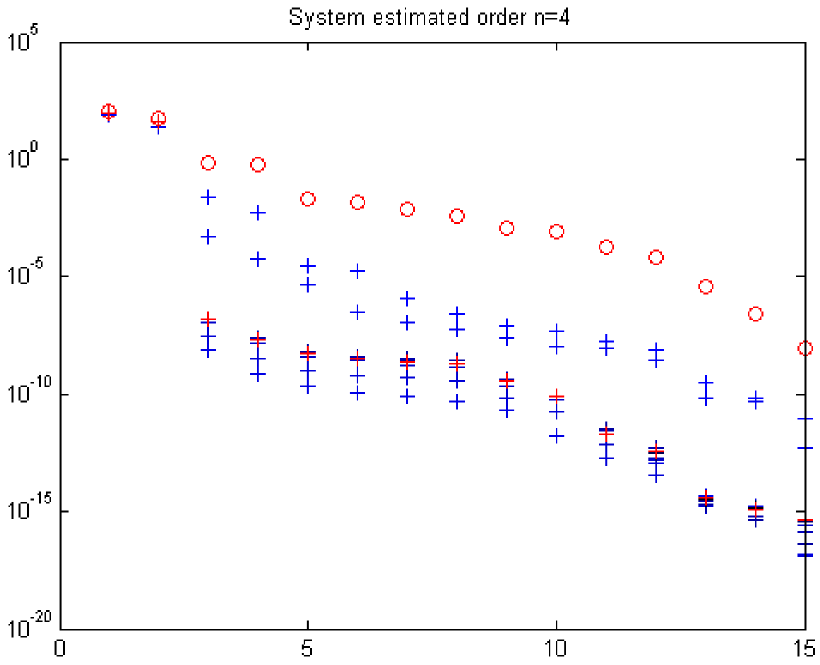


Figure 4-18: VAF and Order for small number of samples DaISy #2 - Steam hear exchanger

In this experiment the fitting accuracy decreased compared to the ones obtained from the original VAF in all λ cases as seen in Table 4-7.

4-2-1 Conclusions

As conclusions it can be seen on the first DaISy experiments that in 2 out of 3 cases the accuracy was superior than all the other 3 method compared. Not only it gave better accuracy, the model order was, if not the same, inferior to the one chosen by N4SID or N2SID, giving an advantage in the simplicity of the models found. Furthermore, the N2PBSID has an advantage in its computation time compared to the N2SID because in its iterations no Fourier transformation matrix needs to be calculated.

Independenly from the last benefits, the N2PBSID method created a bigger gap in the singular values than what N2SID could improved form the WNN$_{opt}$ method ([5], [12]), making it even more simple to identify the order of the system.

Certainly, we can also mention in this analysis the ability of the N2PBSID algorithm to handle small data batches because of its structural properties. Where the formulation of the matrices doesn't lead to any projections, avoiding any loss of information. On the contrary, methods like the N4SID and PBSID$_{opt}$ cannot perform effectively in this kind of instances. When tested on a defined transfer function it seem that it has an advantage over the N2SID method when the amount of data available is 3 times as much as the number of block rows. When having available only 2 times the number of block rows of data, it was observed that the N2SID method had a better performance. We wanted to verify this by using the DaISy datasets to test this results with real life data.

When using only data samples that were 2 times the number of block rows and the validation data

test	λ	order	NN VAF	RewNN VAF
1	0.001	0	0	0
2	0.002	0	0	0
3	0.004	0	0	0
4	0.008	0	0	0
5	0.018	0	0	0
6	0.037	1	0	0
7	0.078	2	0	0
8	0.162	2	0	0
9	0.336	2	8.526	0
10	0.695	2	30.189	0
11	1.438	3	81.278	0
12	2.976	3	80.494	0
13	6.158	3	83.705	0
14	12.742	3	82.559	0
15	26.366	5	83.767	0
16	54.555	5	72.984	0
17	112.883	4	89.745	75.249
18	233.572	5	80.228	72.569
19	483.293	6	88.743	56.187
20	1000	7	97.374	74.788

Table 4-7: VAF results for each λ for N2PBSID and then applying reweighted iterations # 3 - Robot arm N$_{ide}$ =300

being the same amount as defined in the first experiment the results were decisive. Both methods obtained good results, but the N2PBSID seem to be the favorable method when comparing accuracy and model complexity.

Finally, the reweighted nuclear norm brings an interesting perspective on how to choose the system's order. But this option seem to be efficient in reducing the singular values, but not necessarily in making the gap between the dominant and weak singular values more clear. In the same way, the results that did show a bigger gap between singular values didn't seem to give a better fitting results than the results obtained with the order chose by the regular N2PBSID algorithm. Moreover, not only the fitting results obtained were not better, but the iterations sometimes didn't converge to stable solutions, making this implementation not recommended for N2PBSID.

Chapter 5

Conclusions

This chapter gives an overview of the conclusions drawn in this thesis, and the recommendations for future work.

In this thesis we introduce both PEM and SID methods as our main focus for system identification purposes to a single optimization problem using PBSID$_{opt}$ theory. This with the purpose of having the benefits of the dual cost function, a good fitting criterion while maintaining a low order model. In addition, commonly an SID method is performed as a first step to be the initial estimate for the PEM to be performed secondly. In this algorithm we have the advantage that there is only one step involved for the user.

This new perspective to solve system identification methods by combining both families not only takes the advantages of each method, both opens new fields for new research and discoveries. In this case, this new approach consist in reformulating the $low\ rank\ matrix$ and the fitting cost function inspired by the PBSID$_{opt}$ algorithm.

This method was tested to see if searching for the Markov parameters using the N2PBSID method could bring any further benefits compared with the N2SID method with its structural analysis, or better fitting results than the N4SID or the classic PBSID$_{opt}$ algorithm.

From the results presented in Chapter 4, it can be concluded that the N2PBSID method is effective for system identification purposes.

We can mention that the advantages on N2PBSID as follow: in general it provides better accuracy results, in these cases a lower or equal model order is selected compared to all the other methods, the computational time to execute the method is lower than the N2SID method (which is the method that also perfoms iterations for its solution), it creates a bigger gap in the singular values for model order distinction and can handle small data batches in a suitable way.

Concluding that the algorithm gives what it promises in an conceptual way. It not only has a good trade-off between the complexity and accuracy of the system, its results in both aspects are noticeably better than the other methods already tested thanks to its structural properties.

Finally, the reweighted nuclear norm implementation didn't show the same benefits as promised. The method definitely changes the singular value positions, but this didn't necessarily create an advantage on the model order or fitting results. Moreover, the iterations sometimes gave unstable results, making this implementation not a satisfactory one for N2PBSID.

Overall we can conclude that the N2PBSID by its own is a good way to analyse system's data if both a large or a small amount of data is available. However, it is important to notice that even though the N2PBSID seem to have favorable results compared to the existing methods, the algorithm is not foolproof.

At first it was speculated that this algorithm was a simple method to implement in any system identification problem due to the fact that only the data for analysis and some user parameters are required to obtain the complete model of the system. The user parameters that are needed for this implementation are only the correct range of λs and the size of the past window p. Nonetheless, these are not the only specifications that this method needs for the correct use and analysis of the algorithm. An engineer is needed to tune certain important factors that crucial for its performance, such as the a vector change (as mentioned in Chapter 4), pick the adequate tolerance accuracy limits and select the data to identify. Likewise there are other elements that require to make an educated decision like selecting the model order once the algorithm finds a solution, see whether or not the data needs to be detrended, choose what kind of approach will be chosen to compute the system matrices, among other tuning details makes this algorithm complex enough not to be suitable for implement with an inexperienced user. That's why it is important to have a control engineer behind the implementation of this kind of algorithms that can comprehend how this kind of parameters can influence the results obtained. Debunking the idea that this is a simple method for implementation.

Finally, from the foregoing, or from this report in general, it may be concluded that this algorithm is, even if not the most straightforward easy, suitable for system identification purposes. Making this method a fresh view on how system identification algorithms are advancing and developing, always looking for easier, robust and more effective methods to solve our everyday system problems with the information available.

Appendix A

Linear mapping and adjoints

The ADMM algorithm requires to have a generic form equation of the nuclear norm and quadratic regularization term. In this appendix it will be described how the linear mapping and consequently the adjoint of the optimization variable is slightly changed due to the nature of the $PBSID_{opt}$ algorithm low rank nuclear norm matrix.

$$x = \begin{bmatrix} \Xi_1 & \Xi_2 & \Xi_3 & \Xi_4 & \cdots & \Xi_{p(l+r)} \end{bmatrix}'$$

$$A(x) = \begin{bmatrix} \Xi_1 & \Xi_2 & \Xi_3 & \Xi_4 & \cdots & \Xi_{p(l+r)-1} & \Xi_{p(l+r)} \\ 0 & 0 & \Xi_1 & \Xi_2 & \cdots & \Xi_{p(l+r)-3} & \Xi_{p(l+r)-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \Xi_1 & \Xi_2 \end{bmatrix}$$

$$B = \begin{bmatrix} b_1 & b_{(p\times l)+1} & b_{2(p\times l)+1} & b_{3(p\times l)+1} & \cdots & b_{(p\times l)(p(l+r)-2)+1} & b_{(p\times l)(p(l+r)-1)+1} \\ b_2 & b_{(p\times l)+2} & b_{2(p\times l)+2} & b_{3(p\times l)+2} & \cdots & b_{(p\times l)(p(l+r)-2)+2} & b_{(p\times l)(p(l+r)-1)+2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{p\times l} & b_{2(p\times l)} & b_{3(p\times l)} & b_{4(p\times l)} & \cdots & b_{(p\times l)(p(l+r)-1)} & b_{(p\times l)(p(l+r))} \end{bmatrix}$$

$$\langle A(x), B\rangle \tag{A-1}$$

$$\left\langle \begin{bmatrix} \Xi_1 & \Xi_2 & \Xi_3 & \cdots & \Xi_{p(l+r)} \\ 0 & 0 & \Xi_1 & \cdots & \Xi_{p(l+r)-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Xi_2 \end{bmatrix}, \begin{bmatrix} b_1 & b_{(p\times l)+1} & \cdots & b_{(p\times l)(p(l+r)-1)+1} \\ b_2 & b_{(p\times l)+2} & \cdots & b_{(p\times l)(p(l+r)-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p\times l} & b_{2(p\times l)} & \cdots & b_{(p\times l)(p(l+r))} \end{bmatrix} \right\rangle$$

$$\langle A(x), B\rangle = tr(B^* A(x)) \tag{A-2}$$

$$tr\left(\begin{bmatrix} b_1 & b_2 & \cdots & b_{p\times l} \\ b_{(p\times l)+1} & b_{(p\times l)+2} & \cdots & b_{2(p\times l)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(p\times l)(p(l+r)-1)+1} & b_{(p\times l)(p(l+r)-1)+2} & \cdots & b_{(p\times l)(p(l+r))} \end{bmatrix} \begin{bmatrix} \Xi_1 & \Xi_2 & \Xi_3 & \cdots & \Xi_{p(l+r)} \\ 0 & 0 & \Xi_1 & \cdots & \Xi_{p(l+r)-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Xi_2 \end{bmatrix}\right)$$

$tr(B^* A(x)) =$

$$\begin{array}{llllllr} b_1\Xi_1 & +b_{(p\times l)+1}\Xi_2 & +b_{2(p\times l)+1}\Xi_3 & +b_{3(p\times l)+1}\Xi_4 & +b_{4(p\times l)+1}\Xi_5 & \cdots & +b_{(p\times l)(p(l+r)-1)+1}\Xi_{p(l+r)} \\ & & +b_{2(p\times l)+2}\Xi_1 & +b_{3(p\times l)+2}\Xi_2 & +b_{4(p\times l)+2}\Xi_3 & \cdots & +b_{(p\times l)(p(l+r)-1)+2}\Xi_{p(l+r)-2} \\ & & & & +b_{4(p\times l)+3}\Xi_1 & \cdots & +b_{(p\times l)(p(l+r)-1)+3}\Xi_{p(l+r)-4} \\ & & & & & \ddots & \vdots \\ & & & & & & +b_{(p\times l)(p(l+r))}\Xi_2 \end{array}$$

$$\langle A(x), B\rangle = \langle \mathscr{A}(B), x\rangle \tag{A-3}$$

It is easy to verify that the adjoint of B is given by summing upwards the diagonals of B

$$\left\langle \begin{bmatrix} b_1 & +b_{2(p\times l)+2} & +b_{4(p\times l)+3} & \cdots & +b_{(p\times l)(p(l+r)-1)} \\ b_{(p\times l)+1} & +b_{3(p\times l)+2} & +b_{5(p\times l)+3} & \cdots & +b_{(p\times l)(p(l+r))} \\ & & b_{2(p\times l)+1} & \cdots & +b_{(p\times l)(p(l+r)-1)-1} \\ & & b_{3(p\times l)+1} & \cdots & b_{(p\times l)(p(l+r))-1} \\ & & & \ddots & \vdots \\ & & & & b_{(p\times l)(p(l+r)-2)+1} \\ & & & & b_{(p\times l)(p(l+r)-1)+1} \end{bmatrix}, \begin{bmatrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \\ \Xi_4 \\ \vdots \\ \Xi_{p(l+r)-1} \\ \Xi_{p(l+r)} \end{bmatrix} \right\rangle \tag{A-4}$$

In the case of the $PBSID_{opt}$ algorithm the mapping of our optimization variable converts into ΓK ($A(x) = \Gamma K$), which is not the complete low rank matrix $\Gamma K Z$. This new term will add a variable in our regular mapping, which will affect our adjoint term.

If we apply the same theory,

$$\langle A(x) \times Z, B\rangle$$

$$\left\langle \begin{bmatrix} \Xi_1 & \Xi_2 & \Xi_3 & \cdots & \Xi_{p(l+r)} \\ 0 & 0 & \Xi_1 & \cdots & \Xi_{p(l+r)-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \Xi_2 \end{bmatrix} \overbrace{\left[\begin{array}{c|c|c|c} z_1 & z_{p(l+r)+1} & \cdots & z_{(p(l+r))^2-p(l+r)+1} \\ z_2 & z_{p(l+r)+2} & \cdots & z_{(p(l+r))^2-p(l+r)+2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{p(l+r)} & z_{2p(l+r)} & \cdots & z_{(p(l+r))^2} \end{array}\right]}^{\left[\begin{array}{c|c|c|c} Z_1 & Z_2 & \cdots & Z_{p(l+r)} \end{array}\right]}, \begin{bmatrix} b_1 & b_{(p\times l)+1} & \cdots & b_{(p\times l)(p(l+r)-1)+1} \\ b_2 & b_{(p\times l)+2} & \cdots & b_{(p\times l)(p(l+r)-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p\times l} & b_{2(p\times l)} & \cdots & b_{(p\times l)(p(l+r))} \end{bmatrix} \right\rangle$$

From here it is easy to verify that $\mathscr{A}(BZ)$ is given by rearranging the structure of the columns of the Z matrix and to make a vector out of the columns of the B matrix, as shown in Eq. (A)

$$\langle A(x)Z, B\rangle = \langle \mathscr{A}(BZ), x\rangle \tag{A-5}$$

$$\left\langle \begin{bmatrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \\ \Xi_4 \\ \vdots \\ \Xi_{p(l+r)-3} \\ \Xi_{p(l+r)-2} \\ \Xi_{p(l+r)-1} \\ \Xi_{p(l+r)} \end{bmatrix}, \begin{bmatrix} \mathscr{Z} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_{(p\times l)(p(l+r))-3} \\ b_{(p\times l)(p(l+r))-2} \\ b_{(p\times l)(p(l+r))-1} \\ b_{(p\times l)(p(l+r))} \end{bmatrix} \right\rangle$$

where

$$\overbrace{\left[\begin{array}{cccc|cccc|c|cccc}
z_1 & z_3 & \cdots & z_{p(l+r)-1} & z_{p(l+r)+1} & z_{p(l+r)+3} & \cdots & z_{2p(l+r)-1} & \cdots & z_{(p(l+r))^2-p(l-r)+1} & z_{(p(l+r))^2-p(l-r)+3} & \cdots & z_{(p(l+r))^2-1} \\
z_2 & z_4 & \cdots & z_{p(l+r)} & z_{p(l+r)+2} & z_{p(l+r)+4} & \cdots & z_{2p(l+r)} & \cdots & z_{(p(l+r))^2-p(l-r)+2} & z_{(p(l+r))^2-p(l-r)+4} & \cdots & z_{(p(l+r))^2} \\
z_3 & z_5 & \cdots & 0 & z_{p(l+r)+3} & z_{p(l+r)+5} & \cdots & 0 & \cdots & z_{(p(l+r))^2-p(l-r)+3} & z_{(p(l+r))^2-p(l-r)+5} & \cdots & 0 \\
z_4 & z_6 & \cdots & 0 & z_{p(l+r)+4} & z_{p(l+r)+6} & \cdots & 0 & \cdots & z_{(p(l+r))^2-p(l-r)+4} & z_{(p(l+r))^2-p(l-r)+6} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\
z_{p(l+r)-3} & z_{p(l+r)-1} & \cdots & 0 & z_{2p(l+r)-3} & z_{2p(l+r)-1} & \cdots & 0 & \cdots & z_{(p(l+r))^2-3} & z_{(p(l+r))^2-1} & \cdots & 0 \\
z_{p(l+r)-2} & z_{p(l+r)} & \cdots & 0 & z_{2p(l+r)-2} & z_{2p(l+r)} & \cdots & 0 & \cdots & z_{(p(l+r))^2-2} & z_{(p(l+r))^2} & \cdots & 0 \\
z_{p(l+r)-1} & 0 & \cdots & 0 & z_{2p(l+r)-1} & 0 & \cdots & 0 & \cdots & z_{(p(l+r))^2-1} & 0 & \cdots & 0 \\
z_{p(l+r)} & 0 & \cdots & 0 & z_{2p(l+r)} & 0 & \cdots & 0 & \cdots & z_{(p(l+r))^2} & 0 & \cdots & 0
\end{array}\right]}^{\mathscr{Z}\text{ is made out of}\left[\; Z_1 \;\middle|\; Z_2 \;\middle|\; \cdots \;\middle|\; Z_{p(l+r)} \;\right]}$$

Appendix B

Matlab implementation

```
clear all; clc; close all

% Parameters
Min_PastWindow=9;
Max_PastWindow=0;
Num_Periods=10;
Min_lambda=4;
Max_lambda=20;
Lambda_Points=20;

% Defining system
Sim.P       = Min_PastWindow;   % period (only required for periodic scheduling)
Sim.np      = Num_Periods;      % number of periods (only required for periodic scheduling)
Sim.N       = Sim.np*Sim.P;     % number of data points
Sim.noise2  = 1;                % Noise on state
Sim.SNR     = 80;               % Noise level
Sim.Monte   = 1;                % Number of monte carlo simulations

% Creating system
Sim.l=1;
Sim.r=1;
Sim.n=2;
sys=tf([1],[1 2*0.01 1]); sys=c2d(sys,0.05);
[Sim.A,Sim.B,Sim.C,Sim.D]=ssdata(sys);
[sys1.A,sys1.B,sys1.C,sys1.D]=ssdata(sys);
sys=ss(sys1.A,sys1.B,sys1.C,sys1.D,0.05);

loops=20;                       % number of iterations
for g=1:loops

    % Simulation of the system
    u=randn(Sim.N,Sim.r);
    uv=randn(Sim.N,Sim.r);
    x0 = zeros(Sim.n,1);
    [yn,x] = lsim(sys,u);
    vary = sqrt(var(yn).*10^(-Sim.SNR/20));
    e=Sim.noise2*(diag(vary)*randn(Sim.l,Sim.N))';

    % Time sequences
    t=1:0.05:size(u,1)*0.05+1-.05;
    t2=1:0.05:size(uv,1)*0.05+1-.05;

    % Outputs
    y=yn+e;
    y_val=lsim(sys,uv,t2);

    % Stack data vector
    z=[u y]';
    l=size(y,2);        % # of outputs
    N=size(y,1);        % sample size
    r=size(u,2);        % # of inputs
    lr=l+r;             % # of inputs + # of outputs

    if Max_PastWindow==0
        Max_PastWindow=Min_PastWindow;
    end

    for h=Min_PastWindow:Max_PastWindow

        p=h;
        % Stacked data Hankel matrix
        ZZ=zeros(p*lr,N-p);
        for i=1:1:p
            ZZ((i-1)*size(z,1)+1:(i)*size(z,1),:)=z(:,i:end-p+i-1);
        end

        % Output matrix of the rest of the data (p+1:end)
        Y=y(p+1:end,:)';

        % Computing M
        [M]=M_matrix(ZZ,p,l,r);

        %% ADMM algorithm

        % Initialization
        lambda=logspace(Min_lambda,Max_lambda,Lambda_Points);        % logspace(a,b
            ,n) n points between 1e^a and 1e^b

        for k=1:size(lambda,2)

            x=transpose([zeros(size(Y*ZZ'))]);
            a=inv(ZZ*ZZ')*ZZ*transpose(Y);

            H_2norm=lambda(k)*(ZZ*transpose(ZZ)*ZZ*transpose(ZZ));

            A0=[zeros(l*p,size(ZZ',2))];
            X=A0;
            Z=[zeros(l*p,size(ZZ',2))];
            rho=1;

            %% Algorithm

            % Stopping criterion
            eps_rel=10^(-1);
            eps_abs=10^(-6);

            [order,VAF3,x,X,varx]=algorithm(x,a,H_2norm,A0,X,Z,rho,M,Y,p,l,lr,ZZ,
                eps_rel,eps_abs);

            NN(k,:)=order
            FC5(k,:)=VAF3;
            CKs(:,:,k)=x;
            Xs(:,:,k)=X;

            GammaK1=Operator(x,p,l,lr);

            % Computing the system matrices
            [U1,S1,V1]=svd(GammaK1*ZZ);
            state_vec1=S1(1:order,1:order)*V1(:,1:order)';

            try
                [Ae_new,Be_new,Ce_new,De_new,Ke_new] = dx2abcdk(state_vec1,u,y,p,p)
                    ;
                % Simulation
                [y_new,x_new] = dlsim(Ae_new,Be_new,Ce_new,De_new,u);
                VAF_new(k,:)= mean(max(vaf(y,y_new),0));
            catch
                try
                    [Ae_new,Be_new,Ce_new,De_new,Ke_new] = dx2abcdk(state_vec1,u,y,
                        p,p,'stable1');
                    % Simulation
                    [y_new,x_new] = dlsim(Ae_new,Be_new,Ce_new,De_new,u);
                    VAF_new(k,:)= mean(max(vaf(y,y_new),0));
                catch
                    VAF_new(k,:)=0;
                end
            end

        end

        answer=2;

        while (answer==2)

            % Selecting best choice for system
            Lambda = [1:size(lambda,2)]';

            T = table(Lambda,NN,VAF_new)
```

```

I = input ('Enter the lambda number that its more suitable for implementation: ');

%% Computing system matrices

% Choosing optimal variable 'x'
q=size(Y,1);
GammaK2=zeros(q*p,(l+r)*p);
CK2=CKs(:,:,I)';
NN_order=NN(I,:);

for i=1:p
    GammaK2(1+(q)*(i-1):i*q,1+(l+r)*(i-1):(l+r)*p)=CK2(:,1:(l+r)*(p+1-i));
end

% Computing the system matrices
[U2,S2,V2] = svd(GammaK2*ZZ);
state_vec2=S2(1:NN_order,1:NN_order)*V2(:,1:NN_order)';
[Ae_newADMM,Be_newADMM,Ce_newADMM,De_newADMM] = dx2abcd(state_vec2,u,y,p,p);

% Simulation
[y_newADMM,x_newf] = dlsim(Ae_newADMM,Be_newADMM,Ce_newADMM,De_newADMM,u);     VAF_NEW=vaf(y,y_newADMM);
[y_new_val,x_new_val] = dlsim(Ae_newADMM,Be_newADMM,Ce_newADMM,De_newADMM,uv);  VAF_NEW_val=max(vaf(y_val,y_new_val),0);

% PBSID/SVD approach
Gammahat = U2(:,1:NN_order);
S        = pinv(Gammahat);
Up = u(p+1:end-1,:)';
Zp = ZZ(:,1:end-1); Zf=ZZ(:,2:end);
Zpi    =    ZZ'*inv(ZZ*ZZ')*ZZ;
Zor    =    eye(N-p)-Zpi;
% Estimate of the state
Xe   =   S*GammaK2*ZZ;
Xep  =   S*GammaK2*Zp;
Xef  =   S*GammaK2*Zf;
% Computing the system matrices
Ee     =   Y-CK2*ZZ;
Eep    =   Ee(:,1:end-1);
Ce2    =   (Y-Ee)*Xe'*inv(Xe*Xe');
ABKhat =   Xef*pinv([Xep; Up; Eep]);
Ae2    =   ABKhat(:,1:NN_order);
Be2    =   ABKhat(:,NN_order+1:NN_order+r);
Ke2    =   ABKhat(:,NN_order+r+1:NN_order+r+l);

%Simulations
y_NN_PBSID=lsim(ss(Ae2,Be2,Ce2,0,0.05),u,t);        VAF_NN_PBSID=vaf(y,y_NN_PBSID);
```

```
            y_NN_PBSID_val=lsim(ss(Ae2,Be2,Ce2,0,0.05),uv,t2);  VAF_NN_PBSID_val=vaf(y_val,y_NN_PBSID_val);

            VAF                = {'Original'           ;   'Validation'                  };
            NEW_ADMM_PBSID     = [ vaf(y,y_NN_PBSID)   ;   vaf(y_val,y_NN_PBSID_val)   ];
            NEW_ADMM_DX2       = [ vaf(y,y_newADMM)    ;   vaf(y_val,y_new_val)        ];

            T = table(VAF,PBSID,NEW_ADMM_PBSID,NEW_ADMM_DX2,N2SID)

            answer = input ('Are you happy with the lambda you chose? 1/2 (Yes/No): ');

            if answer==2
                close all
            end
        end
    end

    Final_N2PBSID(g,I)=max([vaf(y_val,y_NN_PBSID_val),vaf(y_val,y_new_val)]);

end

%% Plots

% Simulation of output with validation data

figure
plot(t,y_val,'k'); hold on
if I==1
    plot(t,y_NN_PBSID_val,'c-.');
else
    plot(t,y_new_val,'c-.');
end
legend('Real system','N2PBSID')
title(['Validation fit using different input with past window ',num2str(p)])
```

Bibliography

[1] L. Ljung, ed., *System Identification (2Nd Ed.): Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[2] M. Verhaegen and V. Verdult, “Filtering and identification: A least squares approach.” Cambridge University Press, 2007.

[3] M. Fazel, *Matrix Rank Minimization with Applications*. PhD thesis, Standford University, 2002.

[4] Z. Liu, A. Hansson, and L. Vandenberghe, “Nuclear norm system identification with missing inputs and outputs,” *Systems & Control Letters*, vol. 62, pp. 605–612, April 2013.

[5] M. Verhaegen and A. Hansson, “N2sid: Nuclear norm subspace identification,” *CoRR*, January 2015.

[6] D. Sadigh, H. Ohlsson, S. Seshia, and S. Sastry, “Robust subspace system identification via weighted nuclear norm optimization,” in *The International Federation of Automatic Control* (B. Edward, ed.), (Cape Town, South Africa), pp. 9510–9515, August 2014.

[7] E. Canès and B. Recht, “Exact matrix completion via convex optimization,” May 2008.

[8] K. Mohan and M. Fazel, “Reweighted nuclear norm minimization with application to system identification,” *Proc. American Control Conference (ACC)*, 2010.

[9] Z. Liu and L. Vandenberghe, “Interior-point method for nuclear norm approximation with application to system identification,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, p. 1235âĂŞ1256, 2009.

[10] G. van der Veen, J. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen, “Closed-loop subspace identification methods: an overview,” *IET Control Theory Applications*, vol. 7, p. 1339 âĂŞ 1358, July 2013.

[11] J. Cai, E. Candes, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *Arxiv preprint arXiv:0810.3286*, October 2008.

[12] A. Hansson, Z. Liu, and L. Vandenberghe, "Subspace System Identification via Weighted Nuclear Norm Optimization," in *IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 3439–3444, IEEE; 1998, 2012.

[13] B. De Moor, P. De Gersem, B. De Shutter, and W. Favoreel, "Daisy: Database for the identification of systems," *Journal A*, vol. 38, no. 3, pp. 4–5, 1997.

[14] K. Hu, P. Ivanov, Z. Chen, P. Carpena, and E. Stanley, "Effect of trends on detrended fluctuation analysis," *PHYSICAL REVIEW E*, vol. 64, June 2001.

[15] L. Ljung, *System Identification Toolbox: User's guide.* MathWorks Incorporated, 1993.