Time-Domain System Identification of Low-Order Models for Flexible Spacecraft

by

Robert John Bauer

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto

© Copyright by Robert John Bauer 1998
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-35107-6
Time-Domain System Identification of Low-Order Models for Flexible Spacecraft
Degree of Doctor of Philosophy, 1998
Robert John Bauer
Graduate Department of Aerospace Science and Engineering
University of Toronto

Abstract

System Identification (SI) is the process of developing or improving a mathematical representation of a physical system using experimental data. Accurate SI is often a precursor to sophisticated control algorithms which assume that the 'plant' to be controlled is known. To achieve accurate models, many of the best current SI methods require that the size of the identified state-space model be significantly larger than the expected system size, a process called overspecification. Large models are impractical for model-based controller design and create numerical difficulties during the SI process. Low order and high accuracy are two conflicting requirements for SI.

Appropriate SI methods for on-orbit modeling of lightly-damped flexible spacecraft are established, including methods such as OKID with ERA, Q-
Markov CovER, ORSE and Subspace. Tests on Daisy, a flexible spacecraft emulator, demonstrate that these methods exhibit the overspecification problem.

To investigate SI of low-order models, model reduction techniques are employed. Balanced model reduction offers promising results for stable models. Since model stability is not guaranteed by many SI methods, three new approaches to balanced model reduction are derived and tested when identified models are unstable.

A new identification approach using OKID and cubic smoothing splines is presented, allowing low-order highly-accurate models to be directly identified. Avoiding impractically large models reduces computational requirements and potential for numerical problems.

Augmented SI is an approach that allows existing linear system identification techniques to better model non-idealities such as nonlinear friction. Augmented and standard SI experiments demonstrate that the linear system assumption made throughout this thesis is appropriate for Daisy.
Acknowledgments

I wish to express my gratitude to my thesis supervisor, Dr. P. C. Hughes of the University of Toronto’s Institute for Aerospace Studies, for his continuous support and guidance. I would also like to thank Dr. J. S. Hansen and Dr. K. A. Carroll for their advice as members of my Doctoral Advisory Committee.

I am grateful to Tony Hong for introducing me to the Daisy facility and Dr. R. E. Zee for the use of his Daisy computer simulator.

I would like to thank my parents for their advice and encouragement throughout my academic career.
This work is dedicated

to Georgeta
3.2 ERA .................................................. 29
3.3 Q-Markov CovER ............................... 42
3.4 ORSE ................................................. 52
3.5 Subspace ........................................... 59

4 ERA and Variations 67
  4.1 Consecutive and Concurrent Approach ............... 67
    4.1.1 Daisy Simulations ............................. 70
    4.1.2 Daisy Experiments ............................ 75

5 High-Order Identified Models 95
  5.1 Model Order Reduction .............................. 96
    5.1.1 Singular Value Decomposition ................. 96
    5.1.2 Frequency-based Model Reduction .............. 97
    5.1.3 Balanced Model Reduction .................... 98
    5.1.4 Model Order Reduction Comparison ........... 100
  5.2 Unstable Models .................................. 102
    5.2.1 Background .................................. 103
    5.2.2 Sources of Model Instability ................. 104
    5.2.3 Stabilize the Model .......................... 107
    5.2.4 Stabilize then Destabilize Model ............. 109
    5.2.5 Partition Into Stable and Unstable Portions ... 111
  5.3 Daisy Simulation Results .......................... 112
    5.3.1 Stable Model Realization .................... 113
    5.3.2 Unstable Model Realization ................... 116
  5.4 Daisy Experimental Results ........................ 118
D.1 ERA/DC .................................................. 214
D.2 Fast ERA .................................................. 221

E SI Software ................................................. 229
   E.1 Running the Software .................................... 229
   E.2 Software Modes .......................................... 232
   E.3 Menus ..................................................... 235
   E.4 Software Directories and Partitions .................. 238
   E.5 Modal Parameters ........................................ 241
   E.6 Mode Removal ........................................... 242

F Cubic Smoothing Spline Results ......................... 244
List of Tables

2.1 Selection Summary .............................................. 11

4.1 Computation Comparison ........................................ 73

5.1 Modal Parameters for Two-Mass Mechanical System ........ 106
5.2 Discrete-Time Eigenvalues: Two-Mass Mechanical System . 106
5.3 Modal Parameters from Daisy Simulation ..................... 114
5.4 Discrete-Time Eigenvalues for Daisy Simulation ........... 115
5.5 Continuous-Time Eigenvalues for Daisy Simulation .......... 117

6.1 Daisy Rib Frequencies ............................................ 131
List of Figures

1.1 System Identification .......................... 2
1.2 The Daisy Facility ............................. 6
1.3 The Daisy Facility .............................. 7
1.4 Dynamics, Identification and Control Experiment: DICE ... 9

2.1 System Identification field ....................... 12
2.2 Control-Model Identification Categories .............. 15
2.3 A linear time-invariant system .................... 17

3.1 Relationships between selected SI algorithms ............ 21
3.2 Daisy Observer Markov Parameters \[\bar{Y}_{k}^{(1)} - \bar{Y}_{k}^{(2)}\] ............... 26
3.3 Daisy System Markov Parameters \(Y_{k}\) .................. 27
3.4 ERA Singular Values .............................. 32
3.5 ERA Results for Daisy’s Hub ..................... 36
3.6 ERA Results for Ribs 1 and 2 ...................... 37
3.7 ERA Results for Ribs 3 and 4 ...................... 38
3.8 ERA Results for Ribs 5 and 6 ...................... 39
3.9 ERA Results for Ribs 7 and 8 ...................... 40
3.10 ERA Results for Ribs 9 and 10 .................... 41
3.11 Two Mass System ........................................... 43
3.12 Markov and covariance parameters using stochastic approach . 45
3.13 Prediction errors using Q-Markov CovER (stochastic) ........ 46
3.14 Covariance parameters calculated using deterministic approach 46
3.15 Prediction errors using Q-Markov CovER (deterministic) ...... 47
3.16 Prediction Errors for Q-Markov CovER and ERA ............. 51
3.17 ORSE Singular Values ....................................... 54
3.18 Prediction Errors for ORSE and ERA .......................... 56
3.19 Orthogonal Projection in two-dimensional space ............... 60
3.20 Oblique Projection in two-dimensional space .................. 60
3.21 Prediction Errors for Subspace and ORSE .................... 66

4.1 Relationships between ERA variations .......................... 68
4.2 OKID with ERA .................................................. 69
4.3 Effects of Noise on SI ........................................... 71
4.4 Sample Response: Fast ERA, signal-to-noise ratio = 50 ....... 72
4.5 Computations for ERA and variations for different observer orders ......................................................... 74
4.6 Decreasing SI Experiment Time .................................. 77
4.7 Combining Multiple SI Experiments ............................. 80
4.8 Effect of observer order $p$ on model prediction error ........ 81
4.9 Daisy hub results for $p = 4$ and $p = 15$ ..................... 83
4.10 Daisy hub results for $p = 4$ and $p = 15$ ..................... 84
4.11 Daisy rib 1 results for $p = 4$ and $p = 15$ .................. 85
4.12 Daisy rib 2 results for $p = 4$ and $p = 15$ .................. 86
4.13 Daisy rib 3 results for $p = 4$ and $p = 15$ .................. 87
4.14 Daisy rib 4 results for \( p = 4 \) and \( p = 15 \) .................................. 88
4.15 Daisy rib 5 results for \( p = 4 \) and \( p = 15 \) .................................. 89
4.16 Daisy rib 6 results for \( p = 4 \) and \( p = 15 \) .................................. 90
4.17 Daisy rib 7 results for \( p = 4 \) and \( p = 15 \) .................................. 91
4.18 Daisy rib 8 results for \( p = 4 \) and \( p = 15 \) .................................. 92
4.19 Daisy rib 9 results for \( p = 4 \) and \( p = 15 \) .................................. 93
4.20 Daisy rib 10 results for \( p = 4 \) and \( p = 15 \) .................................. 94

5.1 Fast ERA Singular Values ................................................................. 97
5.2 Fast ERA Identified Frequencies ......................................................... 98
5.3 Spectral Density of Daisy’s Measurements ........................................... 99
5.4 Model Order Reduction Comparison .................................................... 100
5.5 Stability Region for Continuous-Time and Discrete-Time Systems ............... 104
5.6 Two-Mass-Spring-Damper System Exhibiting Rigid Modes ......................... 105
5.7 Stabilizing an eigenvalue outside the stability boundary ......................... 108
5.8 Partition Into Stable and Unstable Portions ......................................... 112
5.9 Prediction Error Comparison: Truncation of High Frequencies ..................... 120
5.10 Prediction Error Comparison: Unstable Model Order Reduction .................. 121
5.11 Sample of Unstable Model Results .................................................... 123

6.1 OKID Markov Parameters for Daisy’s Hub ............................................. 127
6.2 OKID Markov Parameters for Daisy’s Ribs, Impulse at roll RW .................. 128
6.3 OKID Markov Parameters for Daisy’s Ribs, Impulse at pitch RW .................. 129
6.4 OKID Markov Parameters for Daisy’s Ribs, Impulse at yaw RW .................. 130
F.4  Directly identifying models of order $n = 46$: rib 1 OC . . . . . 248
F.5  Directly identifying models of order $n = 46$: rib 1 IC . . . . . 249
F.6  Directly identifying models of order $n = 46$: rib 2 OC . . . . . 250
F.7  Directly identifying models of order $n = 46$: rib 2 IC . . . . . 251
F.8  Directly identifying models of order $n = 46$: rib 3 OC . . . . . 252
F.9  Directly identifying models of order $n = 46$: rib 3 IC . . . . . 253
F.10 Directly identifying models of order $n = 46$: rib 4 OC . . . . . 254
F.11 Directly identifying models of order $n = 46$: rib 4 IC . . . . . 255
F.12 Directly identifying models of order $n = 46$: rib 5 OC . . . . . 256
F.13 Directly identifying models of order $n = 46$: rib 5 IC . . . . . 257
F.14 Directly identifying models of order $n = 46$: rib 6 OC . . . . . 258
F.15 Directly identifying models of order $n = 46$: rib 6 IC . . . . . 259
F.16 Directly identifying models of order $n = 46$: rib 7 OC . . . . . 260
F.17 Directly identifying models of order $n = 46$: rib 7 IC . . . . . 261
F.18 Directly identifying models of order $n = 46$: rib 8 OC . . . . . 262
F.19 Directly identifying models of order $n = 46$: rib 8 IC . . . . . 263
F.20 Directly identifying models of order $n = 46$: rib 9 OC . . . . . 264
F.21 Directly identifying models of order $n = 46$: rib 9 IC . . . . . 265
F.22 Directly identifying models of order $n = 46$: rib 10 OC . . . . . 266
F.23 Directly identifying models of order $n = 46$: rib 10 IC . . . . . 267
List of Symbols

\( \mathbb{R} \) Field of real numbers

\( A \in \mathbb{R}^{n \times n} \) state matrix

\( B \in \mathbb{R}^{n \times r} \) input influence matrix

\( C \in \mathbb{R}^{m \times n} \) output influence matrix

\( D \in \mathbb{R}^{m \times r} \) transmission matrix

\( D_q \) data matrix

\( E \) eigenmatrix of eigenvectors

\( G \in \mathbb{R}^{n \times m} \) observer gain matrix

\( H_j \in \mathbb{R}^{m \times dr} \) generalized Hankel matrix

\( \mathcal{H}_j \) block correlation Hankel matrix

\( I_i \) identity matrix of order \( i \)

\( O_i \) null matrix of order \( i \)

\( P_q \) observability matrix

\( \mathcal{P}_k \) combined system and observer gain Markov parameters

\( Q_d \) controllability matrix

\( R, S \) orthonormal matrices
\( \mathbf{R}_j^{hh} \) data correlation matrix

\( \mathbf{R}_k \) covariance parameters

\( \mathbf{S} \) balancing transformation matrix

\( \mathbf{T} \) similarity transformation matrix

\( \mathbf{U}_{0|2q-1} \) input block Hankel matrix

\( \mathbf{X}_k \) state sequence matrix

\( \mathbf{Y}_k \in \mathbb{R}^{m \times r} \) system Markov parameters

\( \mathbf{Y}_k \in \mathbb{R}^{m \times (r+m)} \) observer Markov parameters

\( \mathbf{Y}_k \in \mathbb{R}^{m \times m} \) observer gain Markov parameters

\( \mathbf{Y}_{0|2q-1} \) output block Hankel matrix

\( k \) enumerates the sample instants, \( k = 0, 1, 2, \ldots \)

\( l \) number of data samples

\( m \) number of outputs

\( n \) order of the system or the number of state variables

\( p \) observer order

\( r \) number of inputs

\( u \in \mathbb{R}^r \) input vector

\( x \in \mathbb{R}^n \) state vector
\( y \in \mathbb{R}^m \) output vector

\( \Delta \) stability perturbation matrix

\( \Lambda \) diagonal matrix of eigenvalues

\( \Pi_B \) operator that projects the row space of a matrix onto the row space of the matrix \( B \)

\( \Sigma \) diagonal matrix

\( \eta \) smoothing parameter

\( \lambda \) eigenvalue

\( \dagger \) denotes the pseudo-inverse
Chapter 1

Introduction

1.1 Motivation

Most modern spacecraft require active attitude control to accomplish their mission objectives. Unknown differences between the mathematical model used to design model-based controllers and the working system significantly limit the achievable performance of the closed-loop system. Juang [72] indicates that these errors often originate in finite element models. Error in finite element models may arise from:

- approximations in finite element derivations such as neglecting transverse shear deformation in Kirchhoff’s thin plate theory [26].

- differences between the actual material properties and those used in the finite element model; the former can also change in orbit.

- differences between actual material dimensions and those assumed in the finite element model.
poor modeling of characteristics such as damping [33].

Even the subtle differences introduced during manufacturing and assembly are enough to impair controller design. A more complete list of the sources of modeling uncertainty can be found in Denman et al. [39]. System Identification (SI) alleviates these modeling difficulties by synthesizing a model that fits the input/output measurements of the system (see Figure 1.1).

Flexible structures are inherently difficult to control and, therefore, SI of lightly-damped flexible space structures (FSS) is important. Two space shuttle Middeck 0-g Dynamics Experiments (MODE) and, more recently, a Middeck Active Control Experiment (MACE) have been conducted to study on-orbit SI and control [113, 114]. The Spacecraft Dynamics and Control group at the University of Toronto’s Institute for Aerospace Studies (UTIAS) researches different SI and control techniques using an FSS emulator called Daisy [55, 178]. UTIAS is currently developing Dynamics, Identification and
Control Experiments (DICE) [118]. These middeck experiments are planned for a 1999 flight of the space shuttle.

1.2 Unresolved Issues

The following unresolved issues exist:

- There is an immense number of SI methods. Ibáñez [60] has discussed the “bewildering number of [SI] methods” and he has characterized SI as a developing body of knowledge with myriad researchers, methods, ad hoc approaches, and heuristic reasoning. Denman et al. [39] support this conclusion and have indicated that, with new methods continuously being suggested, the SI field looks more like a “bag of tricks.”

- Although inventors of SI methods often claim superiority for their approach, each method has its own advantages and disadvantages. Denman et al. [39] point out that SI researchers claim their methods work well on both simulated and test data. Juang and Pappa [79] support this claim when the SI methods are applied to simple structures that require low-order models. Juang [72] points out, however, that most SI techniques have difficulties when they are used on more complex structures such as FSS. These structures can only be accurately described by models whose order is in the hundreds [71].

- It is not known what method is appropriate for specific applications. Denman et al. [39] indicate that numerous surveys and reviews have
been compiled on the subject of structural SI. In these surveys, the dominant themes have been:

- to organize the extensive literature within some logical framework
- to present case studies or examples which examine specific methods
- to compile a bibliography

Denman et al. [39] have pointed out that the examples chosen in case studies vary from method to method, offering little opportunity for the comparison of methods. Bibliographies tend to reflect the perceptions held by different writers on SI.

Juang [72] supports this view, indicating that it is difficult to sort out the relative merits of various SI methods from the multitude of approaches within the field. Despite the large amount of SI literature available, Juang [72] has reported that few papers actually compare different SI techniques using experimental data. Juang [72] has stated that, in order to apply SI methods to structures, it is desirable to have comparisons available. Juang and Pappa [79] have concluded that better methods for comparing various identification techniques with complex data are needed. They go on to say that a principal goal is to find a common basis to explain and to select from the variety of possible SI techniques.

- When identifying a model, high accuracy and low order tend to be conflicting requirements. A methodology to achieve both characteristics is needed.
1.3 Objective

Given these unresolved issues, the objective of this research is to:

Establish appropriate SI methods for on-orbit modeling of lightly-damped flexible spacecraft and extend the selected SI methods to improve their performance using Daisy test data. The identified models should be in a form suitable for designing model-based controllers.

1.4 Daisy

The Daisy structure, shown in Figures 1.2 and 1.3, emulates FSS exhibiting the following characteristics:

- **flexible modes**: The assumption of the spacecraft being a single rigid body is not valid for high control performance.

- **rigid modes**: The FSS will exhibit rigid modes.

- **clustered frequencies**: SI techniques will be required to identify many vibration modes in a given frequency range.

- **lightly-damped**: Long test times may be required to characterize the damping.

- **large model order**: Models of large dimension will be required to characterize the dynamics of the spacecraft.
Figure 1.2: The Daisy Facility
Figure 1.3: The Daisy Facility
CHAPTER 1. INTRODUCTION

- sensors and actuators: Several sensors and actuators will exist on the spacecraft. Often the same sensors and actuators used in SI are employed during closed-loop control.

Developed at UTIAS, Daisy is a testbed used to evaluate the effectiveness of SI and control techniques for flexible spacecraft. Daisy consists of a rigid hub, representative of the main body of a satellite, which has three rotational degrees of freedom (roll, pitch, yaw) aligned with the principal axes. Attached to the hub are ten slender rigid ribs forming a cone. Each rib is attached at its mass center to the hub via universal joints and is free to rotate about its pivot in two mutually perpendicular directions: "in cone" and "out of cone" (by analogy to "in plane" and "out of plane"). Ribs are coupled to adjacent ribs by weak springs and have stiffness relative to the hub in both rotational directions. This configuration presents 20 elastic degrees of freedom (DOF) and 3 rigid DOF.

For SI purposes, Daisy is excited by three reaction wheels located in the hub. Rib motion is sensed via an optical deflection sensing system for measuring rib deflections relative to the hub. Attitude encoders measure hub angular displacements. A total of three inputs and twenty-three outputs are used for SI.

The fact that the ribs have discrete flexibility instead of continuous flexibility is important in coping with gravity effects, but has no drawbacks in representing flexible spacecraft generically since the salient dynamic characteristics of flexible spacecraft are successfully represented.
1.5 DICE

The current Dynamics, Identification and Control Experiment (DICE) design resembles Daisy’s structure (see Figure 1.4). As a joint project between Dynacon Enterprises Ltd. and the Spacecraft Dynamics Group at UTIAS, DICE will be a free-flying module in the space shuttle middeck. Since DICE also exhibits the FSS characteristics described in § 1.4, the SI approaches developed in this thesis are equally applicable to DICE.

Figure 1.4: Dynamics, Identification and Control Experiment: DICE
Chapter 2

SI Review and Selection

Establishing appropriate SI methods for on-orbit modeling of lightly-damped flexible spacecraft requires an extensive literature review. The author reviews over 180 papers and divides the SI field into several categories. The SI methods are then evaluated and the results of the selection process are summarized in Table 2.1. Refer to Appendix A for a list of acronyms and Appendix B for the details of the selection process.
Table 2.1: Selection Summary

<table>
<thead>
<tr>
<th>SI Category</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Representations</td>
<td>state-space</td>
</tr>
<tr>
<td></td>
<td>ARX</td>
</tr>
<tr>
<td>Estimators</td>
<td>LS</td>
</tr>
<tr>
<td>Calculating Markov Parameters</td>
<td>OKID</td>
</tr>
<tr>
<td>Calculating Covariance Parameters</td>
<td>stochastic</td>
</tr>
<tr>
<td></td>
<td>deterministic</td>
</tr>
<tr>
<td>Time-Domain Algorithms</td>
<td>ERA</td>
</tr>
<tr>
<td></td>
<td>ERA/DC</td>
</tr>
<tr>
<td></td>
<td>Fast ERA</td>
</tr>
<tr>
<td></td>
<td>Q-Markov CovER</td>
</tr>
<tr>
<td></td>
<td>ORSE</td>
</tr>
<tr>
<td></td>
<td>Subspace</td>
</tr>
</tbody>
</table>
2.1 System Identification Categories

SI on structural systems can be grouped into three categories as shown in Figure 2.1.

![Figure 2.1: System Identification field](image)

2.1.1 Control-Model Identification

Juang [72] explains that the identification of a model which adequately describes the input/output relationships of a structure is required when designing model-based controllers. The actuators used as inputs and the sensors used as outputs are chosen to meet the control performance requirements.

2.1.2 Modal Parameter Identification

Modal parameters of a structure include damping, frequencies and mode shapes. Modal parameter identification, or modal testing, is the process of
exciting a structure, measuring signals produced by the structure and then identifying the modal parameters [72]. The type of actuators (or sensors) and their location on the structure are chosen to excite (or measure) the structural modes of interest. For active control of space structures, experimental modal data can be used directly for controller design [79]. Alternatively, SI techniques are used to identify a state-space model for modal parameter identification and the identified model is then used in controller design [72]. It should be noted that if active control hardware is used for modal testing, the hardware may not be able to excite all the modes of interest and there may not be enough response channels to sufficiently characterize the modes. Additional instrumentation may be required. Denman et al. [39] indicate that in order to be able to excite all the modes of the structure and then identify that structure, the system must be controllable as well as observable. Controllability assures that all the modes will be excited. Observability means that the output data contain information on all of the desired vibrational modes of the structure.

A survey paper by Allemang and Brown [3] provides a brief overview of methods used in the field of the modal testing. Juang and Pappa [79] present an overview of the parallel historical development of modal testing used in structural engineering and SI used for controller design. They indicate that modal parameter identification and control-model identification are closely related. If, for example, the identified model is a linear model in state-space representation, then the model can be used for controller design and for identification of the modal characteristics. The eigensolution of the model determines the modal parameters of the structure.
2.1.3 Structural-Model Parameter Identification

The identification of model parameters, such as mass, stiffness and damping, is based on the assumption that the structure of the model is known.

Ibáñez [60] reviewed the literature pertaining to structural-model parameter identification techniques prior to 1978. He divides model parameter identification into time-domain and frequency-domain methods. Since then, several classification techniques have been developed to further categorize the identification of model parameters. Denman et al. [39] summarize these classification techniques and conclude that model parameter identification methods can be classified according to:

- the model and its corresponding parameters
- the experimental data used in parameter estimation
- the estimation algorithms used to estimate parameter values from experimental data

Stiles and Kosmatka [154] compare several SI methods used to correct structural models using experimental data.

2.2 Control-Model Identification Categories

The above three SI fields are distinguished from each other by their underlying assumptions which become increasingly restrictive and difficult to satisfy. The present author selects control-model identification to satisfy the thesis
objective (see § 1.3) since the fundamental requirement for designing model-based controllers is a model that accurately describes the input/output relationship of a system.

Following the categories recommended by Ibáñez [60] and Denman et al. [39] for structural-model parameter identification, the present author divides the control-model identification field into the four categories shown in Figure 2.2.

**Figure 2.2: Control-Model Identification Categories**

**Time-Domain versus Frequency-Domain SI**

An important consideration is whether to choose a time-domain or frequency-domain algorithm. Time-domain methods use impulse response functions, or Markov parameters, while frequency-domain methods use transfer functions to identify system properties. Theoretically there is no reason why a time-
domain method would be superior to a frequency-domain implementation of the method or vice versa [159]. Differences are observed when time- and frequency-domain algorithms are applied in practice. When data records are relatively short, for example, then it is well accepted that time-domain algorithms are not only preferred, but also necessary.\footnote{Refer to Hollkamp [51] and [52], Yao and Pandit [174], and Fassois et al. [40] for good comparisons of time and frequency-domain SI approaches.} The Daisy facility at UTIAS is capable of recording 2400 data points per output channel during an experiment. Based on MACE experiments, between 10 and 100 times as many data are required to calculate accurate frequency-domain data. Note that as flight hardware improves, data handling capabilities will become less of an issue.

The present author selects time-domain SI algorithms to satisfy the thesis objective (see § 1.3).

2.3 System Identification Selection

It is evident that there are many different types of SI algorithms available and a variety of survey and comparison papers exist in the literature. In order to select appropriate SI algorithms it is necessary to understand the characteristics of the structure to be identified and define criteria to be used in the selection process. Appendix B describes this process in more detail. Note that the goal of this selection procedure is not to select the single best SI method. This selection procedure provides a quantitative ranking of all
methods studied. A small number of methods having the highest score are selected for further analysis and testing using the Daisy structure. Table 2.1 summarizes the results of this selection process.

2.3.1 State-Space Models

The selected SI methods identify discrete-time, linear, time-invariant state-space models described by the following set of difference equations:

\[
\begin{align*}
    x(k + 1) &= \mathbf{Ax}(k) + \mathbf{Bu}(k) \\
    y(k) &= \mathbf{Cx}(k) + \mathbf{Du}(k)
\end{align*}
\]

where state matrix \( \mathbf{A} \in \mathbb{R}^{n \times n} \), input influence matrix \( \mathbf{B} \in \mathbb{R}^{n \times r} \), output influence matrix \( \mathbf{C} \in \mathbb{R}^{m \times n} \), and transmission matrix \( \mathbf{D} \in \mathbb{R}^{m \times r} \). State vector \( x \in \mathbb{R}^n \), output vector \( y \in \mathbb{R}^m \), and input vector \( u \in \mathbb{R}^r \). Note that \( m \) is the number of outputs, \( r \) is the number of inputs, \( n \) is the order of the system, and \( k \) enumerates the sample instants \( (k = 0, 1, 2, \ldots) \). The unknown state-space system is represented in Figure 2.3 where \( \Delta \) represents a delay.

![Figure 2.3: A linear time-invariant system](image)
Overschee and De Moor [164] indicate that many physical systems can be described very accurately by this type of model. There are also many control system design tools available that can be used to build a controller based on this type of model.

The present author performed SI experiments with the Daisy facility by exciting the structure with three reaction wheels located in the hub. For a linear system, excitation amplitude does not theoretically matter. For a nonlinear system such as Daisy, the excitation level does matter. Hong et al. [53] have demonstrated that if the excitation level is too low, then nonlinear hub friction (produced by the rolling friction in the ball bearings that support Daisy's weight) dominates the measurements. If the excitation level is too high, then other nonlinear effects become significant. Such effects include the restriction of motion caused by Daisy's wire harness as well as sensor saturation caused when rib tip motion is outside the field of view. The level of excitation used throughout this thesis lies between these two extremes where the linear system assumption made by the selected SI methods is reasonable.\footnote{Note that on-orbit SI experiments such as DICE (see § 1.5), being free of the support structures required on Earth, are not expected to exhibit the nonlinear hub friction observed with Daisy.}

Even for this level of excitation, nonlinearities such as friction will still be present in Daisy's response. Nonlinear friction is, therefore, one source of model error observed throughout this thesis.\footnote{Section 6.2 demonstrates the relatively small effect nonlinear hub friction has on the accuracy of the identified models for the level of excitation used in this thesis.} Other sources of model er-
ror include errors in measurements such as sensor resolution, time-varying parameters such as reaction-wheel gyricity, time delays\(^4\) associated with the Daisy facility and numerical round-off errors in the SI algorithms.

\(^4\)Refer to Hong [55] for detailed analyses of time delays with Daisy.
Chapter 3

Comparison

The survey and selection process described in Appendix B uses published literature to establish that OKID with ERA, Q-Markov CovER, ORSE and Subspace are appropriate SI methods for on-orbit modeling of lightly-damped flexible spacecraft. In this chapter, the author examines the selected algorithms in detail, clarifies the relationships between the algorithms (shown by the dotted lines in Figure 3.1), and successfully tests each algorithm on Daisy.
3.1 OKID

The Observer Kalman filter Identification (OKID) algorithm is discussed by Juang [72]. OKID begins with the discrete-time state-space linear model given by Equations (2.1) and (2.2).

When OKID is applied to Daisy, the present author uses the three reaction wheels in the hub ($r = 3$) to randomly excite the structure (see Figure 1.2). Reaction wheel rates are used as the inputs to OKID and all 23 degrees of freedom (DOF) are measured ($m = 23$). The relationship between the input $u$ and output $y$, assuming zero initial conditions, can be expressed in the following matrix form:

$$ y = YU $$

(3.1)
where

\[
Y \triangleq \begin{bmatrix} y(0) & y(1) & y(2) & \cdots & y(l-1) \end{bmatrix} \quad (3.2)
\]

\[
Y \triangleq \begin{bmatrix} D & CB & CAB & \cdots & CA^{l-2}B \end{bmatrix} \quad (3.3)
\]

\[
U \triangleq \begin{bmatrix} u(0) & u(1) & u(2) & \cdots & u(l-1) \\
& u(0) & u(1) & \cdots & u(l-2) \\
& & u(0) & \cdots & u(l-3) \\
& & & \ddots & \vdots \\
& & & & u(0) \end{bmatrix} \quad (3.4)
\]

The parameter \( l \) is the number of data samples and the matrix \( Y \) contains the Markov parameters. If the input vector \( u(k) \) is the discrete-time impulse then the response \( y \) can be assembled into a sequence of \( m \times r \) Markov parameters \( Y_k \) as follows:

\[
Y_0 = D, \quad Y_1 = CB, \quad Y_2 = CAB, \quad \ldots, \quad Y_k = CA^{k-1}B \quad (3.5)
\]

A state estimator, or observer, provides an estimate of the system state from the input and output measurements. To derive the observer model, add and subtract the term \( Gy(k) \) to Equation (2.1) as follows:

\[
x(k+1) = Ax(k) + Bu(k) + Gy(k) - Gy(k) \quad (3.6)
\]

where \( G \) is an \( n \times m \) matrix.

The original system described by Equations (2.1) and (2.2) becomes the discrete-time state-space observer model:

\[
x(k+1) = \tilde{A}x(k) + \tilde{B}v(k) \quad (3.7)
\]

\[
y(k) = Cx(k) + \tilde{D}v(k) \quad (3.8)
\]
where
\[
\mathbf{v}(k) \triangleq \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}, \quad \widetilde{\mathbf{A}} \triangleq \mathbf{A} + \mathbf{G} \mathbf{C}, \quad \widetilde{\mathbf{B}} \triangleq [\mathbf{B} + \mathbf{G} \mathbf{D} - \mathbf{G}] \quad \widetilde{\mathbf{D}} \triangleq [\mathbf{D} \quad \mathbf{0}]
\]
(3.9)

Although the structure of Equation (3.7) is similar to Equations (2.1), Equation (3.7) uses different system matrices and has a different input. A parameter sequence, similar to Equation (3.5) is defined for Equations (3.7) and (3.8) as follows:
\[
\mathbf{\bar{Y}}_0 = \widetilde{\mathbf{D}}, \quad \mathbf{\bar{Y}}_1 = \mathbf{C} \mathbf{\bar{B}}, \quad \mathbf{\bar{Y}}_2 = \mathbf{C} \mathbf{\bar{A}} \mathbf{\bar{B}}, \quad \ldots, \quad \mathbf{\bar{Y}}_k = \mathbf{C} \mathbf{\bar{A}}^{k-1} \mathbf{\bar{B}}
\]
(3.10)

The matrices in this sequence are called observer Markov parameters.

The eigenvalues of \( \widetilde{\mathbf{A}} \) are different from \( \mathbf{A} \) due to the \( \mathbf{G} \mathbf{C} \) term (Equation 3.7). Since \( \mathbf{G} \) can be arbitrarily chosen, \( \widetilde{\mathbf{A}} \) can be made as asymptotically stable as desired.

Using the gain matrix \( \mathbf{G} \), OKID places the eigenvalues of \( \widetilde{\mathbf{A}} \) at the origin so that the observer Markov parameters \( \mathbf{C} \mathbf{\bar{A}}^k \mathbf{\bar{B}} \) equal \( \mathbf{0} \) for a finite \( k \geq p \) (where \( p \) is a sufficiently large integer). The observer gain matrix \( \mathbf{G} \) used to satisfy the above condition is called a deadbeat observer.

If a deadbeat observer is implemented then the input/output description in matrix form of Equations (3.7) and (3.8) can be expressed in a format similar to Equation (3.1):
\[
\mathbf{y} = \mathbf{\bar{Y}} \mathbf{v}
\]
(3.11)
where

\[ y \triangleq \begin{bmatrix} y(0) & y(1) & y(2) & \ldots & y(p) & \ldots & y(l-1) \end{bmatrix} \]  \hspace{1cm} (3.12)

\[ \tilde{Y} \triangleq \begin{bmatrix} D & C\tilde{B} & C\tilde{A}\tilde{B} & \ldots & C\tilde{A}^{p-1}\tilde{B} \end{bmatrix} \]  \hspace{1cm} (3.13)

\[ V \triangleq \begin{bmatrix} u(0) & u(1) & u(2) & \ldots & u(p) & \ldots & u(l-1) \\ v(0) & v(1) & v(2) & \ldots & v(p-1) & \ldots & v(l-2) \\ \vdots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ v(0) & \ldots & v(p-2) & \ldots & v(l-3) \\ \vdots & \ddots & \vdots & \ldots & \vdots & \ldots & \vdots \\ v(0) & \ldots & v(l-p-1) \end{bmatrix} \]  \hspace{1cm} (3.14)

where \( l \) is the number of data samples and the matrix \( \tilde{Y} \) contains the \( p + 1 \) observer Markov parameters.

To solve for the observer Markov parameters \( \tilde{Y} \) the following least-squares solution can be used:

\[ \tilde{Y} = yV^T[VV^T]^{-1} \]  \hspace{1cm} (3.15)

The above equation is applied using Daisy experimental data. The present author uses \( l = 2400 \) data samples collected at 10 Hz and selects an observer order\(^1\) of \( p = 8 \). \( \tilde{Y} \) can be partitioned as follows:

\[ \tilde{Y} = [\tilde{Y}_0 \quad \tilde{Y}_1 \quad \tilde{Y}_2 \quad \ldots \quad \tilde{Y}_p] \]  \hspace{1cm} (3.16)

\(^1\)The effect of different observer orders is examined in Chapter 4.
where

\[
\begin{align*}
\widetilde{Y}_0 &= D \\
\widetilde{Y}_k &= C\tilde{A}^{k-1}\tilde{B} \\
&= [C(A + GC)^{k-1}(B + GD) - C(A + GC)^{k-1}G] \\
&= [\tilde{Y}^{(1)}_k - \tilde{Y}^{(2)}_k]
\end{align*}
\]

for \(k = 1, 2, 3, \ldots\)

Figure 3.2 shows a sample of the observer Markov parameters calculated using Daisy experimental data. Since an observer order of \(p = 8\) is selected, the observer Markov parameters go to zero after 8 time steps. Note from Equation (3.20) that Daisy's observer Markov parameters consist of matrices \(\tilde{Y}^{(1)}_k\) and \(\tilde{Y}^{(2)}_k\) having dimensions of \(23 \times 3\) and \(23 \times 23\) respectively. Figure 3.2 is tracking only the first element in each matrix.

Juang [72] demonstrates that the desired system Markov parameters \(Y\) can be recovered from the observer Markov parameters \(\widetilde{Y}\) using the following relationship:

\[
\begin{align*}
D &= \ Y_0 = \widetilde{Y}_0 \\
Y_k &= \tilde{Y}^{(1)}_k - \sum_{i=1}^{k} \tilde{Y}^{(2)}_i Y_{k-i} \quad k = 1, \ldots, p \\
Y_k &= -\sum_{i=1}^{p} \tilde{Y}^{(2)}_i Y_{k-i} \quad k = p + 1, \ldots, \infty
\end{align*}
\]

Note that Equations (3.22) and (3.23) imply that there are only \(p\) independent system Markov parameters. Figure 3.3 shows a sample of the resulting
Figure 3.2: Daisy Observer Markov Parameters $[\tilde{Y}_k^{(1)} \quad \tilde{Y}_k^{(2)}]$ 

$23 \times 3$ system Markov parameters for Daisy. This figure tracks the first 50 seconds of the matrix element corresponding to rib three's in-cone (IC) response to an impulse at the pitch reaction wheel. Note that the response is dimensionless since OKID is applied to normalized Daisy data.\textsuperscript{2} This low-frequency, lightly-damped response is characteristic of all Daisy's system Markov parameters.

To identify the observer gain $G$, observer gain Markov parameters $Y_k^o$ are required:

$$Y_k^o = CA^{k-1}G \quad k = 1, 2, 3, \ldots \quad (3.24)$$

The observer gain Markov parameters $Y_k^o$ can be recovered from the observer

\textsuperscript{2}Integrated Systems [64] recommends normalizing raw SI data to improve matrix conditioning.
Markov parameters $\tilde{Y}$ using the following relationship:

\begin{align}
Y_1^o &= CG = \tilde{Y}_1^{(2)} \\
Y_k^o &= \tilde{Y}_k^{(2)} - \sum_{i=1}^{k-1} \tilde{Y}_i^{(2)}Y_{k-i}^o \quad k = 2, \ldots, p \\
Y_k^o &= -\sum_{i=1}^{p} \tilde{Y}_i^{(2)}Y_{k-i}^o \quad k = p + 1, \ldots, \infty
\end{align}

The observer gain $G$ can then be computed from

$$G = (P^TP)^{-1}P^TY^o$$

(3.28)
where

\[
\begin{align*}
P &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^k \end{bmatrix} \\
Y^o &= \begin{bmatrix} Y_1^o \\ Y_2^o \\ Y_3^o \\ \vdots \\ Y_{k+1}^o \end{bmatrix}
\end{align*}
\] (3.29)

Juang [72] points out that, due to the presence of factors such as disturbances, nonlinearities and the non-whiteness of the process and measurement noise, the identified observer gain matrix G is not the Kalman filter. When these factors are present (the practical case), the identified filter is an observer that minimizes the prediction error in a least-squares sense.

Having determined Daisy’s Markov parameters, a method such as ERA (see § 3.2) can be used to realize the desired discrete-time state-space matrices \([A \ B \ C \ D]\).
CHAPTER 3. COMPARISON

3.2 ERA

The Eigensystem Realization Algorithm (ERA), as discussed by Juang [72], begins with the discrete-time state-space model described by Equations (2.1) and (2.2). The system realization problem is as follows:

Given the sequence of Markov parameters \( Y_k \) for a system:

\[
Y_k, \quad 0 \leq k < \infty
\]  

(3.30)

find the corresponding state-space matrices \( A, B, C \) and \( D \).

Each Markov parameter is an \( m \times r \) matrix where \( m \) is the number of outputs and \( r \) is the number of inputs. Daisy's Markov parameters are, therefore, of size \( 23 \times 3 \). Using the system Markov parameters \( Y_k \) from OKID (§ 3.1), the generalized \( qm \times dr \) Hankel matrix is formed as follows:

\[
H_j \triangleq \begin{bmatrix}
Y_{j+1} & Y_{j+2} & \ldots & Y_{j+d} \\
Y_{j+2} & Y_{j+3} & \ldots & Y_{j+d+1} \\
\vdots & \vdots & \ddots & \vdots \\
Y_{j+q} & Y_{j+q+1} & \ldots & Y_{j+q+d-1}
\end{bmatrix}
\]

(3.31)

Assuming that \( dr > qm \), the maximum rank of \( H_j \) is \( qm \). Since the observer order \( p \) determines the maximum number of independent system Markov parameters (Equations (3.22) and (3.23)), the maximum rank of \( H_j \) becomes \( pm \) where \( q = p \). Having already selected an observer order of \( p = 8 \) in § 3.1, the rank of the Hankel matrix \( H_j \) for Daisy is 184.
ERA first processes $H_0$:

$$H_0 = \begin{bmatrix}
  Y_1 & Y_2 & \ldots & Y_d \\
  Y_2 & Y_3 & \ldots & Y_{d+1} \\
  \vdots & \vdots & \ddots & \vdots \\
  Y_q & Y_{q+1} & \ldots & Y_{q+d-1}
\end{bmatrix} \tag{3.32}$$

Substituting the Markov parameters into Equation (3.32) yields:

$$H_0 = \begin{bmatrix}
  CB & CAB & \ldots & CA^{d-1}B \\
  CAB & CA^2B & \ldots & CA^dB \\
  \vdots & \vdots & \ddots & \vdots \\
  CA^{q-1}B & CA^qB & \ldots & CA^{q+d-2}B
\end{bmatrix} \tag{3.33}$$

Equation (3.33) can be rewritten as:

$$H_0 = \begin{bmatrix}
  C \\
  CA \\
  CA^2 \\
  \vdots \\
  CA^{q-1}
\end{bmatrix} \begin{bmatrix}
  B & AB & A^2B & \ldots & A^{d-1}B
\end{bmatrix} \tag{3.34}$$

where $P_q$ and $Q_d$ are the observability and controllability matrices respectively.

Similarly, $H_1$ can be written as:

$$H_1 = P_qA_Qd \tag{3.36}$$
and in general,

$$H_j = P_q A^j Q_d$$  \hspace{1cm} (3.37)

ERA performs a singular value decomposition (SVD) of the Hankel matrix $H_0$:

$$H_0 = P_q Q_d = R \Sigma S^T$$  \hspace{1cm} (3.38)

where the columns of matrices $R$ and $S$ are orthonormal and $\Sigma$ is a diagonal matrix containing the singular values. Having picked $q = p = 8$, the maximum number of singular values is $mp = 184$ (the rank of the Hankel matrix). Figure 3.4 plots the singular values calculated using Daisy experimental data.

$P_q$ and $Q_d$ are chosen to satisfy Equation (3.38) as follows:

$$P_q = R \Sigma^{1/2} \hspace{1cm} Q_d = \Sigma^{1/2} S^T$$  \hspace{1cm} (3.39)

The state matrix $A$ is obtained from Equation (3.36) using a least-squares solution:

$$A = P_q^\dagger H_1 Q_d^\dagger$$  \hspace{1cm} (3.40)

where the $\dagger$ indicates the pseudo-inverse. Given Equations (3.39) and (3.40) it is evident that the quantity $mp$ represents an upper bound on the order $n$ of the identified state matrix $A$. This implies that the observer order $p$ in OKID should be selected such that $mp \geq n$.

Using Equation (3.39), the pseudo-inverse of $P_q$ and $Q_d$ can be calculated from:

$$P_q^\dagger = \Sigma^{-1/2} R^T \hspace{1cm} Q_d^\dagger = S \Sigma^{-1/2}$$  \hspace{1cm} (3.41)
Substituting Equation (3.41) into Equation (3.40) yields the following form of the state matrix $A$:

$$A = \Sigma^{-1/2} R^T H_1 S \Sigma^{-1/2}$$  \hspace{1cm} (3.42)

If $O_i$ is a null matrix of order $i$, and $I_i$ is an identity matrix of order $i$, then:

$$E_m^T = \begin{bmatrix} I_m & O_m & \cdots & O_m \end{bmatrix} \quad E_r^T = \begin{bmatrix} I_r & O_r & \cdots & O_r \end{bmatrix}$$  \hspace{1cm} (3.43)

where $m$ is the number of outputs and $r$ is the number of inputs.

By examining the structure of the controllability matrix $Q_d$ in Equation (3.34) and utilizing the definition given in Equation (3.39) it becomes evident
that the \( B \) matrix may be calculated from:

\[
B = Q_d E_r = \Sigma^{1/2} S^T E_r
\]  

(3.44)

Similarly, the \( C \) matrix may be obtained from the observability matrix \( P_q \) (see Equations (3.34) and (3.39)) as follows:

\[
C = E_m^T P_q = E_m^T R \Sigma^{1/2}
\]  

(3.45)

The \( D \) matrix is obtained immediately from the system Markov parameters shown in Equation (3.5):

\[
D = Y_0
\]  

(3.46)

Note that the expression for \( A \) in Equation (3.42) can also be derived by partitioning the observability matrix \( P_q \) as:

\[
P_q = \begin{bmatrix}
C \\
CA \\
CA^2 \\
\vdots \\
CA^{q-1}
\end{bmatrix}
\]

\[
P_q^{(1)} = \begin{bmatrix}
C \\
CA \\
CA^2 \\
\vdots \\
CA^{q-2}
\end{bmatrix}
\]  

(3.47)

\[
P_q^{(2)} = \begin{bmatrix}
CA \\
CA^2 \\
\vdots \\
CA^{q-1}
\end{bmatrix}
\]

Observe that

\[
P_q^{(1)} A = P_q^{(2)}
\]  

(3.48)

and

\[
A = P_q^{(1)^T} P_q^{(2)}
\]  

(3.49)
Adding an additional block row to the observability matrix $P_q$ allows the definitions of $H_0$ and $H_1$ in Equations (3.35) and (3.36) to be rewritten as:

$$H_0 = P_q^{(1)}Q_d = RS^T$$  \hspace{1cm} (3.50)
$$H_1 = P_q^{(2)}Q_d$$  \hspace{1cm} (3.51)

from which one can write

$$P_q^{(1)} = R \Sigma^{1/2} \hspace{1cm} Q_d = \Sigma^{1/2}S^T$$  \hspace{1cm} (3.52)

Solving for $P_q^{(1)\dagger}$ and $P_q^{(2)}$ gives:

$$P_q^{(1)\dagger} = \Sigma^{-1/2}R^T \hspace{1cm} P_q^{(2)} = H_1 Q_d^\dagger = H_1 S \Sigma^{-1/2}$$  \hspace{1cm} (3.53)

Substituting Equation (3.53) into Equation (3.49) yields the same result shown in Equation (3.42).

When all singular values are selected from Figure 3.4, ERA identifies a state-space model of Daisy of order $n = 184$ as expected. Although there are a variety of methods available to reduce the size of an identified model (see Chapter 5), a straightforward method is to truncate rows and columns in the identified model corresponding to high frequencies. The model of order $n = 184$ is reduced to a more practical size of $n = 100$ following this approach.

To quantitatively compare the performance of a model, it is necessary to examine the model's ability to predict the input/output relationship of the system. Simulating the identified state-space model using the input data yields the predicted output. This predicted output is then compared with
the desired output data and the prediction error is calculated as follows:

\[
\text{Prediction Error} = \frac{\text{RMS(desired - predicted)}}{\text{RMS(desired)}} \quad (3.54)
\]

\[
= \frac{\|\text{desired - predicted}\|_2}{\|\text{desired}\|_2} \quad (3.55)
\]

Figures 3.5 through 3.10 plot the actual and predicted outputs for Daisy using \( l = 2400 \) data samples collected at 10 Hz. Note that the number appearing in the upper left hand side of each plot represents the corresponding prediction error. The present author also estimates the number of computations required for ERA as shown in Appendix C. A total of 270 mega floating-point operations (MFLOP) — approximately 5 minutes on a SUN Sparc station 20 — and 26 megabytes (MB) are required to identify the state-space Daisy model.

\[3\text{This formula for prediction error was used by Liu and Miller [97].}\]
Figure 3.5: ERA Results for Daisy’s Hub
Figure 3.6: ERA Results for Ribs 1 and 2
Figure 3.7: ERA Results for Ribs 3 and 4
Figure 3.8: ERA Results for Ribs 5 and 6
Figure 3.9: ERA Results for Ribs 7 and 8
Figure 3.10: ERA Results for Ribs 9 and 10
3.3 Q-Markov CovER

The Q-Markov Covariance Equivalent Realization Algorithm (Q-Markov CovER), developed by Liu and Skelton [98], requires both Markov and covariance parameters. Q-Markov CovER has a stochastic as well as a deterministic formulation. For the stochastic case, the input $u$ is a zero-mean white-noise sequence and the Markov and covariance parameters are defined as follows:

$$Y_k \triangleq R_k^{yu}W^{-1} = \left[\lim_{d \to \infty} \frac{1}{d} \sum_{i=0}^{d} y(i+k)u^T(i)\right]W^{-1} \quad (3.56)$$

$$R_k \triangleq R_k^{yy} = \lim_{d \to \infty} \frac{1}{d} \sum_{i=0}^{d} y(i+k)y^T(i) \quad (3.57)$$

where $W$ is the input covariance matrix defined as:

$$W \triangleq R^{uu} = \lim_{d \to \infty} \frac{1}{d} \sum_{i=0}^{d} u(i)u^T(i) \quad (3.58)$$

Note that a white noise signal, being physically unrealizable, must be substituted for colored noise generated by a finite length random signal.

The stochastic interpretation of Q-Markov CovER matches the first $q$ output-input cross-correlation parameters $R_k^{yu}$ and output covariance parameters $R_k^{yy}$ of a system.

Similar to ERA, the deterministic formulation of Q-Markov CovER requires that the input $u$ be impulses. This formulation matches the first $q$ Markov and covariance parameters of a system. The Markov parameters are defined in Equation (3.5) and the covariance parameters $R_k$ can be calculated as
follows:

\[
R_k \triangleq \lim_{d \to \infty} \sum_{i=0}^{d} Y_{i+k} Y_i^T
\]  \hspace{1cm} (3.59)

where \( Y_i \) are the system Markov parameters.

Both the stochastic and deterministic formulations of Q-Markov CovER are tested on the simple two-mass-spring-damper system shown in Figure 3.11.

![Figure 3.11: Two Mass System](image)

The mass, stiffness, and damping parameters are, respectively, \( m_1 = m_2 = 1 \) kg, \( k_1 = k_2 = 1 \) N/m, and \( c_1 = c_2 = 0.01 \) kg/s. The forces \( u_1 \) and \( u_2 \) are the inputs and the displacements of each mass \( y_1 \) and \( y_2 \) are the outputs. The system is simulated under ideal conditions with a random input signal and Equations (3.56) and (3.57) are used to calculate the Markov parameters \( Y_k \) and covariance parameters \( R_k \). Figure 3.12 tracks the first 50 seconds of the resulting parameters using 2048 and 4096 data points sampled at 5 Hz. Q-Markov CovER is applied with these parameters and Figure 3.13 compares the prediction errors (over the first 2048 data points) of the identified models. It is evident that the results change as more data is collected.
Alternatively, the deterministic formulation of Q-Markov CovER can be implemented. When OKID is used to calculate the Markov parameters of the two-mass system, the calculated Markov parameters overlap the desired parameters. Although Markov parameters are available from OKID, calculating covariance parameters using Equation (3.59) requires an infinite number of them. For a highly-damped system, the Markov parameters can become negligible relatively quickly and $R_k$ can be accurately determined for relatively small values of $d$. For lightly-damped systems, such as flexible spacecraft, the Markov parameters decay slowly and extremely large values of $d$ are required to get good approximations of covariance parameters. As $d$ increases, the number of Markov parameters required from OKID increases. For FSS, having many inputs and outputs, the number of computations may become excessive.

The two-mass system of Figure 3.11 is lightly-damped and, as shown in Figure 3.14, $d$ must be greater than 1000 to get reasonable covariance parameters. Figure 3.15 compares the resulting prediction errors of the models identified using Q-Markov CovER. Model prediction error decreases as more Markov parameters are used in Equation (3.59). Setting $d = 20,000$ yields prediction errors on the order of $10^{-4}$.

The 2400 data points available from Daisy experiments are not sufficient to apply the stochastic approach to Q-Markov CovER. Instead, the deterministic approach is tested where OKID supplies the Markov parameters and Equation (3.59) calculates the covariance parameters. The resulting Markov
Figure 3.12: Markov and covariance parameters using stochastic approach
CHAPTER 3. COMPARISON

Prediction Errors for Q-Markov CovER (Stochastic)

Figure 3.13: Prediction errors using Q-Markov CovER (stochastic)

Covariance Parameters

Figure 3.14: Covariance parameters calculated using deterministic approach
CHAPTER 3. COMPARISON

Figure 3.15: Prediction errors using Q-Markov CovER (deterministic)

and covariance parameters are used to create the following data matrix:

\[ D_q \triangleq R_q - \mathcal{H}_q \tilde{W} \mathcal{H}^T_q \]  \hspace{1cm} (3.60)

where

\[ \mathcal{R}_q \triangleq \begin{bmatrix} R_0 & R_1^T & \ldots & R_{q-1}^T \\ R_1 & R_0 & \ldots & R_{q-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ R_{q-1} & R_{q-2} & \ldots & R_0 \end{bmatrix} \]  \hspace{1cm} (3.61)

\[ \mathcal{H}_q \triangleq \begin{bmatrix} Y_0 & 0 & \ldots & 0 \\ Y_1 & Y_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ Y_{q-1} & Y_{q-2} & \ldots & Y_0 \end{bmatrix} \]  \hspace{1cm} (3.62)

\[ \tilde{W} \triangleq \text{diag}[W \ W \ldots \ W] \]  \hspace{1cm} (3.63)

For the deterministic case, \( \tilde{W} \) is a diagonal matrix of the impulse magnitudes.
and can be replaced by the identity matrix so that:

$$D_q = R_q - H_q H_q^T$$  \hspace{1cm} (3.64)

Performing the singular value decomposition of $D_q$ yields the following:

$$D_q = R \Sigma S^T$$  \hspace{1cm} (3.65)

where the matrices $R$ and $S$ are orthonormal and $\Sigma$ is a diagonal matrix containing the singular values. Let the observability matrix $P_q$ be defined as:

$$P_q \triangleq R \Sigma^{1/2}$$  \hspace{1cm} (3.66)

where $P_q$ can be partitioned similar to ERA as:

$$P_q = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{q-1} \end{bmatrix} \hspace{1cm} P_q^{(1)} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{q-2} \end{bmatrix} \hspace{1cm} P_q^{(2)} = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{q-1} \end{bmatrix}$$  \hspace{1cm} (3.67)

It is evident that the previous partitions are related through $A$ as follows:

$$P_q^{(1)} A = P_q^{(2)}$$  \hspace{1cm} (3.68)

A least-squares solution for $A$ is then:

$$A = P_q^{(1)\dagger} P_q^{(2)}$$  \hspace{1cm} (3.69)

where the $\dagger$ indicates the pseudo-inverse.
CHAPTER 3. COMPARISON

The matrix $B$ can be related to the Markov parameters as follows:

$$
P_q^{(1)} B = \begin{bmatrix} C B \\ C A B \\ C A^2 B \\ \vdots \\ C A^{q-2} B \end{bmatrix} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{q-1} \end{bmatrix}
$$

(3.70)

The matrix $B$ is then calculated as:

$$
B = P_q^{(1)\dagger} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{q-1} \end{bmatrix}
$$

(3.71)

The matrix $C$ can be obtained from $P_q^{(1)}$ using Equation (3.43) as follows:

$$
C = E_m^T P_q^{(1)}
$$

(3.72)

The $D$ matrix is obtained immediately from the system Markov parameters shown in Equation (3.5):

$$
D = Y_0
$$

(3.73)

Applying the deterministic formulation of Q-Markov CovER to Daisy experimental data requires an observer order of $p = 8$ and $d = 2000$ Markov parameters to obtain comparable results to ERA. Figure 3.16 demonstrates that, for the parameters selected, the prediction errors of Q-Markov CovER are comparable to those of ERA in § 3.2. The output channel numbers refer
CHAPTER 3. COMPARISON

to the roll, pitch and yaw hub angular displacement, rib 1 "out of cone" (OC) tip displacement, rib 1 "in cone" (IC) tip displacement, rib 2 OC, rib 2 IC, \ldots, and rib 10 IC. The average prediction errors for Q-Markov CovER and ERA are 0.193 and 0.191 respectively. The present author also estimates the number of computations required for Q-Markov CovER as shown in Appendix C. Approximately 430 MFLOP and 84 MB are required to identify the state-space Daisy model.

ERA and Q-Markov CovER were developed independently and, although Hufford and Robinson [58] suggest that little work has been done on comparing the two methods, some comparisons have been performed. In particular, the present author’s observations are supported by Lew et al. [91] who, using a computer simulation of the NASA Mini-Mast structure, concluded that Q-Markov CovER requires the use of significantly more data than ERA to produce comparable results.
CHAPTER 3. COMPARISON

Figure 3.16: Prediction Errors for Q-Markov CovER and ERA
3.4 ORSE

The Observability Range Space Extraction (ORSE) algorithm is described by Liu and Miller [96]. ORSE uses the \(mq \times mq\) data matrix \(D_q\):

\[
D_q \triangleq YY^T - YU^T(UU^T)^+YU^T^T
\]  \hspace{1cm} (3.74)

where \(m\) is the number of outputs and

\[
Y \triangleq \begin{bmatrix}
y(0) & y(1) & \ldots & y(d-1) \\
y(1) & y(2) & \ldots & y(d) \\
\vdots & \vdots & \ddots & \vdots \\
y(q-1) & y(q) & \ldots & y(d+q-2)
\end{bmatrix}
\]

\[
U \triangleq \begin{bmatrix}
u(0) & u(1) & \ldots & u(d-1) \\
u(1) & u(2) & \ldots & u(d) \\
\vdots & \vdots & \ddots & \vdots \\
u(q-1) & u(q) & \ldots & u(d+q-2)
\end{bmatrix}
\]

Note that the data matrix does not require specific signals such as Markov or covariance parameters. Instead, Daisy’s experimental measurements can be directly used in Equation (3.74).

As defined in Equation (3.34), the observability matrix \(P_q\) has the following form:

\[
P_q = \begin{bmatrix}
C \\
CA \\
CA^2 \\
\vdots \\
CA^{q-1}
\end{bmatrix}
\]  \hspace{1cm} (3.77)
Liu and Miller [95] demonstrate that the data matrix $D_q$ has the same range space as the observability matrix range space of the system generating the input and output data:

$$\text{Range}(D_q) = \text{Range}(P_q) \quad (3.78)$$

The basis vectors of the observability range space can be obtained from the independent column basis-vector matrix $R$ of the data matrix $D_q$. The basis-vector matrix $R$ can be calculated using the singular value decomposition of $D_q$:

$$D_q = R\Sigma S^T \quad (3.79)$$

To obtain comparable SI results to ERA and Q-Markov CovER, the size of the data matrix $D_q$ must be set to $q = 25$. Figure 3.17 plots the $mq = 575$ singular values calculated using Daisy experimental data.

Given the singular value decomposition of Equation (3.79), the relationship between $R$ and $P_q$ is:

$$R = P_q T \quad (3.80)$$

where $T$ is a similarity transformation matrix. Selecting different matrices for $T$ results in equivalent identified state-space models that differ only by a similarity transformation.

If $T$ is set to the identity matrix $I$ then the matrix $R$ can replace $P_q$ in the following equations (see Equations (3.69) and (3.71)) used to calculate $A$ and $C$: 
Having calculated $A$ and $C$, the output $y(k)$ becomes a linear function of the elements of the matrices $B$ and $D$. Let $B_{ij}$ and $D_{ij}$ denote the $ij$-th elements of the matrices $B$ and $D$ respectively. Then the system represented by Equations (2.1) and (2.2) can be rewritten as:

$$A = P_q^{(1)}P_q^{(2)} \quad C = E_m^T P_q^{(1)} \quad \text{(3.81)}$$

$$x(k+1) = Ax(k) + \sum_{i=1}^{n} \sum_{j=1}^{r} I_{ni} u_j(k) B_{ij} \quad \text{(3.82)}$$

$$y(k) = Cx(k) + \sum_{i=1}^{m} \sum_{j=1}^{r} I_{mi} u_j(k) D_{ij} \quad \text{(3.83)}$$
where \( I_n \) and \( I_m \) are identity matrices of dimensions \( n \times n \) and \( m \times m \) and \( I_{ni} \) and \( I_{mi} \) are the \( i \)-th column vectors of \( I_n \) and \( I_m \). Let \( y_{B,ij} \) and \( y_{D,ij} \) be defined as follows:

\[
\begin{align*}
x_{ij}(k + 1) &= Ax_{ij}(k) + I_{ni}u_j(k) \\
y_{B,ij}(k) &= Cx_{ij}(k) \\
y_{D,ij}(k) &= I_{mi}u_j(k)
\end{align*}
\]

The linear relationship between \( y(k) \) and the elements of \( B \) and \( D \) is:

\[
y(k) = \sum_{i=1}^{m} \sum_{j=1}^{r} y_{D,ij}(k)D_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{r} y_{B,ij}(k)B_{ij}
\]

\[
= \phi(k)\theta
\]

where

\[
\phi(k) = [y_{D,11}(k), \ldots, y_{D,m1}(k), \ldots, y_{D,1r}(k), \ldots, y_{D,mr}(k)]
\]

\[
\theta = [D_{11}^T, \ldots, D_{m1}^T, \ldots, D_{1r}^T, \ldots, D_{mr}^T, B_{11}^T, \ldots, B_{n1}^T, \ldots, B_{1r}^T, \ldots, B_{nr}^T]^T
\]

The elements of \( B \) and \( D \) can be calculated using the linear least squares solution (see Equation (3.15)).

The present author finds that ORSE requires large amounts of computer memory when applied to Daisy experimental data. To make the method computationally tractable, only the first 100 singular values of Figure 3.17
are selected. When calculating $B$ and $D$, using Equation (3.87), computer memory limitations allow only one-quarter of the experimental data to be used. The resulting model of order $n = 100$ yields prediction errors similar to ERA and Q-Markov CovER as shown in Figure 3.18. Averaging the prediction errors over all 23 Daisy output channels yields 0.208. Referring to Appendix C, approximately $5.3 \times 10^3$ MFLOP and 160 MB of computer memory are required. Note that a partial eigensolution could be used in place of the full order singular value decomposition to reduce the computational requirements.
CHAPTER 3. COMPARISON

Relationship between ORSE and ERA

Liu and Miller [96] demonstrate that the ORSE formulation for A and C is equivalent to ERA when the impulse response is used. Setting $U$ in Equation (3.74) to be an impulse and $Y$ to be the impulse response gives:

\[
D_q \triangleq YY^T - YU^T(UU^T)^\dagger(YU^T)^T = H_0H_0^T
\]

(3.91)

Appendix D.1 demonstrates that using the data matrix $H_0H_0^T$ to calculate A and C is equivalent to ERA to within a diagonal coordinate transformation.

Relationship between ORSE and Q-Markov CovER

Liu and Miller [96] demonstrate that the ORSE formulation for A and C is equivalent to Q-Markov CovER in the case of an infinite white random input signal. To demonstrate this, multiply the data matrix in Equation (3.74) by the scalar $\frac{1}{d}$ and take the limit of $d$ to infinity:

\[
\lim_{d \to \infty} \frac{1}{d} \left[ YY^T - YU^T(UU^T)^\dagger(YU^T)^T \right] = \mathcal{R}_q - \mathcal{H}_q \tilde{W} \mathcal{H}_q^T
\]

(3.92)

where

\[
\lim_{d \to \infty} \frac{1}{d} YY^T = \mathcal{R}_q \quad \lim_{d \to \infty} \frac{1}{d} YU^T = \mathcal{H}_q \tilde{W} \quad \lim_{d \to \infty} \frac{1}{d}(UU^T)^\dagger = \tilde{W}^{-1}
\]

(3.93)

However, for finite $d$ and a non-white input signal (the practical case) the above equalities do not hold.
Note that Liu and Miller [96] examined the theoretical relationships between ORSE, ERA and Q-Markov CovER. This paper did not compare these SI methods using experimental data. Only the ORSE algorithm was demonstrated using a computer simulation of the MACE test article. Liu and Miller [96] did mention that it took 36 hours of computation on a SUN Sparc station which supports the present author's conclusion that ORSE requires large computational resources.
3.5 Subspace

Subspace algorithms, as discussed by De Moor and Overschee [37], use geometric tools such as orthogonal projections and oblique projections to identify state-space models of a system.

Let the operator $\Pi_B$ project the row space of a matrix onto the row space of the matrix $B$:

$$\Pi_B \triangleq B^T(BB^T)^+B$$  \hfill (3.94)

$A/B$ is shorthand for the projection of the row space of the matrix $A$ onto the row space of the matrix $B$:

$$A/B \triangleq A\Pi_B$$  \hfill (3.95)

$$= AB^T(BB^T)^+B$$  \hfill (3.96)

Let $\Pi_{B\perp}$ be the geometric operator that projects the row space of a matrix onto the orthogonal complement of the row space of the matrix $B$:

$$A/B\perp \triangleq A\Pi_{B\perp}$$  \hfill (3.97)

where

$$\Pi_{B\perp} = I - \Pi_B$$  \hfill (3.98)

Figure 3.19 demonstrates the interpretation of the orthogonal projection in two-dimensional space.
Figure 3.19: Orthogonal Projection in two-dimensional space

The oblique projection of the row space of a matrix $A$ along the row space of $B$ onto the row space of $C$ is defined as:

$$\frac{A}{B}C \triangleq [A/B^\perp][C/B^\perp]^\dagger C$$

(3.99)

Refer to Figure 3.20 for the interpretation of the oblique projection in two-dimensional space.

Figure 3.20: Oblique Projection in two-dimensional space
The input block Hankel matrix $U_{0|2q-1}$ is defined as:

$$
U_{0|2q-1} \triangleq \begin{bmatrix}
  u(0) & u(1) & \ldots & u(d-1) \\
  u(1) & u(2) & \ldots & u(d) \\
  \vdots & \vdots & \ddots & \vdots \\
  u(q-1) & u(q) & \ldots & u(q+d-2) \\
  u(q) & u(q+1) & \ldots & u(q+d-1) \\
  u(q+1) & u(q+2) & \ldots & u(q+d) \\
  \vdots & \vdots & \ddots & \vdots \\
  u(2q-1) & u(2q) & \ldots & u(2q+d-2)
\end{bmatrix}
$$

(3.100)

where the subscript $f$ stands for "future" and the subscript $p$ stands for "past". The output block Hankel matrix $Y_{0|2q-1}$ is defined in a similar way. The block Hankel matrix $V_{0|q-1}$ consisting of inputs and outputs $V$ (following the notation used in Equation (3.7)) is defined as:

$$
V_{0|q-1} \triangleq \begin{bmatrix}
  U_p \\
  Y_p
\end{bmatrix} = V_p
$$

(3.102)

State sequences are important in the derivation of the Subspace identification algorithm. The state sequence $X_k$ is defined as:

$$
X_k \triangleq [x(k) \ x(k+1) \ \ldots \ x(k+d-1)]
$$

(3.103)

where $k$ denotes the first element of the state sequence. The past $X_p$ and future $X_f$ state sequences are given by:

$$
X_p = X_0 \quad X_f = X_q
$$

(3.104)
De Moor and Overschee [37] make the following assumptions:

- The input \( u(k) \) is persistently exciting (the rank of the input covariance matrix \( R^{uu} \) has full rank).

- The intersection of the row space of the future inputs \( U_f \) and the row space of the past states \( X_p \) is empty.

- The user-defined weighting matrices \( W_1 \in \mathbb{R}^{m \times qm} \) and \( W_2 \in \mathbb{R}^{d \times d} \) are such that \( W_1 \) has full rank and \( W_2 \) satisfies: \( \text{rank}(V_p) = \text{rank}(V_p W_2) \), where \( V_p \) is the block Hankel matrix containing the past inputs and outputs.

Under these assumptions, the oblique projection:

\[
\mathcal{O}_q \triangleq Y_f/U_f V_p
\]  

is equal to the product of the observability matrix \( P_q \) (see Equation (3.34)) and the future states \( X_f \):

\[
\mathcal{O}_q = P_q X_f
\]  

This oblique projection can be calculated using Equation (3.99) as follows:

\[
\mathcal{O}_q = [Y_f/U_f]^+[V_p/U_f]^+ V_p
\]  

De Moor and Overschee [37] suggest using an RQ decomposition to perform this calculation. Given Equation (3.106), the observability matrix \( P_q \) can be calculated by taking the singular value decomposition of \( W_1 \mathcal{O}_q W_2 \):

\[
W_1 \mathcal{O}_q W_2 = (W_1 P_q)(X_f W_2)
\]

\[
= R \Sigma S^T
\]

\[
= (R \Sigma^{1/2} T)(T^{-1} \Sigma^{1/2} S^T)
\]
where the columns of matrices $R$ and $S$ are orthonormal, $\Sigma$ is a diagonal matrix containing the singular values, and $T$ is a similarity transformation matrix. Comparing Equations (3.108) and (3.109) gives the following expression for the observability matrix:

$$ P_q = W_1^{-1} R \Sigma^{1/2} T $$  \hfill (3.111)

Following a similar discussion used for ORSE, the transformation matrix $T$ can be set to the identity matrix $I$.

Having calculated the observability matrix $P_q$, the $A$ and $C$ matrices are obtained similar to Q-Markov CovER:

$$ A = P_q^{(1)} P_q^{(2)} \quad C = E_m^T P_q^{(1)} $$ \hfill (3.112)

In a similar manner as ORSE, matrices $B$ and $D$ are obtained by observing the linear relationship between the output $y(k)$ and the elements of $B$ and $D$. The choice of weighting matrices $W_1$ and $W_2$ in Equation (3.108) is made in the following section.

**Relationship between Subspace and ORSE**

Overschee and De Moor [164] classify ORSE as a projection algorithm. They indicate that this class of algorithms can be incorporated into Subspace algorithms by taking:

$$ W_1 = I \quad W_2 = \Pi_{U_1^\perp} $$ \hfill (3.113)
CHAPTER 3. COMPARISON

The present author applies these weights and derives the data matrix $D_q$ used by ORSE by proving that the data matrix used by ORSE has the same range space as $W_1C_qW_2$ used by Subspace methods. With these weights, the difference between $A$ and $C$ identified with Subspace and ORSE is just a similarity transformation. The proof is as follows:

Observe that

$$Y_f = P_qX_f + \xi_qU_f$$

where $\xi_q$ was defined in Equation (3.62). Multiplying Equation (3.114) from the right by $\Pi_{U_f}$ gives:

$$Y_f\Pi_{U_f} = P_qX_f\Pi_{U_f}$$

Substituting Equations (3.113) and (3.115) into Equation (3.108) gives:

$$W_1C_qW_2 = IP_qX_f\Pi_{U_f}$$

$$= Y_f\Pi_{U_f}$$

Using Equation (3.98) one can replace $\Pi_{U_f}$ by $(I - \Pi_{U_f})$ and rewrite Equation (3.117) as (see the definition in Equation (3.94)):

$$W_1C_qW_2 = Y_f(I - \Pi_{U_f})$$

$$= Y_f(I - U_fU_f^T(U_fU_f^T)^T)$$

Since multiplying by the transpose of a matrix will not change its range space, the above equation can be rewritten as:

$$\text{Range}(W_1C_qW_2) = \text{Range}(Y_f(I - U_fU_f^T(U_fU_f^T)^T)(I - U_fU_f^T(U_fU_f^T)^TY_f^T))$$
which can be simplified as follows (see [97]):

\[
\text{Range}(W_1 Q_q W_2) = \text{Range}(Y_f (I - U_f^T (U_f U_f^T) U_f) Y_f^T) \tag{3.120}
\]

\[
= \text{Range}(Y_f Y_f^T - Y_f U_f^T (U_f U_f^T) Y_f^T)
\]

\[
= \text{Range}(D_q) \tag{3.121}
\]

where \(D_q\) is the same data matrix used by ORSE (Equation (3.74)).

The present author tests the above theory using Daisy experimental data. To obtain comparable SI results to ERA, Q-Markov CovER and ORSE, the size of the data matrix \(D_q\) is set to \(q = 25\), weights \(W_1\) and \(W_2\) are chosen as shown in Equation (3.113), and the first 100 singular values are selected. The resulting model is then compared with ORSE where the input and outputs matrices \(U\) and \(Y\) in ORSE are replaced with \(U_f\) and \(Y_f\) from the Subspace algorithm (see Equation (3.120)). The results of this analysis are shown in Figure 3.21 where, as expected, the prediction errors of Subspace and ORSE are the same. The average prediction error over all 23 Daisy output channels is 0.207. Appendix C demonstrates that Subspace requires approximately \(8.5 \times 10^3\) MFLOP and 230 MB of computer memory. The use of a partial eigensolution in place of the full order singular value decomposition could reduce the computational requirements. Computations may also be reduced by making use of the Hankel structure. See [164].

It is evident that, while ERA, Q-Markov CovER, ORSE and Subspace SI algorithms are appropriate methods of identifying flexible spacecraft such as Daisy, OKID with ERA is capable of achieving comparable results with rela-
Prediction Errors for Subspace and ORSE

Figure 3.21: Prediction Errors for Subspace and ORSE

...tively few computations and computer memory requirements. The following section, therefore, focuses on ERA and its variations selected in § B.2.
Chapter 4

ERA and Variations

Figure 4.1 shows the various relationships between ERA and the variations of ERA studied in this chapter.

4.1 Consecutive and Concurrent Approach

Having established that OKID with ERA is capable of identifying flexible spacecraft using relatively low computational resources, the present author examines the variations of ERA selected in § B.2. Appendix D.1 describes the Eigensystem Realization Algorithm with Data Correlations (ERA/DC) [72]. The formulation of ERA/DC involves creating a block correlation Hankel matrix $\mathcal{H}_j$ as shown in Equation (D.3). In general, this matrix has $\xi$ block rows and $\zeta$ block columns. Each block consists of a data correlation matrix $R_{j}^{hh}$ separated by $\tau$ to prevent significant overlap of adjacent blocks. As more block correlation matrices are included, the matrix dimensions increase and the computational requirements also increase.
Lew et al. [91], Juang et al. [75] and Peterson [127] demonstrate that ERA is equivalent to the special case of ERA/DC when $\tau = 0$. Appendix D.1 describes this special case as well as its relationship to ERA. What these papers do not examine, however, are the following two implementations of ERA and ERA/DC suggested by Juang [72].

As discussed in § 3.2, when OKID is used with ERA, the Markov parameters $Y_k$ are used to realize the desired discrete-time state-space matrices $[A \ B \ C \ D]$. This procedure is summarized in Figure 4.2. OKID (see § 3.1) also allows the observer gain $G$ to be computed using the least-
squares relationship described in Equation (3.28).

![Diagram of OKID with ERA]

Figure 4.2: OKID with ERA

An alternative method to realize the system and observer gain matrices is to combine Equations (3.21) to (3.23) and Equations (3.25) to (3.27) and solve for $Y_k$ and $Y_k^o$ concurrently.

\[
\mathcal{P}_k \triangleq \begin{bmatrix} Y_k & Y_k^o \end{bmatrix} \tag{4.1}
\]

\[
= [\tilde{Y}_k^{(1)} - \tilde{Y}_k^{(2)}]D \quad \tilde{Y}_k^{(2)} - \sum_{i=1}^{k-1} \tilde{Y}_i^{(2)}[Y_{k-i} \quad Y_{k-i}^o] \tag{4.2}
\]

The combined system and observer gain Markov parameters $\mathcal{P}_k$ are used in the Hankel matrix to identify the matrices $[A \quad [B \quad G] \quad C \quad D]$ using ERA or ERA/DC. The present author coined the words consecutive (c) to represent the former approach and concurrent (C) to represent the latter SI approach of identifying the system and observer gain matrices.

The other variation of ERA selected in § B.2 is called Fast ERA. Summarized in Appendix D.2, Fast ERA employs an eigenvalue decomposition rather than the singular value decomposition of ERA. Peterson [126] demonstrates that Fast ERA is equivalent to the special case of ERA/DC where $\tau = 0$ (and,
therefore, Fast ERA is equivalent to ERA to within a diagonal coordinate transformation). While Fast ERA has sometimes been referred to as a Hankel approximation to Q-Markov CovER (HQMC) [127], Appendix D.2 clarifies the differences between Fast ERA and Q-Markov CovER.

An understanding of the theory is important when comparing SI algorithms; however, the true test of an algorithm occurs when it is applied to simulated and experimental data.

### 4.1.1 Daisy Simulations

A linear computer simulation of Daisy is used to compare ERA c, ERA C, the special cases of ERA/DC c and ERA/DC C, and Fast ERA. The Daisy simulator is excited by three reaction wheels and the simulated 3 hub and 20 rib motions are used as outputs for SI. A normally distributed signal is added to each output $y$ to simulate sensor noise as follows:

$$ y = y + \max(|y|) \times (W) \times (\text{rand}) $$  \hspace{1cm} (4.3)

where $\text{rand}$ is a normal distribution of random numbers with mean 0.0 and variance 1.0, and $W$ is a weighting factor related to the signal-to-noise ratio (SNR) (i.e. for an SNR of 20, $W = 0.05$). Figure 4.3 shows the results of this noise analysis using an observer order $p = 8$ (see § 3.1). Figure 4.3 demonstrates that the SI methods are sensitive to noise and, as expected, all methods yield the same prediction errors for a given level of noise. Figure 4.4 shows a sample of the response for Fast ERA with a signal-to-noise ratio of 50 where the number appearing in the upper left hand side of each plot
represents the corresponding prediction error. Noise significantly affects the ability of the SI algorithms to identify the low frequency hub motion. Note that when applied to Daisy experimental data, all algorithms also yield the same prediction error.

Since all five SI algorithms tested in this chapter yield the same model prediction error, the present author employs the computer requirements criteria of § B.1.1 to finalize the selection. Appendix C summarizes the detailed computations involved in each of the algorithms. Table 4.1 summarizes the computational requirements of the different algorithms for an observer order
Figure 4.4: Sample Response: Fast ERA, signal-to-noise ratio = 50
\( p = 8 \). Note that the same parameters are used in each method and that all methods are applied with the same size Hankel matrix.

Table 4.1: Computation Comparison

<table>
<thead>
<tr>
<th></th>
<th>ERA c</th>
<th>ERA C</th>
<th>ERA/DC c</th>
<th>ERA/DC C</th>
<th>Fast ERA</th>
</tr>
</thead>
<tbody>
<tr>
<td>OKID (MFLOP)</td>
<td>150</td>
<td>130</td>
<td>150</td>
<td>130</td>
<td>150</td>
</tr>
<tr>
<td>ERA variation</td>
<td>120</td>
<td>110</td>
<td>70</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>Total (MFLOP)</td>
<td>270</td>
<td>240</td>
<td>220</td>
<td>190</td>
<td>220</td>
</tr>
</tbody>
</table>

Table 4.1 shows that ERA c requires 1.2 times as many computations as ERA/DC c while ERA C requires 1.3 times as many computations as ERA/DC C. Lew et al. [91] also noted that ERA/DC provides a computational advantage over ERA since ERA/DC performs an SVD on a square matrix while ERA performs an SVD on a rectangular matrix. Table 4.1 goes on to show that ERA c requires 1.2 times as many computations as Fast ERA. Peterson and Bullock [127] supported this observation when they tested Fast ERA and ERA on a computer-simulated simply-supported beam. They concluded that Fast ERA requires less time than ERA (represented in Table 4.1 as ERA c).

Figure 4.5 plots the amount of computations over a variety of observer order \( p \). The computational requirements of Fast ERA and ERA/DC c are expected to be similar since, as demonstrated in Appendix D.2, the only difference between the algorithms is that ERA/DC c uses a singular value decomposition while Fast ERA employs an eigenvalue decomposition.
Consecutive approaches should be preferred to concurrent approaches since it is often necessary to realize only discrete-time state-space system matrices for controller design. Methods such as Fast ERA, ERA/DC c and ERA c are, therefore, favored since they realize the state-space matrices and observer gain matrix as distinct steps. If the additional step of calculating the observer gain is avoided, then, in the case of Fast ERA, computations are reduced by an additional 14%. Peterson [126] explains how computations may be further reduced by using a partial eigensolution, a transpose formulation, and a recursive formation of the data matrix.
CHAPTER 4. ERA AND VARIATIONS

The present author has found that the singular value decomposition (SVD) routines used by MATLAB and Xmath do not always converge successfully. Although the present author has reduced this problem by increasing the number of iterations in the SVD algorithm, the eigenvalue decomposition routine used in Fast ERA does not have convergence problems.

In summary, Fast ERA performs well in the selection process of § B.2, the noise analysis in Figure 4.3 and the computation comparison in Table 4.1. The algorithm allows the observer gain to be calculated as an optional step and it employs a numerically reliable eigenvalue decomposition routine. Fast ERA is, therefore, preferred over the other SI algorithms studied in this thesis.

4.1.2 Daisy Experiments

Having selected Fast ERA, the present author tests its performance on Daisy experimental data using different experiment lengths and observer orders $p$.

Reducing Experiment Time

SI experiments on Daisy use input/output time histories lasting 4 minutes (a hardware limitation). The sample time is 0.1 seconds; therefore, there are 2400 data points per output channel. To determine if less time is sufficient for Fast ERA to characterize the system, a state-space model (referred to as Model 1) is realized using only the first 1200 data points (120 seconds). Model 2 is realized using all 2400 data points (240 seconds). Figure 4.6 sum-
CHAPTER 4. ERA AND VARIATIONS

marizes the results of this analysis. Plot A demonstrates that Model 1 is not representative of the Daisy structure. When the prediction error over the first 120 seconds is examined, Model 1 performs better than Model 2. If, however, the error over all 240 seconds is examined, then Model 1 exhibits a relatively large prediction error while Model 2 has only a slight increase in prediction error.

Plot B of Figure 4.6 shows a sample of the predicted and actual response for Rib 1 OC and demonstrates that Model 1 is capable of representing only the first 120 seconds. Plot C shows the ability of Model 2 to represent all 240 seconds. The following section examines the effect of increasing experiment time.

Increasing Experiment Time

As mentioned above, only 4 minutes of data can be captured with the Daisy facility. Increasing the experiment time beyond 4 minutes requires an understanding of the ergodic hypothesis. This hypothesis states that any statistical quantity calculated by averaging over all repeated experiments of a stationary process at a fixed time step can be calculated by averaging over all time on a single time series [72]. Note that the single time series must display the full range of amplitudes and frequencies which would be found among all repeated experiments of the stationary random process. According to Juang [72], almost all identification results for engineering systems are derived by assuming a stationary process satisfying the ergodic property. In the case of
Figure 4.6: Decreasing SI Experiment Time
Daisy, the present author performs a series of three SI experiments, referred to as Experiment A, B and C. Each lasts 4 minutes at a sampling frequency of 10 Hz and each experiment excites Daisy with a different random input signal. All experiments start from rest and are performed successively to minimize any variations from one experiment to another.

Assuming that Daisy satisfies the ergodic property, results from multiple SI experiments can be incorporated into the OKID algorithm. The input/output data from each experiment $i$ can be assembled into matrices $y_i$ and $V_i$ following the structure defined by Equations (3.12) and (3.14). Equation (3.15) is composed of $yV^T$ and the inverse of $VV^T$. These matrices can be formed as follows:

$$yV^T = \sum_{i=1}^{n_{\text{exp}}} y_i V_i^T \quad VV^T = \sum_{i=1}^{n_{\text{exp}}} V_i V_i^T$$

where $n_{\text{exp}}$ is the number of experiments. It is evident that the above operations increase the apparent number of data points $l$ in Equation (3.12) to incorporate results from multiple experiments.

Having calculated $yV^T$ and $VV^T$, the observer Markov parameters $\tilde{Y}$ can be calculated according to Equation (3.15). The remaining steps in OKID stay the same. Figure 4.7 summarizes the results of this analysis. Continuing the notation of the previous section, Model 3 is realized using the input/output data obtained from Experiment A. Similarly, Model 4 is realized by combining the input/output data obtained from Experiments A and B. Model 5 is realized by combining the input/output data obtained from Experiments A, B and C.
In Plot A of Figure 4.7, Model 3 is simulated using data from Experiments A, B and C. The prediction error is larger when Model 3 is simulated with new input/output data (Experiments B and C). Plot B shows how combining Experiments A and B allows Model 4 to better predict Experiment B. Furthermore, Model 4 can predict Experiment C better than Model 3. Plot C shows the results of combining all three experiments and, as expected, Model 5 can accurately predict Experiment C data. These observations support the conclusion that the longer the experiment, the more representative the SI models are of the actual system.
Plot A: SI with Different Input/Output Data

Plot B: Combining Two Experiments (A,B)

Plot C: Combining Three Experiments (A,B,C)

Figure 4.7: Combining Multiple SI Experiments
CHAPTER 4. ERA AND VARIATIONS

Observer Order

Fixing the experiment length to 4 minutes, Figure 4.5 of § 4.1.1 demonstrates that as the observer order $p$ increases, the number of computations also increases. This is to be expected since, as discussed in § 3.2, the observer order is related to the identified model order $n$ by $n = mp$, where $m$ is the number of outputs. As the observer order $p$ increases, the identified model order $n$ increases. The present author tests Fast ERA on Daisy experimental data using different observer orders. Figure 4.8 demonstrates that as the observer order increases, the accuracy of the identified model also increases. Note that for each observer order, the identified model is reduced to $n = 46$ using a balanced reduction algorithm (see § 5.1.3) so that all prediction errors are calculated using a consistent model order.

![Effect of Observer Order](image)

Figure 4.8: Effect of observer order $p$ on model prediction error
By comparing Figure 4.5 and Figure 4.8 it becomes evident that as the observer order becomes larger, significant increases in computation result in progressively smaller increases in model accuracy. Figures 4.9 through 4.20 compare the actual and predicted response of Daisy for $p = 4$ and $p = 15$. Note that the observer order cannot increase indefinitely due to numerical ill-conditioning problems encountered with large models ($n > 200$). Numerical ill-conditioning often prevents the singular value decomposition procedure of § 3.2 from converging. As discussed in § 5.1.4, small models are much more practical for controller design and existing model order reduction algorithms have difficulty handling models in excess of $n = 200$. 
Figure 4.9: Daisy hub results for $p = 4$ and $p = 15$
Figure 4.10: Daisy hub results for $p = 4$ and $p = 15$
Figure 4.11: Daisy rib 1 results for $p = 4$ and $p = 15$
Figure 4.12: Daisy rib 2 results for $p = 4$ and $p = 15$
Figure 4.13: Daisy rib 3 results for $p = 4$ and $p = 15$
Figure 4.14: Daisy rib 4 results for $p = 4$ and $p = 15$
Figure 4.15: Daisy rib 5 results for $p = 4$ and $p = 15$
Figure 4.16: Daisy rib 6 results for $p = 4$ and $p = 15$
Figure 4.17: Daisy rib 7 results for $p = 4$ and $p = 15$
Figure 4.18: Daisy rib 8 results for $p = 4$ and $p = 15$
Figure 4.19: Daisy rib 9 results for $p = 4$ and $p = 15$
Figure 4.20: Daisy rib 10 results for $p = 4$ and $p = 15$
Chapter 5

High-Order Identified Models

Having established appropriate SI methods for on-orbit modeling of lightly-damped flexible spacecraft, the present author continues to satisfy the research objective (see § 1.3) by addressing the model size issue common to all selected SI methods to improve their performance. As demonstrated in Chapters 3 and 4, in order to get accurate results using methods such as OKID with Fast ERA, the size of the identified state-space model must be significantly larger than the expected system size, a process called *overspecification* [96]. These overspecified models contain the principal modes related to the system and extraneous modes related to measurement errors, non-linearities and numerical round-off errors. The extraneous modes can be eliminated after the identification process through model order reduction since they usually contribute negligibly to the response of the identified system. The resulting low-order, high-accuracy models can be used to design model-based controllers.
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

5.1 Model Order Reduction

This section examines several approaches to model order reduction including model reduction based on singular values of the Hankel matrix $H_j$, frequency-based model reduction, and balanced model reduction techniques.

5.1.1 Singular Value Decomposition

As discussed in § 3.2, the number of Hankel matrix singular values chosen determines the order of the realized model. Figures 3.4 and 3.17 show plots of the singular values calculated using ERA and ORSE. The singular values can be used as a form of model order reduction by retaining the large singular values and truncating the small values [127]. The present author finds (see § 5.1.4) that model prediction errors are lowest when as many singular values as possible are kept. There is often a limit to the number that can be retained since extremely small singular values can create numerical ill-conditioning problems in the resulting matrices. Experience has shown that if the contribution of a singular value is less than $10^{-10}$ of the sum of all singular values, then that singular value should not be included in the realization process.

Using Daisy experimental data, the Hankel matrix singular values calculated using an observer order of $p = 12$ are shown in Figure 5.1. Fast ERA is used to obtain these results where the singular values of $H_0$ are equal to the square root of the eigenvalues of $H_0 H_0^T$ (see § D.1). While the upper limit on the realized model order $n$ is 276, only the largest 230 singular values are
included in the realization, thus avoiding matrix ill-conditioning problems.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{FastERA_SingularValues.pdf}
\caption{Fast ERA Singular Values}
\end{figure}

\subsection{5.1.2 Frequency-based Model Reduction}

As discussed in § 1.4, Daisy has $N = 23$ DOF and can be modeled by a state matrix $A$ of order $n = 2N = 46$. The overparameterized model of $n = 230$ identified in the previous section using an observer order of $p = 12$ contains modes with a wide range of vibration frequencies. Figure 5.2 plots the frequencies of each mode where the highest frequency corresponds to the Nyquist frequency of 5 Hz. Sorting the realized frequencies from smallest to largest and truncating all modes with frequencies above 0.9 Hz yields a reduced order model of $n = 46$ as desired. Truncating the high frequencies is
justified by examining the spectral density of Daisy's measurements in Figure 5.3. As expected, the majority of frequencies in the measurements fall between 0 and 0.2 Hz. Frequencies above 0.2 Hz in the realized model do not contribute significantly to the measured output.

![Fast ERA Identified Frequencies](image)

Figure 5.2: Fast ERA Identified Frequencies

### 5.1.3 Balanced Model Reduction

Carroll [18] provides a survey of balanced model order reduction techniques. The Xmath *balmoore* function computes the balanced form of a system and then truncates it to the desired order using Moore's algorithm [117]. A bal-
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

Figure 5.3: Spectral Density of Daisy's Measurements

advanced system, as explained by Grace et al. [43], has equal diagonal controllability and observability gramian matrices $G_c$ and $G_o$ defined in continuous time as:

$$G_c \triangleq \int_0^\infty e^{At}BB^Te^{At}dt \quad G_o \triangleq \int_0^\infty e^{At}C^TCe^{At}dt \quad (5.1)$$

These unique symmetric positive semidefinite matrices satisfy the matrix Lyapunov equations

$$AG_c + G_cA^T + BB^T = O \quad A^TG_o + G_oA + C^TC = O \quad (5.2)$$

Since these gramian matrices reflect the combined controllability and observability of the system, it is reasonable to remove those states from models
having small gramians. Elimination of these states usually has a minimal effect on the important input/output characteristics of the original system.

5.1.4 Model Order Reduction Comparison

Figure 5.4 compares singular value truncation, frequency-based and balanced model reduction methods.

![Model Order Reduction Techniques](image)

**Figure 5.4: Model Order Reduction Comparison**

This figure demonstrates that all model reduction techniques increase the prediction error. Note that model order reduction to $n = 46$ using balanced model reduction is significantly better than selecting the first largest 46 sin-
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

singular values. Selecting the largest 46 singular values yields models with very poor prediction errors.

Disadvantages of frequency-based model reduction:

- Prediction errors are higher than with the balanced model order reduction technique.
- User interaction is required to determine what cutoff frequency is appropriate.

Disadvantages of balanced model order reduction:

- Balanced model order reduction cannot reduce unstable models (see § 5.2). The gramian matrices given in Equation (5.1) are required for balanced model order reduction. To ensure that $G_c$ and $G_o$ are defined, the impulse response must decay as time $t$ approaches infinity.
- Balanced reduction techniques require more computational resources than frequency-based model order reduction techniques for large order systems. For a system of order $n = 230$, balancing requires 30 minutes on a SUN Sparc station 20 while truncating frequencies requires only 5 minutes.
- Numerical ill-conditioning problems may occur when performing model order reduction with models larger than $n = 200$. 
Advantages of frequency-based model reduction:

- The frequency-based model reduction method works reasonably well with Daisy.
- The model reduction method is fast (5 minutes).
- The model reduction method works with unstable models.

Advantages of the balanced model order reduction:

- Lower prediction errors are achieved with balanced model reduction than with frequency-based model reduction.
- The model reduction method does not require significant user interaction. Only the reduced model size is required.

Note that if the model order is extremely large ($n > 200$), then the author suggests a combined model order reduction approach. First employ singular value truncation to avoid numerical ill-conditioning problems. Then use frequency-based model order reduction to reduce the model to $n < 200$. Finally, apply balanced model order reduction algorithms to reduce the model to the desired order.

5.2 Unstable Models

Model order reduction techniques such as modal cost analysis (Skelton and Hughes, [148]) perform well on matrix-second-order systems while balanced
 CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

model reduction algorithms (Moore, [117]) tend to be well suited for the over-specified state-space models realized with OKID and Fast ERA (see § 5.1.4).

With some exceptions (Chui and Maciejowski, [24]; Peterson and Bullock, [127]), most SI algorithms do not guarantee that the identified state-space model will be stable (even though the physical system is stable). Peterson and Bullock [127] point out that ERA and its variations, for example, do not guarantee realization of stable models. Note that the discrete-time model in Equation (2.1) is stable if and only if all eigenvalues lie inside the unit circle [162].

Although an unstable realized model can still be used to design model-based controllers, balanced model order reduction algorithms require that the model to be reduced be stable. The following sections examine the model stability problem, determine the sources of model instability and suggest three techniques which allow unstable models to be reduced. These techniques are tested first in simulation and then experimentally on the Daisy facility.

5.2.1 Background

Stability can be represented in both continuous time and discrete time. Figure 5.5 summarizes the relationship between continuous-time and discrete-time systems where system eigenvalues are plotted on real and imaginary axes. In the continuous-time domain, the unstable region is the entire right-half plane where the imaginary axis represents the stability boundary. This stability boundary is represented in discrete time by the unit circle.
Eigenvalues of the discrete-time system $\lambda_d$ are related to the eigenvalues of the continuous-time system $\lambda_c$ as follows:

$$\lambda_d = e^{\lambda_c \Delta t} \quad \lambda_c = \frac{\ln \lambda_d}{\Delta t}$$ (5.3)

where $\Delta t$ is the sampling period. Equivalent locations of a pair of stable eigenvalues are shown for illustration.

Figure 5.5: Stability Region for Continuous-Time and Discrete-Time Systems

5.2.2 Sources of Model Instability

If a physical system is inherently unstable then an identified model of such a system will likely be unstable. Even if a physical system is stable, the sources of model error discussed in § 2.3.1 may cause the SI algorithm to identify an unstable model. Consider, for example, the numerical round-off error
that occurs during the calculation-intensive SI process. To demonstrate this source of model instability, OKID is tested on the two-mass-spring-damper example shown in Figure 5.6.

![Figure 5.6: Two-Mass-Spring-Damper System Exhibiting Rigid Modes](image)

The simple mechanical system of Figure 5.6, being on the stability boundary, exhibits a rigid mode whose corresponding discrete-time eigenvalue is 1.0. The mass, stiffness and damping parameters are, respectively, \( m_1 = 2 \) kg, \( m_2 = 1 \) kg, \( k = 1 \) N/m, and \( c = 0.1 \) kg/s. The force \( u \) is the input and the displacements \( y_1 \) and \( y_2 \) are the outputs. A sample time of 0.2 seconds is used.

A random 30 second input signal is generated and the two-mass example is simulated to obtain input/output data used for SI. Setting the number of block rows in the Hankel matrix to be \( p = 2 \) ensures a system order of 4 (system order \( n = p \times \) number of outputs). OKID calculates observer Markov parameters from which the system Markov parameters are determined. For this example, the Markov parameters calculated by OKID match the desired impulse response up to the eleventh decimal position. Fast ERA is used to realize state-space discrete-time system matrices. Table 5.1 compares the
desired and realized modal parameters for the system. Note that negative damping indicates that the realized model is unstable.

Table 5.1: Modal Parameters for Two-Mass Mechanical System

<table>
<thead>
<tr>
<th>Desired % Damping</th>
<th>Desired Frequency (Hz)</th>
<th>Realized % Damping</th>
<th>Realized Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>0.00</td>
<td>-3.43 x 10^{-4}</td>
<td>8.95 x 10^{-7}</td>
</tr>
<tr>
<td>N/A</td>
<td>0.00</td>
<td>-3.43 x 10^{-4}</td>
<td>8.95 x 10^{-7}</td>
</tr>
<tr>
<td>6.12</td>
<td>1.95 x 10^{-1}</td>
<td>6.12</td>
<td>1.95 x 10^{-1}</td>
</tr>
<tr>
<td>6.12</td>
<td>1.95 x 10^{-1}</td>
<td>6.12</td>
<td>1.95 x 10^{-1}</td>
</tr>
</tbody>
</table>

The realized eigenvalues of the discrete-time system are compared with the desired eigenvalues as shown in Table 5.2. The first complex conjugate set of eigenvalues demonstrates that the realized model is only slightly unstable.

Table 5.2: Discrete-Time Eigenvalues: Two-Mass Mechanical System

<table>
<thead>
<tr>
<th>Desired Eigenvalues</th>
<th>Realized Eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>1.00 ± 1.11 x 10^{-6}</td>
</tr>
<tr>
<td>9.56 x 10^{-1} ± 2.38 x 10^{-1}</td>
<td>9.56 x 10^{-1} ± 2.38 x 10^{-1}</td>
</tr>
</tbody>
</table>

These results indicate that OKID and Fast ERA are capable of realizing systems with rigid modes. For this case, model instability is due to numerical processes during the analysis. These numerical processes include the
least-squares calculation (Equation (3.15)) in OKID as well as the eigenvalue decomposition function in Fast ERA (Equation (D.41)). The author has observed that the process of identifying lightly-damped flexible structures, also relatively close to the stability boundary, tends to yield unstable models. In general, any structure that has eigenvalues near the stability boundary risks being identified as unstable. In the following sections the author introduces three techniques which allow unstable models to be reduced using balanced reduction algorithms.

5.2.3 Stabilize the Model

The author has observed that if an identified model is unstable, it is often only slightly unstable. To quantify the degree of instability, the characteristic time for an unstable mode (the time for a mode to double its amplitude $t_{\text{double}}$) is calculated as follows:

$$t_{\text{double}} = \frac{\ln 2}{\sigma}$$  \hspace{1cm} (5.4)

where $\sigma$ is the real part of the continuous-time eigenvalue.

To test the severity of the model instability, the time to double in amplitude can be compared with the length of the SI experiment $l$. If $t_{\text{double}} \gg l$ then the identified model is only slightly unstable. Forcing a slightly unstable model to be stable allows the application of balanced model reduction algorithms.
To stabilize a state-space model, the author suggests performing an eigenvalue decomposition of the \( A \) matrix as follows:

\[
A E = E A
\]

(5.5)

where \( A \) is a diagonal matrix of eigenvalues and \( E \) is the eigenmatrix of eigenvectors. The unstable eigenvalues in \( A \) are modified so that their magnitudes are less than one — a process similar to the eigenvalue perturbation methods of Chui and Maciejowski [24]. The present author suggests that the approach shown in Figure 5.7 be used to modify an eigenvalue outside the stability boundary. Subtracting \( \Delta_{\text{real}} \) and \( \Delta_{\text{imaginary}} \) from the discrete-time unstable eigenvalue ensures that it lies within the unit circle. This stabilizing process is repeated for each unstable eigenvalue in \( A \).
Given this new diagonal matrix of eigenvalues \( \hat{\mathbf{A}} \) the new (stable) \( \hat{\mathbf{A}} \) matrix can be reconstructed as follows:

\[
\hat{\mathbf{A}} = \mathbf{E} \hat{\mathbf{A}} \mathbf{E}^{-1}
\]  

(5.6)

For the two-mass example discussed in § 5.2.2, Fast ERA realizes an unstable model. The unstable mode within this model corresponds to the rigid mode of the system and has a time to double its amplitude (\( t_{\text{double}} \)) on the order of \( 10^{10} \) seconds. Applying the above stabilization theory on the two-mass example of § 5.2.2 results in a stable model as desired. The prediction error (calculated according to Equation (3.54)) for the unstable model is \( 1.8 \times 10^{-9} \) while the prediction error for the stabilized model is \( 2.2 \times 10^{-9} \). Note that stabilizing the model tends to slightly increase the prediction error (a trend also observed with more complex structures such as Daisy in § 5.3.2).

### 5.2.4 Stabilize then Destabilize Model

To avoid these slight increases in model prediction error, the stabilization theory discussed above is now extended so that the reduced stable model can be re-destabilized.

Let the new, stabilized \( \hat{\mathbf{A}} \) matrix in Equation (5.6) be written as:

\[
\hat{\mathbf{A}} = \mathbf{A} + \Delta
\]  

(5.7)

where \( \Delta \) represent the perturbation required to make \( \mathbf{A} \) stable.
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

The stabilized $\hat{A}$ matrix is balanced as follows:

$$\hat{A}_S = \hat{S}\hat{A}\hat{S}^{-1}$$  \hspace{1cm} (5.8)

where the subscript $\hat{S}$ indicates the balanced form and $\hat{S}$ represents the balancing transformation matrix. The transformation matrix $\hat{S}$ may also be applied to the perturbation matrix $\Delta$:

$$\Delta_\hat{S} = \hat{S}\Delta\hat{S}^{-1}$$  \hspace{1cm} (5.9)

Note that applying the balancing transformation $\hat{S}$ (used to place $\hat{A}$ in balanced form) does not guarantee that the matrix $\Delta_\hat{S}$ will be in balanced form.

Let $A_S$ represent the balanced form of the unstable $A$ matrix:

$$A_S = SAS^{-1}$$  \hspace{1cm} (5.10)

where $S$ is the balancing transformation matrix. The present author claims that the balanced matrix $A_S$ can be approximated by

$$A_S \approx \hat{A}_\hat{S} - \Delta_\hat{S}$$  \hspace{1cm} (5.11)

The explanation is as follows:

$$A_S \approx \hat{A}_\hat{S} - \Delta_\hat{S} = \hat{S}\hat{A}\hat{S}^{-1} - \Delta_\hat{S} = \hat{S}(A + \Delta)\hat{S}^{-1} - \hat{S}\Delta\hat{S}^{-1} = \hat{S}A\hat{S}^{-1} + \hat{S}\Delta\hat{S}^{-1} - \hat{S}\Delta\hat{S}^{-1} = \hat{S}A\hat{S}^{-1}$$  \hspace{1cm} (5.12-5.16)
As $\Delta$ becomes small, the stable and unstable system matrices $\hat{A}$ and $A$ have similar balancing transformation matrices $\hat{S}$ and $S$ respectively. If $\hat{S}$ and $S$ are similar, then the expressions for $A_S$ in Equations (5.10), (5.11) and (5.16) are equivalent as desired.

Further approximations are made in the last step when the reduced unstable model $A_R$ is obtained from Equation (5.11) by truncating the appropriate rows and columns as follows:

$$A_R \approx \hat{A}_R - \Delta_R$$

Since $\Delta_S$ is not in balanced form, truncating rows and columns to obtain $\Delta_R$ will likely result in larger errors than if $\Delta_S$ had been in balanced form. If the realized models are slightly unstable (as is most often the case) then perturbation $\Delta$ will be small; therefore, any errors incurred because $\hat{S}$ and $S$ are not identical and $\Delta_S$ is not balanced will be minimal.

5.2.5 Partition Into Stable and Unstable Portions

A third method to reduce an unstable model is to convert the realized state-space matrices into modal coordinates following the procedure given in Appendix E.6. The system can then be partitioned into its stable and unstable portions as shown in Figure 5.8. Assuming that the unstable modes should not be removed, the balanced model reduction algorithms can be applied only to the stable portion.

The above three model order reduction techniques are applied to the Daisy structure in both simulation and experiment.
5.3 Daisy Simulation Results

OKID and Fast ERA are applied to data generated from a Daisy computer simulation. In the simulation, Daisy's hub has both stiffness and light damping in the roll and pitch directions. The motion of the hub in the yaw direction has only light damping. As discovered in § 5.2.2, if a structure is relatively close to the stability boundary then SI tends to yield unstable models. Daisy simulation results are first examined for the case where a stable model is realized. The amount of hub damping is then artificially reduced in the simulation. This modification moves the eigenvalues of the structure closer to the stability boundary and unstable models are realized.
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

5.3.1 Stable Model Realization

Daisy, having \( N = 23 \) degrees-of-freedom (DOF), can be modeled by a state matrix \( A \) of order \( n = 2N = 46 \). There are \( n \) states and \( n \) eigenvalues associated with the \( A \) matrix. For a flexible structure, the eigenvalues occur in complex conjugate pairs where each of these \( N \) pairs correspond to a mode of vibration. When Daisy is excited by the three reaction wheels, the \( n \) states (\( N \) different modes of vibration) are not all controllable. The rank of the controllability matrix \( Q \) is 18 where \( Q \) is defined as:

\[
Q = [B \ AB \ldots \ A^{n-1}B]
\]  

(5.18)

There are, therefore, only 9 controllable modes of vibration (each having a complex conjugate). An eigenvalue \( \lambda \) of \( A \) corresponds to a controllable mode if the rank of \( [A - \lambda I \ B] = n \). See [20].

These calculations indicate that the realized models obtained from OKID and Fast ERA should be of order 18 since only 9 modes are being controlled and, therefore, excited by the reaction wheels. Note that the Daisy simulator does not include non-idealities such as measurement errors, time-varying parameters or nonlinearities.

Setting the number of block rows in the Hankel matrix \( p \) to 8 ensures a maximum system order of \( n = 184 \). During the SI process, only the first 64 singular values are used since the remaining 120 singular values (184–64) are considered negligible (see § 5.1.1). The resulting model is, therefore, of order 64 and has an average prediction error of \( 5.0 \times 10^{-5} \). Balanced
model order reduction techniques are used to reduce the model to order $n = 18$ as desired. The average prediction error for this reduced model is $5.5 \times 10^{-5}$. The resulting damping and frequencies of the identified model are compared in Table 5.3 with the desired values from the simulation model.

There are ten frequencies and damping ratios shown in the table. The first two frequencies correspond to the motion of the hub in the yaw direction. This motion, as mentioned in § 5.3, has damping but no stiffness. There are no oscillations and, therefore, the eigenvalues do not appear in complex conjugate pairs. It takes two different eigenvalues to describe this motion.

Table 5.3: Modal Parameters from Daisy Simulation

<table>
<thead>
<tr>
<th>Desired % Damping</th>
<th>Desired Frequency (Hz)</th>
<th>Realized Damping</th>
<th>Realized Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>0.00</td>
<td>N/A</td>
<td>2.00 \times 10^{-7}</td>
</tr>
<tr>
<td>9.54 \times 10^{-4}</td>
<td>N/A</td>
<td>9.54 \times 10^{-4}</td>
<td>N/A</td>
</tr>
<tr>
<td>3.56</td>
<td>2.27 \times 10^{-2}</td>
<td>2.84</td>
<td>2.27 \times 10^{-2}</td>
</tr>
<tr>
<td>3.58</td>
<td>2.29 \times 10^{-2}</td>
<td>2.86</td>
<td>2.29 \times 10^{-2}</td>
</tr>
<tr>
<td>6.10</td>
<td>8.09 \times 10^{-2}</td>
<td>3.60</td>
<td>8.10 \times 10^{-2}</td>
</tr>
<tr>
<td>6.43</td>
<td>8.46 \times 10^{-2}</td>
<td>3.78</td>
<td>8.47 \times 10^{-2}</td>
</tr>
<tr>
<td>6.44</td>
<td>8.47 \times 10^{-2}</td>
<td>3.78</td>
<td>8.48 \times 10^{-2}</td>
</tr>
<tr>
<td>7.48</td>
<td>9.91 \times 10^{-2}</td>
<td>4.38</td>
<td>9.92 \times 10^{-2}</td>
</tr>
<tr>
<td>7.49</td>
<td>9.91 \times 10^{-2}</td>
<td>4.38</td>
<td>9.93 \times 10^{-2}</td>
</tr>
<tr>
<td>7.80</td>
<td>1.02 \times 10^{-1}</td>
<td>4.59</td>
<td>1.03 \times 10^{-1}</td>
</tr>
</tbody>
</table>
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

Table 5.4: Discrete-Time Eigenvalues for Daisy Simulation

<table>
<thead>
<tr>
<th>Desired Eigenvalue</th>
<th>Realized Eigenvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>9.99×10⁻¹</td>
<td>9.99×10⁻¹</td>
</tr>
<tr>
<td>9.99×10⁻¹ ± 1.43×10⁻²j</td>
<td>9.99×10⁻¹ ± 1.43×10⁻²j</td>
</tr>
<tr>
<td>9.99×10⁻¹ ± 1.44×10⁻²j</td>
<td>9.99×10⁻¹ ± 1.44×10⁻²j</td>
</tr>
<tr>
<td>9.96×10⁻¹ ± 5.06×10⁻²j</td>
<td>9.97×10⁻¹ ± 5.07×10⁻²j</td>
</tr>
<tr>
<td>9.95×10⁻¹ ± 5.29×10⁻²j</td>
<td>9.97×10⁻¹ ± 5.31×10⁻²j</td>
</tr>
<tr>
<td>9.95×10⁻¹ ± 5.29×10⁻²j</td>
<td>9.97×10⁻¹ ± 5.31×10⁻²j</td>
</tr>
<tr>
<td>9.93×10⁻¹ ± 6.18×10⁻²j</td>
<td>9.95×10⁻¹ ± 6.21×10⁻²j</td>
</tr>
<tr>
<td>9.93×10⁻¹ ± 6.18×10⁻²j</td>
<td>9.95×10⁻¹ ± 6.21×10⁻²j</td>
</tr>
<tr>
<td>9.93×10⁻¹ ± 6.38×10⁻²j</td>
<td>9.95×10⁻¹ ± 6.42×10⁻²j</td>
</tr>
</tbody>
</table>

and instead of a damping ratio, a damping factor is reported. The next two frequencies correspond to hub modes in the roll and pitch directions. The remaining modes involve rib motions.

Note that although the desired and realized frequencies are within 0.2% of each other, there is a rather large discrepancy between the desired and realized damping ratios. Each damping ratio is calculated by taking the negative of the real part of the continuous-time eigenvalue and dividing it by the magnitude of the eigenvalue. A comparison between the desired and realized discrete-time eigenvalues is shown in Table 5.4.
It is evident from Table 5.4 that small errors in the realized eigenvalues result in large errors in the damping ratio. Fassois et al. [40] support this finding and indicate that inaccuracies in modal parameter estimates occur during the discrete-to-continuous transformation. In the case of lightly-damped structures, Denman et al. [39] point out that the eigenvalues are difficult to compute accurately due to the logarithmic relationship described by Equation (5.3). If the real part of $\lambda_c$ in Equation (5.3) is close to zero (as in the case of lightly-damped structures), small errors in the identification of $\lambda_d$ can lead to relatively large errors in the computed values of $\lambda_c$. Table 5.5 demonstrates the rather large discrepancy between desired and realized continuous-time eigenvalues for the Daisy simulation. This discrepancy explains why the realized damping ratios in Table 5.3 (calculated using $\lambda_c$) have relatively large errors.

### 5.3.2 Unstable Model Realization

If hub damping in the roll, pitch and yaw directions is reduced then the realized model becomes unstable. For the cases tested, the unstable modes correspond to hub motion with $t_{\text{double}}$ on the order of six times the length of the experiment ($6 \times 240 \text{ seconds} = 1440 \text{ seconds}$).

The procedure for realizing a model using OKID and Fast ERA is described in detail in § 5.3.1. The number of block rows in the Hankel matrix is set to $p = 8$ yielding a maximum model order of 184. During the SI process, as in § 5.3.1, only the first 64 singular values are used.
Table 5.5: Continuous-Time Eigenvalues for Daisy Simulation

<table>
<thead>
<tr>
<th>Desired Eigenvalue</th>
<th>Realized Eigenvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>$-1.25 \times 10^{-6}$</td>
</tr>
<tr>
<td>$-6.00 \times 10^{-3}$</td>
<td>$-6.00 \times 10^{-3}$</td>
</tr>
<tr>
<td>$-5.08 \times 10^{-3} \pm 1.43 \times 10^{-1} j$</td>
<td>$-4.06 \times 10^{-3} \pm 1.43 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-5.16 \times 10^{-3} \pm 1.44 \times 10^{-1} j$</td>
<td>$-4.12 \times 10^{-3} \pm 1.44 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-3.10 \times 10^{-2} \pm 5.07 \times 10^{-1} j$</td>
<td>$-1.83 \times 10^{-2} \pm 5.08 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-3.42 \times 10^{-2} \pm 5.31 \times 10^{-1} j$</td>
<td>$-2.01 \times 10^{-2} \pm 5.32 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-3.43 \times 10^{-2} \pm 5.31 \times 10^{-1} j$</td>
<td>$-2.01 \times 10^{-2} \pm 5.32 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-4.66 \times 10^{-2} \pm 6.21 \times 10^{-1} j$</td>
<td>$-2.73 \times 10^{-2} \pm 6.23 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-4.66 \times 10^{-2} \pm 6.21 \times 10^{-1} j$</td>
<td>$-2.73 \times 10^{-2} \pm 6.23 \times 10^{-1} j$</td>
</tr>
<tr>
<td>$-5.02 \times 10^{-2} \pm 6.42 \times 10^{-1} j$</td>
<td>$-2.96 \times 10^{-2} \pm 6.44 \times 10^{-1} j$</td>
</tr>
</tbody>
</table>
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

The unstable, full order model has an average prediction error of $7.18 \times 10^{-5}$. If an unstable model is unacceptable, then the model can be stabilized using the theory discussed in § 5.2.3. The stabilized reduced model of order $n = 18$ has prediction errors on the order of $1.10 \times 10^{-2}$. Again, stabilizing the model tends to increase the prediction error. If an unstable model is acceptable, then balanced reduction can be applied using the theory developed in § 5.2.4. The resulting reduced unstable model of order $n = 18$ has an average prediction error of $7.39 \times 10^{-5}$ which is comparable to the full order unstable model. Alternatively, the unstable model can be partitioned as described in § 5.2.5 and the average prediction error of the reduced model is $7.40 \times 10^{-5}$. The prediction errors using the two approaches described in § 5.2.4 and § 5.2.5 are almost identical.

5.4 Daisy Experimental Results

We now test the three solutions to the model reduction problem using Daisy experimental data. Setting the number of block rows in the Hankel matrix to $p = 12$ ensures a maximum system order of 276. Fast ERA is used to realize the state-space discrete-time system matrices and the largest 260 singular values are selected to avoid matrix ill-conditioning problems.

The realized model is unstable with the unstable mode corresponding to the yaw hub motion. Again, the instability is slight with the corresponding eigenvalue having magnitude 1.0008. The author has observed that balanced reduction algorithms are numerically unreliable when model dimensions ex-
ceed 200. A model of order 260 must therefore be reduced in two steps:

- Reduce the model to order 180 by truncating the highest frequencies.
- Apply balanced reduction algorithms to reduce the model further.

Truncation of high frequencies should not be used exclusively since the reduced models may yield large prediction errors. Figure 5.9 compares the prediction error obtained from a full order model \((n = 260)\), frequency truncation to \(n = 180\), and frequency truncation to \(n = 46\). The prediction error is calculated as shown in Equation (3.54). The output channel numbers refer to the roll, pitch and yaw hub angular displacement, rib 1 “out of cone” (OC) tip displacement, rib 1 “in cone” (IC) tip displacement, rib 2 OC, rib 2 IC, . . . , and rib 10 IC. Frequency truncation to \(n = 180\) does not result in a significant increase in model prediction error.

Having reduced the realized unstable model from \(n = 260\) to \(n = 180\), the stabilizing theory discussed in § 5.2.3 can be applied. Plot A in Figure 5.10 shows the prediction errors of the stabilized reduced model. As expected, the prediction error increases when a model is stabilized. Note, however, that this increase is isolated to only the yaw hub angular displacement (Daisy output channel number 3).

Alternatively, the stabilizing theory described in § 5.2.4 can be applied to the unstable model of order \(n = 180\). The resulting prediction error is shown in plot B of Figure 5.10. Note that the slight increase in prediction error relative to the full order model is due only to the frequency truncation process.
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

Truncation of High Frequencies

![Graph showing prediction error comparison between full order and frequency truncation.]

Figure 5.9: Prediction Error Comparison: Truncation of High Frequencies
Figure 5.10: Prediction Error Comparison: Unstable Model Order Reduction
CHAPTER 5. HIGH-ORDER IDENTIFIED MODELS

from $n = 260$ to $n = 180$. Figure 5.11 shows a sample of the results obtained using the theory described in § 5.2.4.

The unstable model of order $n = 180$ can also be partitioned and reduced as discussed in § 5.2.5. The corresponding results (average prediction error $= 0.1565$) are virtually identical with those obtained using the stabilizing theory described in § 5.2.4 (average prediction error $= 0.1566$).
Sample of Unstable Model Results, n=46

Figure 5.11: Sample of Unstable Model Results
Chapter 6

Direct Low-Order Model Identification

Chapter 5 examined model order reduction of high-order models. As the identified model size increases, it becomes more numerically ill-conditioned. With increasing overspecification, numerical difficulties are encountered first in model order reduction and then in the actual SI procedure. Although balanced model reduction approaches yield promising results, the reduction process requires significant computational resources and imposes additional model stability requirements. Techniques are desired to reduce the amount of overspecification required by OKID and Fast ERA so that low-order, highly-accurate models can be identified without the numerical ill-conditioning difficulties and computationally-intensive model order reduction step. In this chapter, the present author introduces and tests an SI technique involving cubic smoothing splines that allows OKID with Fast ERA to directly identify low-order highly-accurate linear models. While the assumption of an accu-
rate linear model may not always be valid for certain kinds of structures, the author uses augmented SI to demonstrate that the linear system assumption is appropriate for the Daisy analyses performed in this thesis.

6.1 Cubic Smoothing Spline Approach

In the following sections the author proposes a new method of reducing and potentially eliminating overspecification with OKID and Fast ERA.

6.1.1 OKID Markov Parameters

Figure 4.2 demonstrates that OKID can calculate Markov parameters from general input/output data while Fast ERA uses these Markov parameters to identify state-space system matrices. The author has observed that the majority of the errors in the SI process occur during the OKID stage. Take, for example, an observer order of $p = 10$. Applying OKID on Daisy experimental data yields Markov parameters of dimension $23 \times 3$. Figures 6.1 through 6.4 plot the first 200 seconds of OKID’s Markov parameters. Note that for each input (3), the response from an impulse can be measured at each output (23). The responses are dimensionless since OKID is applied to normalized Daisy data. Figure 6.1 demonstrates how an impulse at the roll reaction wheel (RW) has, as expected, a significant influence on the hub roll response. Similar observations occur for the pitch and yaw impulse responses. Figure 6.4 exhibits significant rib IC motions which is expected for a yaw RW impulse.
Fast ERA uses these Markov parameters to identify corresponding $A$, $B$, $C$ and $D$ matrices. While the resulting model has a prediction error of 0.155, the difference between the impulse response of the identified model and OKID's Markov parameters is on the order of $10^{-4}$. OKID is, therefore, responsible for virtually all the prediction errors observed with the identified model.

This observation implies that estimates of the model prediction error can be obtained from OKID without having to identify a state-space model using algorithms such as Fast ERA. First calculate the observer Markov parameters $\tilde{Y}$ as shown in Equation (3.15). Then, instead of generating the system Markov parameters, simulate Equation (3.11) with the calculated observer Markov parameters to generate the predicted output $\hat{y}(k)$ as follows:

$$\hat{y}(k) = \sum_{i=1}^{k} \tilde{Y}_i v(k-i) + Du(k)$$

(6.1)

where

$$v = \begin{bmatrix} u \\ \hat{y} \end{bmatrix}$$

(6.2)

The predicted output $\hat{y}(k)$ can then be compared with the actual output $y(k)$. The resulting prediction error represents a lower bound on the SI process for a given observer order. Take, for example, an observer order of $p = 10$. The resulting prediction error using OKID's observer Markov parameters is 0.155 which agrees with the prediction error obtained with both OKID and Fast ERA. Similarly, an observer order of $p = 2$ has a prediction error using OKID's observer Markov parameters of 0.755 which compares well with the prediction error of 0.756 obtained with both OKID and Fast ERA.
Figure 6.1: OKID Markov Parameters for Daisy’s Hub
CHAPTER 6. DIRECT LOW-ORDER MODEL IDENTIFICATION

Impulse at roll RW

Figure 6.2: OKID Markov Parameters for Daisy's Ribs, Impulse at roll RW
Figure 6.3: OKID Markov Parameters for Daisy's Ribs, Impulse at pitch RW
Figure 6.4: OKID Markov Parameters for Daisy's Ribs, Impulse at yaw RW
CHAPTER 6. DIRECT LOW-ORDER MODEL IDENTIFICATION

Since model prediction error is determined primarily during the OKID step of the SI process, attention is focused on improving the quality of OKID's system Markov parameters. The level of overspecification is related to the observer order \( p \) used by OKID. Therefore, as the observer order increases, the identified model size increases and the model prediction error decreases. Figure 6.5 compares a sample of OKID Markov parameters obtained from Daisy experimental data for two different observer orders. As discussed above, the Markov parameters obtained using \( p = 10 \) produce more accurate models than with \( p = 2 \). In Figure 6.5, plot B magnifies plot A and shows high frequencies in the impulse response. Table 6.1 summarizes the transverse, torsional and longitudinal vibrations of one of Daisy’s ribs.

<table>
<thead>
<tr>
<th>First Mode of Vibration</th>
<th>Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transverse</td>
<td>70</td>
</tr>
<tr>
<td>Torsional</td>
<td>440</td>
</tr>
<tr>
<td>Longitudinal</td>
<td>720</td>
</tr>
</tbody>
</table>

Table 6.1: Daisy Rib Frequencies

Note that there is almost three orders of magnitude between Daisy’s 0.1 Hz system frequency and the 70 Hz rib frequency associated with the first bending mode. These high frequency rib dynamics have a negligible influence on Daisy’s response and the the high frequencies in OKID’s Markov parameters are primarily due to non-idealities in the data such as measurement errors, time-varying parameters, system nonlinearities, and computer round-off errors which are being modeled with large observer orders.
Using the Markov parameters for $p = 10$, Fast ERA realizes a model of order $n = 191$. Balanced model reduction yields a model of order $n = 32$ having an average prediction error of 0.16. The impulse response of this low-order (desired) model is compared with OKID's system Markov parameters for $p = 10$ in Figures 6.6 through 6.8. It is evident that the high frequencies in the OKID Markov parameters are not required to obtain accurate models.

6.1.2 Cubic Smoothing Spline

Low-pass filters could be applied to the Markov parameters to remove the high frequencies. Unfortunately, filtering tends to distort the response and introduces several parameters including filter order and cut-off frequency. Instead of filtering the data, the author proposes curve fitting as an approach to removing the high frequencies. Smooth piecewise polynomial functions, called splines, are applied to OKID's Markov parameters. Let the Markov parameters involved in the curve fit span the interval $[a...b]$. Then this interval is subdivided into smaller intervals $[\xi_j...\xi_{j+1}]$ such that $a = \xi_1 < ... < \xi_{l+1} = b$ where $l$ is the number of breakpoints describing the spline. On each interval $[\xi_j...\xi_{j+1}]$ a polynomial of low degree provides a good approximation to the local Markov parameters. The polynomial pieces also blend smoothly so that the resulting patched function has several continuous derivatives. To obtain a spline, the author employs the variational approach where splines are obtained as a best interpolant that minimizes a specific criterion. In particular, given the data $(x_i, y_i)$ with $x_i \in [a...b] \forall i$, the
Figure 6.5: OKID Markov Parameters for Daisy
Figure 6.6: Daisy's Hub Markov Parameters
Figure 6.7: Daisy's Rib OC Markov Parameters
Figure 6.8: Daisy's Rib IC Markov Parameters
cubic smoothing spline $s$ minimizes the cost function $C$

$$C = \eta \sum_{i=a}^{b} (y_i - f(x_i))^2 + (1 - \eta) \int_{a}^{b} \left( \frac{d^2 f(t)}{dt^2} \right)^2 dt$$  \hspace{1cm} (6.3)

over all functions $f$ with two derivatives where $\eta$ is called the smoothing parameter. The smoothing spline $s$ is a cubic spline with a breakpoint at every data point. Smoothing is achieved by selecting $\eta \in [0 \ldots 1]$ so that $s$ contains as much of the information and as little of the noise in the data as possible. Figure 6.9 demonstrates the effect of varying the smoothing parameter $\eta$. Selecting $\eta = 0$ yields a least-squares straight line fit while $\eta = 1$ gives the natural cubic spline interpolant. The term 'natural' specifies that the spline end conditions at the first and last data point have second derivatives equal to zero. A cubic spline is generated using the first 500 Markov parameters (50 seconds); however, only the first 10 seconds are shown. Reasonable values of $\eta$ appear to be between 0.7 and 0.9.

Results of a cubic smoothing spline fit to OKID's Markov parameters for $p = 10$ are shown in Figure 6.10. The first 500 Markov parameters are used to generate the spline. The high frequencies at the beginning of each impulse response are effectively eliminated while the curve fit still follows the rest of the response.

These cubic smoothing splines are compared with the low-order (desired) model of $n = 32$ in Figures 6.11 through 6.13. The first 500 system Markov parameters from OKID ($p = 10$) are curve fit using a smoothing parameter of $\eta = 0.9$. A total of 69 impulse responses (3 inputs $\times$ 23 outputs) are individ-
Figure 6.9: Effect of changing smoothing parameter $\eta$
Figure 6.10: Sample of spline curve fit
usually curve fit. The figures show reasonable hub, and excellent rib agreement between the calculated cubic spline and the desired ($n = 32$ model) impulse responses.

### 6.1.3 Daisy Experimental Results

Using 1000 system Markov parameters, the first 500 of which are curve fit using a smoothing parameter of $\eta = 0.9$, Fast ERA can identify a model of order $n = 46$ directly. The prediction error for this low-order model is 0.185. The resulting prediction errors for various values of $\eta$ are shown in Figure 6.14. As $\eta$ increases from 0.3 to 0.9 the resulting model prediction error decreases. A minimum prediction error occurs near $\eta = 0.9$ and after this point the prediction error begins to increase. This increase is expected since beyond $\eta = 0.9$ there is very little smoothing of the data and a model of order $n = 46$ is not sufficient to capture the high frequencies. Using the cubic smoothing spline with $\eta = 0.9$ in conjunction with OKID and Fast ERA, a model of order $n = 46$ can be directly identified having a prediction error of 0.185. To get equivalent results using standard SI, Figure 6.14 demonstrates that a model of order $n = 184$ would have to be identified and then reduced to $n = 46$. The total computational requirement for such a realization is approximately 630 MFLOP of which 410 MFLOP are used during the model reduction stage. The cubic smoothing spline approach can directly identify a model of order $n = 46$ with only 360 MFLOP. The computational requirements are reduced by 43% and approximately 95% of these computations are required by OKID to generate the observer and system Markov parameters. The curve fitting procedure requires only 2.3 MFLOP while Fast ERA uses
Figure 6.11: Spline Curve Fit Results: Hub
Figure 6.12: Spline Curve Fit Results: OC Ribs
Figure 6.13: Spline Curve Fit Results: IC Ribs
15 MFLOP. Focusing the intensive calculations on the OKID stage not only ensures accurate system Markov parameters, but also enables several low-order models to be identified with minimal computations. The prediction error of these low-order models can be minimized by, for example, systematically varying parameters such as \( \eta \). In this example, up to 16 different low-order models could be identified without exceeding the 630 MFLOP required for one low-order model identified using standard SI techniques.

Figures F.1 through F.23 of Appendix F demonstrate that the cubic smoothing spline is an integral part of the low-order high-accuracy model identification procedure. Plot A in each figure shows model performance using OKID
and Fast ERA with an observer order of \( p = 2 \). Although a model of order \( n = 46 \) is directly identified, model accuracy is poor. Plot B in each figure shows the results of using OKID with an observer order of \( p = 10 \) and then applying Fast ERA with \( q = 2 \). Again, a model of order \( n = 46 \) is directly identified but without introducing the cubic smoothing spline, the model is inaccurate. Plot C in each figure of Appendix F compares the actual and predicted response by generating system Markov parameters with OKID and \( p = 10 \), applying a cubic smoothing spline with \( \eta = 0.9 \), and using Fast ERA with \( q = 2 \) to directly identify an accurate model of order \( n = 46 \).

The author tested the cubic smooth spline approach on additional Daisy experimental data. The results shown in Figure 6.15 are comparable to the above analyses, demonstrating that this new approach to SI with OKID and Fast ERA is simple, efficient and robust.
Figure 6.15: Cubic smoothing spline results for additional Daisy experiments
6.2 Augmented vs. Standard SI

The SI methods examined in this thesis assume that the system to be identified can be represented by a linear state-space model described by Equations (2.1) and (2.2). It is important to realize that such an assumption may not always be valid for certain kinds of structures. In the case of Daisy, Hong et al. [53, 54] suggest that there are inherent nonlinearities such as hub friction, and time dependencies such as gyricity, which reduce the performance of SI techniques. To improve SI results, Hong et al. [54] augment the SI of the original linear system model with additional inputs which represent the effects of the nonlinearities (see Figure 6.16).

![Diagram](image.png)

Figure 6.16: Augmented SI

Daisy possesses significant friction associated with its hub. Hub friction in roll and pitch directions is produced by the rolling friction in the ball bearings that support Daisy's weight, while allowing Daisy to rotate about these axes. Daisy is supported by a low-friction air bearing about the yaw axis. Friction is a nonlinear function of hub angular speed. The present author models Daisy's friction as Coulomb friction. Figure 6.17 demonstrates the Coulomb friction profile where the friction force $F_v$ is positive or negative depending
on the direction of motion. Greenwood [45] represents Coulomb friction mathematically thus:

\[ F_r = K \text{sgn}(v) \]  

(6.4)

where \( K \) is the maximum frictional force magnitude, \( v \) represents speed and \( \text{sgn}(v) \) has the value \( \pm 1 \) depending on the sign of its argument. Note that augmented SI does not require knowledge of the actual magnitude of \( F_r \). OKID with Fast ERA implicitly identifies the magnitude as part of the SI process.

Figure 6.17: The force of Coulomb friction

Augmented SI of Daisy involves augmenting the original linear system by three additional inputs (one for each rotation axis) representing the effects of hub friction. Figure 6.18 demonstrates how measured hub angular displacement is differentiated to yield hub angular speed. Applying Equation (6.4) to hub angular speed yields the desired friction profile.
Daisy's Hub Yaw Angular Displacement

Daisy's Hub Yaw Angular Velocity

Daisy's Hub Yaw Coulomb Friction Profile

Figure 6.18: Generating the augmented SI friction profile
CHAPTER 6. DIRECT LOW-ORDER MODEL IDENTIFICATION

As discussed in § 2.3.1, excitation amplitude does not theoretically matter for a linear system. For a nonlinear system such as Daisy, low excitation levels cause nonlinear hub friction to dominate the measurements and Hong et al. [53, 54] demonstrate that augmented SI of Daisy can significantly improve model accuracy when compared with standard SI. In this section, the present author performs a series of Daisy experiments and observes that the levels of excitation used in this thesis are high enough to prevent nonlinear hub friction from dominating the measurements.

Figure 6.19 shows SI model prediction error vs. model size using eight different observer orders $p$ (which are directly related to the level of overspecification). Note that when using OKID, the overspecification level can only be in integer amounts. The prediction error (Equation (3.54)) is averaged over all 23 Daisy output channels. For each level of overspecification, two models are identified: augmented SI model and standard SI model. Each model is reduced to $n = 100$ using the balanced model reduction method of Moore [117]. The models are then successively reduced to $n = 24$ and the corresponding prediction errors are plotted. It is evident that, for a given overspecification level, augmented SI produces slightly lower prediction errors than standard SI. This small difference in prediction error demonstrates that the use of standard SI methods throughout this thesis is justified.

As discussed in §4.1.2, increasing observer order $p$ increases the size of the problem (larger, more complicated models), increases the computational and computer memory requirements and, therefore, increases the amount of time
required for SI. The present author has also observed that larger, more complicated models tend to have a higher potential for numerical problems such as matrix ill-conditioning. In spite of these problems, increasing \( p \) also tends to reduce the model prediction error for both augmented and standard SI.

Consider a comparison of the model prediction error for reduced models of size \( n = 100 \). These prediction errors are projected in Figure 6.19 and are connected by a solid line. The thick dashed lines on this projection demonstrate that the model prediction error of augmented SI for an observer order of \( p = 8 \) is approximately the same as standard SI for \( p = 9 \) orders of overspecification. Similarly, augmented SI for an observer order of \( p = 10 \) has the same prediction error as standard SI for \( p = 12 \) orders of overspecification. Although the difference in prediction error between augmented and standard SI is small, it represents a difference of 23 and 46, respectively, in the realized model size. For a given tolerable prediction error, augmented SI requires less overspecification than standard SI. Smaller, less complicated models tend to have a lower potential for numerical problems such as matrix ill-conditioning.

Figure 6.19 also demonstrates that reducing the model order from \( n = 100 \) to \( n = 46 \) has a negligible effect on the prediction error. As the model order \( n \) goes below 46, the prediction error increases dramatically. The sudden increase in prediction error near \( p = 46 \) is consistent with the physical system since Daisy, having \( N = 23 \) DOF, can be modeled by a state-space system matrix \( A \) of order \( n = 2N = 46 \). There are \( n \) states and \( n \) eigenvalues of the \( A \) matrix. The eigenvalues occur in complex conjugate pairs and each
Figure 6.19: Standard vs. Augmented SI: Daisy Experimental Results
of these $N$ pairs correspond to a mode of vibration. Note that when the Daisy structure is excited by only the three reaction wheels, the $n$ states ($N$ different modes of vibration) are not all controllable (hence excitable). The controllability analysis of § 5.3.1 revealed that, assuming perfect structural alignment and complete symmetry, Daisy's reaction wheels would be able to excite only 9 modes of vibration. The realized models obtained from OKID and Fast ERA should be, therefore, of order 18. Figure 6.19 shows that poor results are obtained as models are reduced below $n = 46$. Evidently, non-idealities in Daisy cause the reaction wheels to excite more modes than predicted. Hong [55] has demonstrated that there is, for example, spillover of Daisy's out-cone response into the in-cone response for some ribs due to misaligned rib springs. This observation violates the assumption that OC and IC rib motions are independent and explains why more than 9 modes of vibration are excited by the reaction wheels in practice.
Chapter 7

Conclusion

This thesis establishes appropriate SI methods for on-orbit modeling of lightly-damped flexible spacecraft. Key to the selection process are twelve selection criteria used to quantitatively evaluate each SI method. Methods are selected in two stages. The first stage employs existing SI literature to establish that OKID with ERA, Q-Markov CovER, ORSE and Subspace are appropriate methods. The second stage examines these algorithms in detail and tests them on Daisy. These tests reveal that overspecification is a problem common to all selected methods. Highly-accurate models are obtained at the expense of identifying impractically large models.

The conflicting requirement of low model order and high accuracy necessitates the use of model order reduction algorithms. Model reduction based on balanced reduction techniques is shown to be more accurate than model reduction based on singular values of the Hankel matrix $H_j$, or frequency-based model reduction. Balanced model reduction algorithms, however, re-
quire that the models to be reduced be stable — a criterion not guaranteed by many identification algorithms. Three new unstable model order reduction techniques are proposed and examined, first in simulation and then experimentally on Daisy. It is shown that accurate, low-order unstable models can be achieved using existing balanced model reduction algorithms.

The model overspecification and reduction process imposes significant computational resource requirements as well as model stability requirements. Also, as the identified model is further overspecified, it becomes more numerically ill-conditioned. With increasing overspecification, numerical difficulties are encountered first in model order reduction and then in the SI procedure. The selected SI methods are extended by reducing the level of overspecification required and thus their performance is improved.

A new identification approach uses cubic smoothing splines to filter out high frequencies in OKID’s system Markov parameters. Applying this smoothing procedure early in the identification process allows low-order accurate models to be directly identified. Significant computational savings as well as numerical robustness result.

Augmented SI of Daisy involves augmenting the original linear system by three additional inputs (one for each rotation axis) representing the effects of hub friction. Experiments on Daisy using augmented SI demonstrate that the levels of excitation used throughout this thesis are high enough to prevent nonlinear hub friction from dominating the measurements. The use of standard SI approaches on Daisy is, therefore, appropriate.
CHAPTER 7. CONCLUSION

Given these extensions, the recommended methodology for SI of flexible spacecraft is summarized in Figure 7.1.

7.1 Summary of Contributions

The contributions of this research are summarized below.

- **New approach to reducing model overspecification with OKID and ERA**
  - Extended OKID with ERA to allow accurate low-order models to be directly identified using cubic smoothing splines. No model order reduction step was required, increasing the efficiency and reliability of the SI process.

- **New Model Order Reduction Approach**
  - Demonstrated that lightly-damped structures risk being identified as unstable.
  - Extended balanced model-order reduction methods to allow the reduction of unstable models. The approaches used eigenvalue perturbations to stabilize the identified model. The effects of these perturbations can then be removed, resulting in low-order highly-accurate models.
CHAPTER 7. CONCLUSION

Experimental Data

Calculate System Markov Parameters with OKID

Apply Cubic Smoothing Spline

Fast ERA SI

Modify Smoothing Parameter

Calculate Model Prediction Errors

satisfactory

unsatisfactory

Final Balanced Model Reduction using stability extension if required

low-order highly-accurate model

Figure 7.1: Recommended Methodology for System Identification of Flexible Spacecraft
• Which SI algorithms are appropriate for flexible spacecraft
  
  – Reviewed over 180 papers to assess the current state of the art in the SI field.
  
  – Compared time- and frequency-domain SI techniques and selected the time domain because of the relatively short data records available from Daisy.
  
  – Divided the SI field into the following categories:
    * Model representations
    * Estimators
    * Calculating Markov parameters
    * Calculating covariance parameters
    * Time-domain SI techniques
  
  – Established selection criteria to evaluate the suitability of SI methods for on-orbit modeling of lightly-damped flexible spacecraft. Using these criteria, a score was given to each algorithm and the highest ranking algorithms were selected for further testing.

• How selected SI algorithms perform on Daisy

  – Clarified the relationships between the selected SI algorithms using a common mathematical notation. In particular, the mathematical relationship between Subspace and ORSE algorithms was derived.

  – Performed computer simulations and experiments on the unique Daisy UTIAS spacecraft emulator. Using these data, the selected algorithms were tested and OKID with Fast ERA was preferred.
• How Augmented SI reduced model overspecification
  – Demonstrated that even relatively small improvements in model accuracy can result in substantial differences in the identified model size.

• SI computer software
  – Created a user-friendly computer software platform to implement and test selected SI algorithms (see Appendix E). The software also incorporated the various SI algorithm extensions discussed above.

7.2 Research Opportunities and Extensions

Augmented SI focuses on augmenting the original linear system with additional inputs representing nonlinear effects. The technique, therefore, has the potential to work with any SI algorithm — not only those discussed in this research.

Liu and Miller [97] introduce an integrated state-space model identification approach using ORSE, balanced model reduction and a least-squares (LS) model updating algorithm. The ORSE algorithm is applied to experimental data and a high-order, accurate model is realized. The next step involves a LS model updating algorithm which, using the identified model as a first approximation, updates the model assuming that first order sensitivities are applicable. The updated model is subsequently reduced using balanced model
reduction and the updating process is repeated. This iterative technique continues until the prediction error increases at which time a final low-order, highly-accurate model is obtained.

Not only is the above integrated approach applicable to other SI algorithms such as OKID and Fast ERA, but it can also be incorporated into the cubic smoothing spline approach developed in this research. Instead of identifying an extremely large model as a first approximation, the cubic smoothing spline approach with OKID can be used to directly identify a low-order model as shown in Figure 7.2. The remaining LS model updating and balanced model reduction process will further increase model accuracy and decrease model size without the computational requirements and numerical ill-conditioning problems associated with high-order models.

While the cubic smoothing spline technique is limited to OKID, any SI algorithms using Markov parameters as inputs (not just Fast ERA) can be used to identify the state-space model. Furthermore, this extension to OKID is applicable to other flexible structures.
CHAPTER 7. CONCLUSION

Figure 7.2: Integrated Identification Technique using Cubic Smoothing Splines
Appendix A

Acronyms

AR AutoRegressive
ARMA AutoRegressive Moving Average
ARMAX AutoRegressive Moving Average with eXogenous variable
ARX AutoRegressive with eXogenous variable
BJ Box-Jenkins
CVA Canonical Variate Analysis
DOF Degrees Of Freedom
DSPI Direct System Parameter Identification
ERA Eigensystem Realization Algorithm
ERA C Eigensystem Realization Algorithm using the Concurrent method
ERA c Eigensystem Realization Algorithm using the consecutive method
ERA/DC Eigensystem Realization Algorithm with Data Correlations
ERA-FD Eigensystem Realization Algorithm in the Frequency-Domain
FEM Finite Element Method
FFT Fast Fourier Transform
APPENDIX A. ACRONYMS

FIR  Finite Impulse Response
FLOP Floating Point OPerations
FORSE Frequency-domain Observability Range Space Extraction
FRF  Frequency Response Function
FSS  Flexible Space Structure
FTF  Fast Transversal Filter
GIV  Generalized Instrumental Variables
GUI  Graphical User Interface
GWN  Gaussian White Noise
HQMC Hankel approximation to Q-Markov Covariance Equivalent Realization
IDFT Inverse Discrete Fourier Transform
IMAC International Modal Analysis Conference
ITD  Ibrahim Time-Domain
JPL  Jet Propulsion Laboratory
LMFD Left Matrix-Fraction Description
LS  (Linear) Least Squares
LSCE Least Squares Complex Exponential
LSMBT Least Squares Moving Block Technique
LTI  Linear Time-Invariant
MACE Middeck Active Control Experiment
MDVV Moonen, DeMoor, Vandenberghe and Vandewalle
MIMO Multi-Input Multi-Output
MISO Multi-Input Single-Output
MLE Maximum Likelihood Estimation
APPENDIX A. ACRONYMS

MME Minimum Model Error
MODE Middeck 0-gravity Dynamics Experiment
MOESP Multi-input and multi-output Output Error State Space
N4SID Numerical algorithms for Subspace State-Space System Identification
OCID Observer/Controller Identification
ODE Ordinary Differential Equation
OE Output Error
OKID Observer Kalman filter Identification
ORSE Observability Range Space Extraction
PDF Probability Density Function
PEM Prediction Error Method
PRBS Pseudo-Random Binary Signal
Q-Markov CovER Q-Markov Covariance Equivalent Realization
RQ decomposition with Q orthonormal and R lower triangular
RLS Recursive Least Squares
RMA Reference Model Adaptation
RMFD Right Matrix-Fraction Description
SERC Space Engineering Research Center
SI System Identification
SIMO Single-Input Multi-Output
SISO Single-Input Single-Output
SNR Signal-to-Noise Ratio
SSFD State-Space Frequency Domain
SSRA State-Space Realization Algorithm
APPENDIX A. ACRONYMS

SVD Singular-Value Decomposition
TDPI Time-domain Direct Parameter Identification
TPBV Two Point Boundary Value problem
UTIAS University of Toronto Institute for Aerospace Studies
Appendix B

System Identification Selection

B.1 Criteria for Selecting an Algorithm

The present author suggests that the following criteria be used when choosing an appropriate SI algorithm. These criteria are selected based on the literature review conducted by the present author as well as suggestions by Greselda and Mook [46] and Nguyen [118].

B.1.1 Technical Criteria

The following nine criteria relate to the technical characteristics of the SI algorithm.

- **Control-Model SI:** The ability of the SI algorithm to perform the desired type of identification. This thesis focuses on generating an input/output description of a system.
  
  - Award 10 points if the algorithm directly identifies discrete-time
state-space system matrices $A$, $B$, $C$, and $D$.

- Subtract 4 points for additional steps required to achieve state-space matrices.

- **Excitation**: The type of excitations required by the SI algorithm such as step, white noise, sinusoidal or impulsive inputs.
  
  - Award 10 points if the signal is physically possible and simple to implement.
  
  - Subtract up to 8 points if the signal must be approximated or if the existing actuators (or sensors) are not appropriate for the desired signal (or response).
  
  - Award 10 points for band-limited random signals.
  
  - Award 4 points for an impulse signal.

- **Initial Conditions**: The need for the SI algorithm to know the initial conditions of the structure.
  
  - The less information about initial conditions required for the SI method to work properly, the better.
  
  - Award 10 points if no knowledge of the initial conditions is required.
  
  - Award 0 points if all initial conditions must be known and satisfied (the system must be at rest at the beginning of the data set).

- **Iterations**: The number of iterations required by the SI algorithm.
- Award 10 points if no iterations are required.
- Award 8 points if the only iterations involved are those associated with an algorithm such as a singular value decomposition (SVD).

**Computer Requirements:** The amount of computer resources required by the SI algorithm.

- Significant computation and memory requirements are to be avoided to achieve fast SI results (such as calculation of Markov and Covariance parameters).
- The execution time for all methods is not available from the literature and much depends on the computer system employed.
- Award 10 points for an algorithm having low computational requirements.
- Use ERA as a reference and award it a score of 6.

**Model Order:** The size of the realized model required to get accurate results.

- Accuracy is judged by comparing the SI generated (predicted) outputs with real system (actual) outputs.
- Award 10 points if the model is very accurate with a small model order.
- Award 5 points if accurate results are achieved with a large model order.
- Use ERA as a reference and award it a score of 5.
• **Numerical Problems:** Difficulties associated with manipulation of ill-conditioned matrices or convergence problems.
  
  - Award 10 points if no numerical problems are known.
  - Subtract up to 8 points if convergence problems are known.

• **Noisy Data:** The sensitivity of the algorithm to the presence of measurement noise.
  
  - Experimental data are expected to contain noise.
  - Incorrect models may be identified with noisy data.
  - Award 10 points if the algorithm has good noise suppression capability.
  - Award 5 points if it is not known how sensitive the algorithm is.
  - Use ERA as a reference and award it a score of 8.

• **Number of Parameters:** The number of user-selectable parameters required by the SI algorithm.
  
  - Algorithms having several adjustable parameters tend to be more flexible and may allow fine tuning of the realized models.
  - If the relationship between changes in the parameters and the realized model is well-understood, then it is desirable to have several parameters available to the user.
  - Award 2 points for each well-understood parameter.
APPENDIX B. SYSTEM IDENTIFICATION SELECTION

- Points are awarded for any parameters in addition to the following SI parameters:

  * $l =$ number of data points
  * $m =$ number of outputs
  * $r =$ number of inputs
  * $\Delta t =$ sample time (s)

B.1.2 Practicality Criteria

The following three criteria relate to the implementation and testing of the SI algorithm.

- **Software Availability:** The existence of user-friendly software for a particular SI method can be a good indication of the general acceptance and reliability of the algorithm.
  - Award 10 points for a software package that has been verified and has a user-friendly interface with documentation.
  - Subtract 1 point for each known software problem.
  - Subtract 1 point if the user interface needs improvement.
  - Award 0 points if no software is readily available and must be written based on the SI literature.

- **Documentation Availability:** Complete descriptions of the SI algorithms are necessary to make meaningful selections.
  - Award 10 points if the algorithm is fully described.
Experience: Algorithms that have performed well when tested on a variety of FSS are preferred.

- Award 10 points if the SI algorithm has been tested on a variety of FSS.
- Award 5 points if the SI algorithm works well when tested on simulations of FSS.
- Award 0 points if it is not known if the algorithm has been tested on FSS.

The comparisons and analyses in the following sections employ the same categories shown in Figure 2.2. For each category, a table is set up with criteria as the columns, and different SI model representations or methods as the rows. For each table note that the numbers in square brackets [ ] correspond to the references listed at the end of this thesis. Numbers in round brackets ( ) refer to the footnotes listed at the bottom of each table. Shaded regions indicate criteria that are not applicable to the particular method. Also note that the list of acronyms provided in Appendix A may be helpful. If information is unavailable, then a question mark (?) is entered into that field. A score may still be given to these empty fields to allow as many of the criteria to be included in the comparison as possible. Note, however, that such a score represents the present author's own opinion based on the knowledge gained in researching the SI method. Wherever possible, justification for the score is given and referenced.
Having allotted as many scores as reasonably possible, the total scores for each SI method are determined. Unweighted and weighted scores are given for each method. The unweighted score treats all criteria equally (a weight of 1.0) while the weighted score places emphasis on specific criteria. Software availability and experience are considered important and are given a weight of 2.0. Both of these criteria provide a good indication of the general acceptance of the SI method. The number of available parameters is given a weight of 0.5 since an algorithm may have very few parameters and still be a suitable method for FSS. It should be noted that a low score does not imply that the corresponding method is unsuitable for all applications. It represents the appropriateness of the method to on-orbit SI of flexible structures such as Daisy in relation to other methods examined.

B.2 Selection Results

Model Representation

There are several different mathematical SI model representations available. Appendix B.3 summarizes the properties of each model representation and includes an evaluation of the applicability of each representation to FSS. Also note that in the remarks column, any functions refer to MATLAB functions in the System Identification Toolbox [103]. For example, the arx function in MATLAB will use a LS estimation algorithm to fit MIMO data to an ARX model structure.
Based on Appendix B.3 it is evident that the following state-space models are preferred to the other models studied in the survey:

- ARX
- ARMAX
- state-space

Models such as ARARMAX and BJ perform well technically but these more complex models have limited practical applications.

ARX models have proven successful with OKID while state-space models, commonly used in controller design, are obtained with methods such as ERA and ORSE. Identification of ARMAX models requires numerically intensive procedures such as PEM since ARMAX is a more general model structure than ARX.

Estimators

Appendix B.4 summarizes and evaluates the variety of methods available to estimate parameters. The comparison demonstrates that although many algorithms exist, those that rank the highest tend to be the most reliable, simple and efficient algorithms. These features are critical to on-orbit SI of FSS. Issues, such as non-guaranteed convergence and long data histories, present high risks in environments where expense and limited time are constraints.

- Least Squares (LS) is selected as the most appropriate estimator for FSS.
The present author verified that PEM with an ARMAX model performs well on simple two-mass-spring-damper computer simulations but requires impractically large computational resources for structures such as Daisy.

**Markov Parameters**

Markov parameters are the discrete-time sequence of impulse response measurements from the system. Note that in discrete time, the impulse signal is that signal having a value of 1 at time $k = 0$ and a value of 0 everywhere else. Markov parameters are important to several time- and frequency-domain SI algorithms. A variety of methods is available to calculate these parameters as shown in Appendix B.5.

- **OKID** (forward) appears to be the most appropriate method for lightly-damped FSS.

Although OKID backward is based on an unnatural model where the current system state $x(k)$ is determined by the future system state $x(k + 1)$ and current input $u(k)$, it can be useful when trying to distinguish system modes from noise modes. OCID is recommended when Markov parameters of a closed-loop system are desired.

In addition to OKID, the present author tested other methods for calculating Markov parameters including white noise, FFT/IDFT, and a direct input/output matrix approach. A simple two-mass-spring-damper computer simulation was sufficient to demonstrate that these methods are inferior to OKID.
Covariance Parameters

Appendix B.6 evaluates various methods for obtaining covariance parameters. Covariance parameters are the sequence of data correlations of Markov parameters and are required by methods such as Q-Markov CovER.

- The stochastic as well as the deterministic approach seem to offer the best results.

Time-Domain System Identification

Appendix B.7 compares different time-domain SI techniques. The evaluation indicates that the following methods should be studied further:

- ERA
  - ERA/DC
  - Fast ERA
- Q-Markov CovER
- ORSE
- Subspace

The following sections summarize the detailed selection results for each SI category.
B.3 Model Representations
## APPENDIX B. SYSTEM IDENTIFICATION SELECTION

<table>
<thead>
<tr>
<th>Control-Model SI</th>
<th>Initial Conditions</th>
<th>Excitation</th>
<th>Model Representation:</th>
</tr>
</thead>
</table>
| LTI, causal      | yes               | yes        | yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, yes → transfer function, y
<table>
<thead>
<tr>
<th>Model Representation:</th>
<th>Number of Parameters</th>
<th>Software Availability</th>
<th>Documentation Availability</th>
<th>Experience</th>
<th>Remarks</th>
<th>Unweighted Score</th>
<th>Weighted Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>in addition to:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1, m, r, A, t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Markov Parameters</td>
<td># MP</td>
<td>MATLAB</td>
<td>[72 Sec. 2.3, 102]</td>
<td>significant MIMO</td>
<td>10</td>
<td>33</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>n_b</td>
<td>MATLAB</td>
<td>[102, 152]</td>
<td>(1)</td>
<td>2</td>
<td>43</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AR</td>
<td>MATLAB</td>
<td>[102, 103, 152]</td>
<td>time series</td>
<td>8</td>
<td>36</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>n_d, n_b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARX</td>
<td>MATLAB</td>
<td>[72, 102, 103, 152, 180]</td>
<td>significant MIMO</td>
<td>10</td>
<td>52</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>n_d, n_b, n_c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARMAX</td>
<td>MATLAB</td>
<td>[86, 102, 103, 152]</td>
<td>- arma (PEM) MISO</td>
<td>10</td>
<td>48</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>n_d, n_b, n_c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARMA</td>
<td>MATLAB</td>
<td>[3, 41, 86, 92, 102, 152]</td>
<td>- arma (PEM) MISO</td>
<td>10</td>
<td>38</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>n_d, n_c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARIMA</td>
<td>n_d, n_c</td>
<td>[102, 152]</td>
<td>econometrics</td>
<td>4</td>
<td>32</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARAX</td>
<td>n_d, n_b, n_d</td>
<td>[102, 152]</td>
<td>?</td>
<td>1</td>
<td>34</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARARMAX</td>
<td>n_d, n_b, n_c, n_d</td>
<td>[102, 152]</td>
<td>?</td>
<td>1</td>
<td>35</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OE</td>
<td>n_d, n_f</td>
<td>[64, 102, 103, 152]</td>
<td>- oe (PEM) MISO</td>
<td>4</td>
<td>39</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BJ</td>
<td>n_d, n_f, n_r, n_d</td>
<td>[102, 103, 152]</td>
<td>- bj (PEM) MISO</td>
<td>8</td>
<td>41</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>static-Space</td>
<td>n</td>
<td>[72 Sec. 2.2, 102, 103]</td>
<td>significant MIMO</td>
<td>10</td>
<td>55</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ODE</td>
<td>DOF N</td>
<td>[39, 83, 137]</td>
<td>significant MIMO</td>
<td>10</td>
<td>36</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prony Analysis</td>
<td>n</td>
<td>[161]</td>
<td>weighted linear combination of exponential terms [161]</td>
<td>5</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(1) signal processing applications [102]
(2) systems on the order of \( n = 10 \) [39]
(3) the MATLAB function \texttt{armax} is a SISO function employing a recursive PEM
(4) computer simulations [161]
(5) standard tool in controls, use when poles of stochastic and deterministic part of model are known to be identical [102]
(6) moving average of white noise error term [102] \( C(q) e(t) \)
(7) autoregression of white noise error term [102] \( D(q) e(t) \)
(8) ARMA description of white noise error term \( C(q) D^{-1}(q) e(t) \)
(9) use if the additive noise is white (noise model has no dynamics) [64]
(10) prediction errors are linear in the parameters, a unique global minimum can, therefore, be determined
(11) Gaussian white noise (GWN), it is common to assume that \( e(t) \) is GWN [102]
(12) useful for describing drifting disturbances, e.g. low-frequency noise, non-zero mean data [152]
(13) \( B(q) F^{-1}(q) \) and \( C(q) D^{-1}(q) \) have no common parameter, it will be possible to estimate \( B(q) F^{-1}(q) \) consistently even if the parameterization of \( C(q) D^{-1}(q) \) is not appropriate [152]

\[ l = \text{number of data points} \]
\[ m = \text{number of outputs} \]
\[ r = \text{number of inputs} \]
\[ \Delta t = \text{sample time (s)} \]
\[ n = \text{model order to be identified} \]
\[ N = \text{number of DOF in the model (} n = 2N) \]
\[ MP = \text{number of Markov parameters used to describe the model} \]
\[ n_* = \text{order of the polynomial } A(z) \]. The parameters
\[ n_b, n_f, n_r, n_d \] have similar definitions.
B.4 Estimators
<table>
<thead>
<tr>
<th>Estimators:</th>
<th>Control-Model SI</th>
<th>Excitation Initial Conditions</th>
<th>Iterations</th>
<th>Computer Requirements</th>
<th>Model Order</th>
<th>Numerical Problems</th>
<th>Noisy Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td></td>
<td></td>
<td>2</td>
<td>similar to MLE [102]</td>
<td>2</td>
<td>similar to MLE [102]</td>
<td>probabilistic context</td>
</tr>
<tr>
<td>MLE</td>
<td></td>
<td>several [75]</td>
<td>(2)</td>
<td></td>
<td>2</td>
<td>unbiased [39], convergence not guaranteed</td>
<td>(9)</td>
</tr>
<tr>
<td>Covariance matching</td>
<td></td>
<td>(15)</td>
<td>2</td>
<td>?</td>
<td>1</td>
<td>(8)</td>
<td>2</td>
</tr>
<tr>
<td>Correlation matching</td>
<td></td>
<td>(15)</td>
<td>4</td>
<td>long data histories [72]</td>
<td>2</td>
<td>convergence guaranteed [112]</td>
<td>good performance [28]</td>
</tr>
<tr>
<td>LS</td>
<td></td>
<td>pinv, SVD</td>
<td>4</td>
<td>efficient, simple order $n^2$</td>
<td>3</td>
<td>robust [64], (6) biased estimation</td>
<td>(5), (7)</td>
</tr>
<tr>
<td>GIV</td>
<td></td>
<td>similar to LS [102]</td>
<td>8</td>
<td>&gt;LS (instrumental variable)</td>
<td>5</td>
<td>no biased estimation</td>
<td>(6)</td>
</tr>
<tr>
<td>PEM</td>
<td></td>
<td>several [64]</td>
<td>4</td>
<td>(13)</td>
<td>1</td>
<td>(14)</td>
<td>2</td>
</tr>
<tr>
<td>MME</td>
<td></td>
<td>TGBP [142], several [116]</td>
<td>4</td>
<td>&gt;LS</td>
<td>3</td>
<td>few reported</td>
<td>robust [115]</td>
</tr>
<tr>
<td>Estimators:</td>
<td>Number of Parameters</td>
<td>Software Availability</td>
<td>Documentation Availability</td>
<td>Experience</td>
<td>Remarks</td>
<td>Unweighted Score</td>
<td>Weighted Score</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------</td>
<td>-----------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>---------</td>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Bayesian</td>
<td>(1)</td>
<td>no</td>
<td>[60, 72, 102, 112, 152]</td>
<td>3 mass simulation [60]</td>
<td>-PDF seldom available in practice</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>MLE</td>
<td>Xmath</td>
<td>10</td>
<td>[10, 39, 62, 64, 72, 75, 87, 112, 123, 124, 131, 138]</td>
<td>(10), (4)</td>
<td>-maxlike function (MIMO) -for linear system, Gaussian noise, MLE same as LS</td>
<td>37</td>
<td>55</td>
</tr>
<tr>
<td>Covariance matching</td>
<td>(11)</td>
<td>no</td>
<td>[72, 112]</td>
<td>?</td>
<td>2</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>Correlation matching</td>
<td>(12)</td>
<td>no</td>
<td>[28, 72, 102, 112]</td>
<td>time series analysis [112]</td>
<td>-eliminates bias associated with LS [28]</td>
<td>33</td>
<td>38</td>
</tr>
<tr>
<td>LS</td>
<td>Xmath</td>
<td>10</td>
<td>[64, 67, 72, 102, 103, 131]</td>
<td>significant, (3), (4)</td>
<td>-straightforward for ARX</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GIV</td>
<td>Xmath</td>
<td>10</td>
<td>[64, 102, 103, 131]</td>
<td>(4)</td>
<td></td>
<td>57</td>
<td>77</td>
</tr>
<tr>
<td>PEM</td>
<td>MATLAB</td>
<td>10</td>
<td>[64, 102, 103]</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>MME</td>
<td>no</td>
<td>10</td>
<td>[115, 116, 142]</td>
<td>simulated beam [115]</td>
<td>3</td>
<td>33</td>
<td>36</td>
</tr>
</tbody>
</table>
(1) more information required than MLE
(2) significant computations [72], complicated to implement
(3) SERC interferometric testbed - a truss-work tetrahedron 3 meters on a side [67]
(4) 5th order butterworth filter, impedance of a 3.4 MW synchronous motor, flight-flutter analysis [131]
(5) no prior noise knowledge required [131]
(6) a global minimization procedure exists [131]
(7) medium sensitivity to noise [131]
(8) convergence not guaranteed
(9) If noise is Gaussian noise then MLE is called Markov Estimation, prior knowledge of the noise is required, sensitive to noise [131]
(10) JPL flexible beam experiment [138], FEM parabolic reflector
(11) requires approximations of covariance parameters
(12) requires approximations of correlation parameters
(13) significant computations [64]
(14) nonlinear numerical problem
(15) similar to LS [112]

n = parameter specifying the model order used by the estimator
B.5 Markov Parameters
<table>
<thead>
<tr>
<th>Calculating Markov Parameters</th>
<th>Control Model SI</th>
<th>Excitation</th>
<th>Initial Conditions</th>
<th>Iterations</th>
<th>Computer Requirements</th>
<th>Model Order</th>
<th>Numerical Problems</th>
<th>Noisy Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>impulse</td>
<td>impulse</td>
<td>rest</td>
<td>none</td>
<td>4</td>
<td>memory</td>
<td>10</td>
<td>none reported</td>
<td>noise masks</td>
</tr>
<tr>
<td>FFT/IDFT</td>
<td>rest [72]</td>
<td>windows</td>
<td>10</td>
<td>0</td>
<td>long data histories [72]</td>
<td>2</td>
<td>ill-conditioning</td>
<td>response [181]</td>
</tr>
<tr>
<td>SSFD</td>
<td>random</td>
<td>rest [72]</td>
<td>several [72]</td>
<td>10</td>
<td>&lt; data than IDFT still need FRF's</td>
<td>3</td>
<td>ill-conditioning [94]</td>
<td>nonlinear optimization</td>
</tr>
<tr>
<td>SSFD improvement (LMFD or RMFD)</td>
<td>random</td>
<td>rest [72]</td>
<td>none</td>
<td>10</td>
<td>&lt; data than IDFT still need FRF's</td>
<td>3</td>
<td>matrix-fraction (MF) linear problem</td>
<td></td>
</tr>
<tr>
<td>direct</td>
<td>random</td>
<td>rest [72]</td>
<td>pseudo inv.</td>
<td>10</td>
<td>(2)</td>
<td>1</td>
<td>ill-conditioning [72]</td>
<td></td>
</tr>
<tr>
<td>stochastic</td>
<td>white noise</td>
<td>rest [98]</td>
<td>none</td>
<td>6</td>
<td>long data histories</td>
<td>4</td>
<td>none reported</td>
<td></td>
</tr>
<tr>
<td>initial condition</td>
<td>free-decay</td>
<td>arbitrary</td>
<td>several</td>
<td>7</td>
<td>(2)</td>
<td>2</td>
<td>none reported</td>
<td></td>
</tr>
<tr>
<td>OKID forward</td>
<td>random</td>
<td>arbitrary</td>
<td>none</td>
<td>10</td>
<td>significant memory and computations</td>
<td>6</td>
<td>few reported</td>
<td></td>
</tr>
<tr>
<td>OKID backward</td>
<td>random</td>
<td>arbitrary</td>
<td>none</td>
<td>10</td>
<td>significant memory and computations</td>
<td>6</td>
<td>&gt; OKID more inversions</td>
<td></td>
</tr>
<tr>
<td>OCID (RMA)</td>
<td>feedback signal</td>
<td>arbitrary</td>
<td>none</td>
<td>10</td>
<td>significant memory and computations</td>
<td>6</td>
<td>similar to OKID</td>
<td></td>
</tr>
<tr>
<td>random decrement signature</td>
<td>rest</td>
<td>several</td>
<td>long data histories</td>
<td>10</td>
<td>none reported</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculating Markov Parameters</td>
<td>Number of Parameters in addition to:</td>
<td>Software Availability</td>
<td>Documentation Availability</td>
<td>Experience</td>
<td>Remarks</td>
<td>Unweighted Score</td>
<td>Weighted Score</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>------------</td>
<td>---------</td>
<td>-----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>impulse</td>
<td>magnitude</td>
<td>no</td>
<td>[3, 102, 181]</td>
<td>simple structures</td>
<td>-limited energy in practice</td>
<td>41</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>FFT/IDFT</td>
<td>window width</td>
<td>Xmath</td>
<td>[72 Section 4.1, 129]</td>
<td>simulated truss [72]</td>
<td>-IDFT causes aliasing</td>
<td>50</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>SSFD</td>
<td>$p, k$</td>
<td>no</td>
<td>[11, 72 Section 7]</td>
<td>JPL Phase B truss</td>
<td>-no aliasing</td>
<td>43</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>SSFD improvement (LMFD or RMFD)</td>
<td>$p$</td>
<td>no</td>
<td>[71, 72 Section 7]</td>
<td>simulated truss [72]</td>
<td>-no aliasing</td>
<td>51</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>direct</td>
<td>none reported</td>
<td>no</td>
<td>[72 Section 6.1]</td>
<td></td>
<td>Y not unique for $r &gt; 1$</td>
<td>32</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>stochastic</td>
<td>none reported</td>
<td>MATLAB</td>
<td>[91, 98]</td>
<td>Mini-mast [91]</td>
<td>use band-limited random signal</td>
<td>55</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>initial condition</td>
<td>none reported</td>
<td>no</td>
<td>[48]</td>
<td>Daisy [48]</td>
<td>cannot realize $B$ or $D$ matrix</td>
<td>48</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>OKID forward</td>
<td>$p$</td>
<td>MATLAB</td>
<td>[22, 23, 32, 72 Section 6.1, 74, 130]</td>
<td>Hubble [72, 108]</td>
<td>observer gain identification</td>
<td>73</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>OKID backward</td>
<td>$p$</td>
<td>MATLAB</td>
<td>[72 Section 6.7, 155]</td>
<td>10-bay truss</td>
<td>-Markov-like parameters</td>
<td>70</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>OCID (RMA)</td>
<td>$p$</td>
<td>MATLAB</td>
<td>[15, 57, 72 Section 8.0]</td>
<td></td>
<td>-closed-loop systems</td>
<td>68</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>random decrement signature</td>
<td>$i, n$</td>
<td>no</td>
<td>[61]</td>
<td></td>
<td>feedback gain identification</td>
<td>36</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>
(1) LMFD is observable, RMFD is controllable
(2) significant memory and computations
(3) more matrix inversions than with OKID forward
(4) active flexible wing [72], 5 m truss [15]
(5) random, detailed knowledge not required [61]
(6) simulated shuttle NASTRAN model
(7) model is not natural because the current state is predicted by the future state
(8) multiple experiments can be combined

\[ l = \text{number of data points} \]
\[ m = \text{number of outputs} \]
\[ r = \text{number of inputs} \]
\[ \Delta t = \text{sample time (s)} \]
\[ p = \text{observer order} \]
\[ k = \text{number of iterations required to solve the nonlinear optimization problem} \]
\[ i = \text{number of samples taken for the averaging procedure} \]
B.6 Covariance Parameters
### Calculating Covariance Parameters:

<table>
<thead>
<tr>
<th>Control Model SI</th>
<th>Excitation</th>
<th>Initial Conditions</th>
<th>Iterations</th>
<th>Computer Requirements</th>
<th>Model Order</th>
<th>Numerical Problems</th>
<th>Noisy Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>stochastic</td>
<td>white noise</td>
<td>rest [98]</td>
<td>none</td>
<td>long data histories</td>
<td>4</td>
<td>none reported</td>
<td>7</td>
</tr>
<tr>
<td>PRBS</td>
<td>PRBS</td>
<td>rest</td>
<td>none</td>
<td>long data histories</td>
<td>4</td>
<td>none reported</td>
<td>6</td>
</tr>
<tr>
<td>deterministic</td>
<td>impulse</td>
<td>rest</td>
<td>( d \to \infty )</td>
<td>many computations [100]</td>
<td>4</td>
<td>none reported</td>
<td>7</td>
</tr>
<tr>
<td>FFT/IDFT</td>
<td>random</td>
<td>rest</td>
<td>windows</td>
<td>long data histories</td>
<td>2</td>
<td>ill-conditioning [129]</td>
<td>2</td>
</tr>
</tbody>
</table>

### Calculating Covariance Parameters (Software Availability):

<table>
<thead>
<tr>
<th>Number of Parameters</th>
<th>Software Availability</th>
<th>Documentation Availability</th>
<th>Experience</th>
<th>Remarks</th>
<th>Unweighted Score</th>
<th>Weighted Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRBS</td>
<td>( x )</td>
<td>Xmath [181]</td>
<td>3-DOF simulation</td>
<td>-better results than white noise [181]</td>
<td>① 50 ③ 60</td>
<td></td>
</tr>
<tr>
<td>deterministic</td>
<td>( d )</td>
<td>Xmath [91, 99, 100]</td>
<td>Mini-mast [91]</td>
<td>③ 47 ① 64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFT/IDFT</td>
<td>window size</td>
<td>Xmath [129, email with K. Liu]</td>
<td>② 2 ② 2</td>
<td>③ 35 ④ 46</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( x \) = scalar integer indicating the shift register length used to generate the PRBS signal

\( d \) = parameter which approaches infinity when calculating covariance parameters
B.7 Time-Domain SI
<table>
<thead>
<tr>
<th>Time-domain SI:</th>
<th>Control-Model SI</th>
<th>Excitation</th>
<th>Initial Conditions</th>
<th>Iterations</th>
<th>Computer Requirements</th>
<th>Model Order</th>
<th>Numerical Problems</th>
<th>Noisy Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polyreference (time-domain)</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>SVD</td>
<td>&gt; ERA similar to TDPI [89]</td>
<td>not minimal</td>
<td>orthonormal close to 1</td>
<td>similar to ERA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct Parameter (TDPI)</td>
<td>ARMA u(t) [159]</td>
<td>arbitrary</td>
<td>rest</td>
<td>SVD</td>
<td>&gt; ITD, LSCE &amp; ERA [159]</td>
<td>4</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Principal Hankel Component (time-domain)</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>SVD</td>
<td>similar to ERA</td>
<td>minimal</td>
<td>similar to ERA [72]</td>
<td>similar to ERA [72]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERA</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>SVD</td>
<td>significant memory and computations [159]</td>
<td>minimal [94]</td>
<td>few, [77]</td>
<td>SNR &gt; 50 [2], [142]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERA/DC</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>SVD</td>
<td>&gt; ERA τ ≠ 0 &lt; ERA [91]</td>
<td>minimal [91] &lt; ERA [75]</td>
<td>few</td>
<td>better than ERA [72]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recursive ERA</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>Gram-Schmidt (1)</td>
<td>gradually increase</td>
<td>7</td>
<td>5</td>
<td>SNR &gt; 50 [2], [142]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast ERA (HQMC)</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>Eigenvalue Decompart.</td>
<td>&lt; ERA [127] square data matrix</td>
<td>minimal</td>
<td>&lt; ERA eig</td>
<td>not as good as ERA [72]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDVV</td>
<td>state-space</td>
<td>random</td>
<td>rest</td>
<td>SVD</td>
<td>large matrices [91]</td>
<td>high order poor results</td>
<td>&gt; ERA [91] (larger matrices)</td>
<td>not as good as ERA [91]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS Regression (time-domain)</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>pseudo inv.</td>
<td>special case of ERA</td>
<td>not minimal</td>
<td>severe</td>
<td>must exist</td>
</tr>
<tr>
<td>Q-Markov CovER</td>
<td>state-space</td>
<td>impulse</td>
<td>rest</td>
<td>SVD</td>
<td>&gt; ERA since covariance and Markov parameters [91]</td>
<td>minimal</td>
<td>potential [100]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observable Canonical Form (time-domain)</td>
<td>state-space</td>
<td>observer impulse</td>
<td>rest</td>
<td>none</td>
<td>significant memory and computations</td>
<td>not minimal observable (4)</td>
<td>significant overspecification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
tr

01

p y du
mouy 01p q n b a t -

Z8

0
9

L
O1

@
19

lS9 U O ! I W

kzll
[95I Imw-!u!W
A1!1!9"19 I V O " 3 ? l ~ w n s - 11011'(0011 3f99nH
[ z L IS
~ J a l ~ ~ f l m d - ~ PSO w
01
JOJ p p u a w o m IOUL

19

Z9

8

;B

08

8

;s

0

~ L ZII v

~ mil1
a

Iwil vva

@

bs

O

38!OU SS31 SU!€l1U03

6

p1101?~103
IOU

!L

OL

L9

E9

(6s11 sasati 1~31
tgpOU ItflUlüiîi30PM1j~ l o apidui

(161
O

u'p'b

OU

AAaW

9
ou

~VILVM

8
(SEI 'KI

[LZI 1 law-!u!W
( Z L 3NqnH
~
ILL1W!lW

'821 'LtI 'SZ~'901 '501
'81 'U 'E'S u"!Ims ZL
'62 '52 '2 i 'O1 '81
O1

1

5

! ~ S
9

8W.LVH
tliuuX

ma

n aa y s t n q

u'a'p'b

u 'p ' b

9

O

19's u o ! W 21 '6EI
8

ab

8
3~xP
3

[EL'b'S Uo!lmS I L '611

p

u 'b

OU

9

(EL11lh-SU

ul

t

O

01

s

u ' p *li

ou

l"d!=u!'d

Z

O

(ldai)

16811s
mioura~8dfapou uo s n q -

L9

uap'b

9

Ilall

{ Z L 99WH
~

3q uu3 uo!p~uuoj-

59

OU

O

01

8

W!OU J!-

vm maiiriPWP

u 'b

9

(1's uoli=s ZLI

s
( s w u ti)
Ib0l)W V w ! s

O1

0

BbrlLVW
O

rym uopqauoa ua~n

C8

16

1181 1
' 11 '911 '691
LZ 1'86 '96 'P8 '8s '91

(~WOH)

1

9s

t

182;1 '9211

t LZ 11 lrn"-!U!Wl

~ a i msawy
j
01 01 ç-

sayawddu Xnrrnm.

'6

01

u' d

OU

01

8
1161imu-!u!W

0

ZLI

01

(681UmmF

s
uo!im!ltlai
uiio~-pa!uous3-

1 6 ~ '68'1~1
1
01

t

ou

~ ~ ~ m lm
.m
18d

9

O

[891 '19 1 '06
'S'S uows ZL ' ~ E I

tA

ou

u ' p'b

(u!uuiop-mqi)
muaa~dlod


<table>
<thead>
<tr>
<th>Time-domain SI:</th>
<th>Control-Model SI</th>
<th>Excitation</th>
<th>Initial Conditions</th>
<th>Iterations</th>
<th>Computer Requirements</th>
<th>Model Order</th>
<th>Numerical Problems</th>
<th>Noisy Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSRA</td>
<td>state-space</td>
<td>random</td>
<td>rest</td>
<td>0</td>
<td>SVD</td>
<td>significant memory</td>
<td>high order</td>
<td>few</td>
</tr>
<tr>
<td>Unbiased ORSE</td>
<td>state-space</td>
<td>random</td>
<td>rest [92]</td>
<td>0</td>
<td>SVD iterative</td>
<td>&gt; ORSE [96] large amount</td>
<td>similar to ORSE</td>
<td>similar to ORSE</td>
</tr>
<tr>
<td>MOESP (CVA, N4SID [136])</td>
<td>state-space</td>
<td>random</td>
<td>rest [96]</td>
<td>0</td>
<td>SVD RQ</td>
<td>&gt; ORSE [96] large amount of data</td>
<td>similar to ORSE</td>
<td>similar to ORSE</td>
</tr>
<tr>
<td>Subspace</td>
<td>state-space</td>
<td>random</td>
<td>(7)</td>
<td>0</td>
<td>SVD</td>
<td>reduce with lattice</td>
<td>9</td>
<td>none reported</td>
</tr>
<tr>
<td>ITD</td>
<td>modal SI [159]</td>
<td>impulse free-decay</td>
<td>arbitrary [146]</td>
<td>none</td>
<td>(2)</td>
<td>(3)</td>
<td>similar to TDPI [89]</td>
<td>7</td>
</tr>
<tr>
<td>Polya reference LSCE</td>
<td>modal SI [159]</td>
<td>impulse free-decay</td>
<td>none</td>
<td>similar to TDPI [89]</td>
<td>?</td>
<td>similar to TDPI [89]</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>LSMBT</td>
<td>modal SI</td>
<td>impulse free-decay</td>
<td>none</td>
<td>several experiments [4]</td>
<td>?</td>
<td>none reported</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Forward Time Series</td>
<td>modal SI</td>
<td>random</td>
<td>none</td>
<td>significant computations</td>
<td>5</td>
<td>high order</td>
<td>few</td>
<td>highly sensitive</td>
</tr>
<tr>
<td>Backward Time Series</td>
<td>modal SI</td>
<td>random</td>
<td>none</td>
<td>significant computations</td>
<td>5</td>
<td>high order</td>
<td>few</td>
<td>highly sensitive</td>
</tr>
<tr>
<td>Constrained ERA (Temporal Correlation Method)</td>
<td>modal SI</td>
<td>impulse</td>
<td>rest</td>
<td>SVD</td>
<td>&lt; ERA order N (versus 2N)</td>
<td>order N</td>
<td>order N</td>
<td>similar to ERA</td>
</tr>
<tr>
<td>Time-domain SI:</td>
<td>Number of Parameters</td>
<td>Software Availability</td>
<td>Documentation Availability</td>
<td>Experience</td>
<td>Remarks</td>
<td>Unweighted Score</td>
<td>Weighted Score</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>------------</td>
<td>---------</td>
<td>-----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>SSRA</td>
<td>q, d, n</td>
<td>no</td>
<td>12-m truss [50]</td>
<td>7</td>
<td>-use ERA to eliminate excess modes</td>
<td>74</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>ORSE</td>
<td>q, n</td>
<td>no</td>
<td>MACE [95, 96]</td>
<td>10</td>
<td>-potential bias due to covariance term [96]</td>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Unbiased ORSE</td>
<td>q, n, k</td>
<td>no</td>
<td>MACE [96]</td>
<td>10</td>
<td>-uses instrumental variable to remove bias</td>
<td>77</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>MOESP (CVA, N4SID [136])</td>
<td>q, n</td>
<td>MATLAB MIMO MIMO MIMO MIMO MIMO</td>
<td>not applied to structures [96]</td>
<td>10</td>
<td>-formulation for A and C equivalent to ORSE -can use instrumental variable</td>
<td>66</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>Subspace</td>
<td>n</td>
<td>MATLAB MIMO MIMO MIMO</td>
<td>MIMO MIMO MIMO MIMO MIMO</td>
<td>10</td>
<td>-bias or unbiased algorithm</td>
<td>(6)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ITD</td>
<td>q, d, n</td>
<td>no</td>
<td>[3, 4, 19, 146, 159]</td>
<td>8</td>
<td>-focus on modal parameter SI [3]</td>
<td>86</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>Polyreference LSCE</td>
<td>q, d, n</td>
<td>no</td>
<td>[89, 159]</td>
<td>9</td>
<td>-focus on modal-parameter SI [4]</td>
<td>65</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>LSMBT</td>
<td>n</td>
<td>no</td>
<td>5 DOF simulation</td>
<td>7</td>
<td>-focus on modal-parameter SI [4]</td>
<td>58</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>Forward Time Series</td>
<td>n</td>
<td>no</td>
<td>[50]</td>
<td>8</td>
<td>-modal representation polynomials A and B</td>
<td>61</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>Backward Time Series</td>
<td>n</td>
<td>no</td>
<td>[50, 52]</td>
<td>7</td>
<td>-system modes distinguished from noise modes</td>
<td>61</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>Constrained ERA (Temporal Correlation Method)</td>
<td>q, d, n</td>
<td>no</td>
<td>[120]</td>
<td>3</td>
<td># sensors &gt; # modes - assumes normal mode damping calculations separate</td>
<td>56</td>
<td>53</td>
<td></td>
</tr>
</tbody>
</table>
significant computations, low memory requirements [104]
(2) several experiments and exciter positions required [3], similar to TDP [89]
(3) insensitive if the model order is greater than the number of modes of interest
(4) canonical-form [72], this form has numerical problems with the eigenvalue decomposition
(5) assumes Gaussian distributed, zero-mean white noise
(6) two MATLAB functions: sds (subspace deterministic), ssst (subspace stochastic)
(7) initial state is estimated [37]
(8) focus on poles, modal participation factors, no mode shapes [159]

l = number of data points
m = number of outputs
r = number of inputs
Δt = sample time (s)
n = realized model order
N = number of degrees of freedom
q = number of block rows in a Hankel (or data) matrix
    also the number of Markov and covariance parameters used by Q-Markov CovER
d = number of block columns in a Hankel (or data) matrix
τ = delay operator
p = observer order
k = number of iterations specified
Appendix C

Computations
OKID with ERA Calculations:

Number of inputs \( r = 3 \)
Number of outputs \( m = 23 \)
Number of data points per channel \( l = 2400 \)
Number of block rows in Hankel matrix \( p = q = 8 \)
\( X \) times number of columns as rows in \( H_i \) \( X = 3 \)

Initialize Parameters:

scale input data \( u \) \( r \times l \)
scale output data \( y \) \( m \times l \)
matrix multiply \( r l + m l \) \( 6.2 \times 10^4 \) flop

Step 1  Compute Observer Markov Parameters

observer input/output description \( y = \hat{y} V \)

matrix multiply \( y V^T \) \( m \times [(r+m)p+r] \)

matrix multiply \( V V^T \) \( [(r+m)p+r] \times [(r+m)p+r] \)

matrix inverse \( [(r+m)p+r]^{3} \) \( 9.4 \times 10^6 \) flop

Step 2  Recover System and Observer Gain M.P.

(M.P. = Markov Parameters)
(H = Hankel Matrix)

number of block columns in \( H_i \) \( d = \text{ceiling}(Xm p / r) \)

system M.P. \( Y_0, Y_1, Y_2, ..., Y_{d=p} \)
number System M.P. \( Y_{d=p} = Y_{nmp} \)
observer Gain M.P. \( Y^0_0, Y^0_1, Y^0_2, ..., Y^0_{d=p} \)
form \( P_k = [Y_k, Y^0_k] \)
recursive summation required \( \text{numsum} = 28 (1 + 2 + 3 + ... + p-1) \)
matrix multiply \( \text{numsum} \times m [m (r+m)] + (nmp-p) m [m (r+m)] + p m m r \) \( 2.1 \times 10^7 \) flop
TOTAL FOR OKID \( 1.5 \times 10^8 \) flop
APPENDIX C. COMPUTATIONS

Step 3  System Realization with ERA

\( (n = \text{order of realized system}) \)
separate \( P_k \) into \( Y_k \) and \( Y'_k \)
form \( H_0 \) using \( Y_k \)
use all singular values \( n = m p \)
singular value decomposition of \( H_0 \)
form \( H_1 \) using \( Y_k \)

\[ \begin{align*}
\text{matrix multiply (A)} & \quad n \times n + n (d \times n) + d \times n^2 \quad 5.0E+07 \text{ flop} \\
(\text{matrix multiply (B)}) & \quad n \times (n \times n) \quad 1.0E+05 \text{ flop} \\
(\text{matrix multiply (C)}) & \quad m \times (n \times n) \quad 7.8E+05 \text{ flop} \\
\text{form D from } Y & \quad n \times n \times n \quad 6.2E+06 \text{ flop} \\
\end{align*} \]

Step 4  Realize Observer Gain Matrix and Rescale

\[ \begin{align*}
\text{construct observability matrix } P & \quad n \times n \quad 5.5E+06 \text{ flop} \\
(\text{matrix multiply}) & \quad (p-1) \times n \times n \\
(\text{observer gain matrix } G = [P]^{-1} Y^o) & \quad n \times m \\
(\text{matrix inverse}) & \quad n \times n \\
(\text{matrix multiply}) & \quad n \times m \\
(\text{scale matrices } B, C, \text{ and } G) & \quad n \times r + m \times n \times n \quad 9.0E+03 \text{ flop} \\
\text{TOTAL FOR ERA} & \quad 1.2E+08 \text{ flop} \\
\text{TOTAL} & \quad 2.7E+08 \text{ flop} \\
\end{align*} \]
OKID with Q-Markov CovER Calculations:  

Number of inputs \( r = 3 \)
Number of outputs \( m = 23 \)
Number of data points per channel \( l = 2400 \)
Observer order \( p = q - l = 8 \)
Deterministic formula for covariance \( d = 2000 \)

Initialize Parameters:

scale input data \( u \)
matrix multiply \( r \times l \)
scale output data \( y \)
matrix multiply \( m \times l \)
matrix multiply \( r \times m \times l \)

Step 1 Compute Observer Markov Parameters

observer input/output description \( y = \bar{Y} \cdot V \)
matrix multiply \( \bar{Y} \)
matrix multiply \( V \)
solve for \( \bar{Y} = y \bar{V}^T \cdot [V V^T]^{-1} \)
matrix multiply \( y \bar{V}^T \)
matrix multiply \( m \times [(r+m) \times p + r] \)
matrix multiply \( [m]^{-1} \)
matrix multiply \( m \times [(r+m) \times p + r] \)
matrix multiply \( m \times [(r+m) \times p + r] \)
matrix multiply \( m \times [(r+m) \times p + r] \)

Step 2 Recover System and Observer Gain M.P.

(M.P. = Markov Parameters) 
(\( H_i = \) Hankel Matrix)

number of block columns in \( H_i \) \( d = 2000 \)

system M.P. \( Y_0, Y_1, Y_2, \ldots, Y_{\text{dep}} \)
number System M.P. \( Y_{\text{dep}} = Y_{\text{amp}} \)
observer Gain M.P. \( Y^a_1, Y^a_2, \ldots, Y^a_{\text{dep}} \)
form \( P_k = [Y_k \ Y^a_k] \)
recursive summation required \( \text{numsum} = 28 \times (1 + 2 + 3 + \ldots + p-1) \)
matrix multiply \( \text{amp} d + p \)

TOTAL FOR OKID \( 3.5 \times 10^8 \) flop
**APPENDIX C. COMPUTATIONS**

200

**Step 3 System Realization with Q-Markov CovER**

\( n = \text{order of realized system} \)

- separate \( P_k \) into \( Y_k \) and \( Y^*_k \)
- calculate \( R_k \)
- form \( H_k \) using \( Y_k \)
- form \( H_k \) using \( R_k \)
- form \( D_k = R_k H_k \)
- matrix multiply
- form upper left partition \( D_k^{UL} \)
- singular value decomposition of \( D_k^{UL} \)
- use all singular values \( n = m p \)
- form pseudo-inverse of \( P_k^{(1)} \)
- matrix inverse
- matrix multiply
- form lower left partition \( D_k^{LL} \)
- form \( P_k^{(2)} \)
- matrix multiply
- matrix multiply \( (A) \)
- matrix multiply \( (B) \)
- matrix multiply \( (C) \)
- form \( D \) from \( Y \)

\[ (p+1)(d+1) m (rm) \quad 2.9E+07 \text{ flop} \]

\[ m (p+1) x r(p+1) \]

\[ m (p+1) \times m(p+1) \]

\[ m (p+1) r(p+1) m (p+1) \]

\[ m p x m p \]

\[ 4/3 (m p)^3 \]

\[ 8.3E+06 \text{ flop} \]

\[ n \]

\[ m p \]

\[ n = m p \]

\[ n x n \]

\[ n^j \]

\[ 6.2E+06 \text{ flop} \]

\[ n^j \]

\[ 6.2E+06 \text{ flop} \]

\[ n x n \]

\[ n^j \]

\[ 6.2E+06 \text{ flop} \]

\[ n^j \]

\[ 6.2E+06 \text{ flop} \]

\[ n (n r) \]

\[ 1.0E+05 \text{ flop} \]

\[ m(n n) \]

\[ 7.8E+05 \text{ flop} \]

\[ 184 \]

**Step 4 Realize Observer Gain Matrix and Rescale**

- construct observability matrix \( P \)
- matrix multiply
- observer gain matrix \( G = [P]^{-1} Y^* \)
- matrix inverse
- matrix multiply
- scale matrices \( B, C, \) and \( G \)

\[ n x n \]

\[ (p-1) m n n \]

\[ n x m \]

\[ n x m \]

\[ n m \]

\[ n r + m n + n m \]

\[ 9.0E+03 \text{ flop} \]

\[ \text{TOTAL FOR Q-Markov CovER} \]

\[ 7.6E+07 \text{ flop} \]

\[ \text{TOTAL} \]

\[ 4.3E+08 \text{ flop} \]
ORSE Calculations:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of inputs</td>
<td>( r = 3 )</td>
</tr>
<tr>
<td>Number of outputs</td>
<td>( m = 23 )</td>
</tr>
<tr>
<td>Number of data points per channel</td>
<td>( l = 2400 )</td>
</tr>
<tr>
<td>Number of block rows in Y matrix</td>
<td>( q = 25 )</td>
</tr>
<tr>
<td>( X ) times number of columns as rows in Y</td>
<td>( X = 4 )</td>
</tr>
<tr>
<td>Desired model order</td>
<td>( n = 100 )</td>
</tr>
</tbody>
</table>

Number of columns in Y matrix \( d = 2300 \)

Initialize Parameters:

- scale input data: \( u \) \( r \times l \)
- scale output data: \( y \) \( m \times l \)
- matrix multiply: \( r l + m l \) \( 6.2 \times 10^4 \) flop

Step 1  Form Matrix Products

<table>
<thead>
<tr>
<th></th>
<th>( YY^T )</th>
<th>( YU^T )</th>
<th>( UU^T )</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix multiply</td>
<td>( m \times q \times m \times q )</td>
<td>( (m \times q) \times d \times (m \times q) )</td>
<td>( (r \times q) \times d \times (r \times q) )</td>
</tr>
<tr>
<td>flops</td>
<td>( 7.6 \times 10^8 )</td>
<td>( 9.9 \times 10^7 )</td>
<td>( 1.3 \times 10^7 )</td>
</tr>
</tbody>
</table>

Step 2  Apply SVD to \( UU^T \)

<table>
<thead>
<tr>
<th></th>
<th>( \frac{4}{3} (r \times q)^3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>singular value decomposition of ( UU^T )</td>
<td>( 5.6 \times 10^5 ) flop</td>
</tr>
<tr>
<td>use all singular values ( (q \times r) )</td>
<td>( D_u ) ( r \times q \times r \times q )</td>
</tr>
<tr>
<td>form ( D_u ) matrix</td>
<td>( r \times q \times (r \times q)^2 )</td>
</tr>
<tr>
<td>matrix multiply</td>
<td>( 8.4 \times 10^5 ) flop</td>
</tr>
<tr>
<td>invert ( D_u )</td>
<td>( (r \times q)^4 )</td>
</tr>
<tr>
<td>( D_u^{-1} )</td>
<td>( 4.2 \times 10^5 ) flop</td>
</tr>
</tbody>
</table>

Step 3  Apply SVD to Data Matrix

<table>
<thead>
<tr>
<th></th>
<th>( m \times q \times (r \times q)^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>form data matrix</td>
<td>( m \times q \times (r \times q)^2 )</td>
</tr>
<tr>
<td>singular value decomposition</td>
<td>( \frac{4}{3} (m \times q)^{\frac{4}{3}} )</td>
</tr>
<tr>
<td>singular values to keep ( n )</td>
<td>( n = 100 )</td>
</tr>
<tr>
<td>flops</td>
<td>( 6.5 \times 10^6 ) flop</td>
</tr>
<tr>
<td>flops</td>
<td>( 2.5 \times 10^6 ) flop</td>
</tr>
</tbody>
</table>

Step 4  Calculate A and C

<table>
<thead>
<tr>
<th></th>
<th>( P_q^{(1)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>calculate pseudoinverse of ( P_q^{(1)} )</td>
<td>( P_q^{(1)} ) ( m(q-1) \times n )</td>
</tr>
<tr>
<td>matrix pseudo inverse</td>
<td>( m(q-1) \times n )</td>
</tr>
<tr>
<td>matrix multiply ( (A) )</td>
<td>( n \times m(q-1) \times n )</td>
</tr>
<tr>
<td>form C from ( P_q )</td>
<td>( 5.5 \times 10^6 ) flop</td>
</tr>
</tbody>
</table>
Step 5 Calculate B and D

\[ y_b \quad m \times [n,r] \]

matrix multiply

\[ n \times (n^2 + n + m) \quad 2.2E+09 \text{ flop} \]

\[ y_d \quad m \times [m,r] \]

matrix multiply

\[ r \times 4 m \quad 4.1E+04 \text{ flop} \]

form matrix \( \Phi (k) \)

\[ \Phi(k) \quad m \times [m, r + n] \]

use vector \( y(k) \)

\[ y(k) \quad m \times 1 \]

calculate pseudo-inverse of \( \Phi (k) \)

\[ \frac{1}{4} (mr + nr) m (mr + nr) \quad 1.9E+09 \text{ flop} \]

matrix multiply

\[ \frac{1}{4} (mr + nr) m \quad 5.1E+06 \text{ flop} \]

matrix inverse

\[ (mr + nr)^3 \quad 5.0E+07 \text{ flop} \]

matrix multiply

\[ (mr + nr)^2 \quad 1.4E+05 \text{ flop} \]

calculate B and D

Step 6 Rescale Matrices

scale matrices B and C

\[ n \times (r + m) \quad 2.6E+03 \text{ flop} \]

TOTAL

\[ 5.3E+09 \text{ flop} \]
APPENDIX C. COMPUTATIONS

Subspace Calculations:

Number of inputs \( r = 3 \)
Number of outputs \( m = 23 \)
Number of data points per channel \( l = 2400 \)
Number of block rows in Y matrix \( q = 25 \)
\( X \) times number of columns as rows in Y \( X = 4 \)
Desired model order \( n = 100 \)
Number of columns in Y matrix \( d = 2300 \)

Step 1 Calculate Oblique Projection

- Form input block Hankel matrix \( U_{\text{in}} \)
- Form output block Hankel matrix \( V_{\text{out}} \)
- Matrix multiply (scale data) \( U_{\text{in}} \times V_{\text{out}} \)
- QR decomposition of \( [U_{\text{in}}; V_{\text{out}}] \)
- Calculate \( O_k = V_k / U_k \)
- \( V_k / U_k \)
- \( Y_k / U_k \left[ V_k / U_k \right]^T \)
- \( Y_k / U_k \left[ V_k / U_k \right]^T \)

Step 2 Determine A and C

- Singular value decomposition of \( O_k \)
- Singular values to keep \( n \)
- Calculate pseudo-inverse of \( P_k \)
- Matrix pseudo-inverse
- Matrix multiply (A)
- Form C from \( P_k \)

Step 3 Determine B and D

- Form \( y_k \)
- Matrix multiply
- Form \( y_0 \)
- Matrix multiply
- Form matrix \( \Phi(k) \)
- Use vector \( y(k) \)
- Calculate pseudo-inverse of \( \Phi(k) \)
- Matrix multiply
- Matrix multiply
- Matrix inverse
- Matrix multiply
- Calculate B and D

Total: 8.5E+09 flop
OKID with ERA c Calculations: [A,B,C,D,G]

Number of inputs \( r = 3 \)
Number of outputs \( m = 23 \)
Number of data points per channel \( i = 2400 \)
Number of block rows in Hankel matrix \( p = q = 8 \)
\( X \) times number of columns as rows in \( H_i \) \( X = 3 \)

Initialize Parameters:

scale input data \( u \)

scale output data \( y \)

matrix multiply \( r I + m I \) \( 6.2E+04 \) flop

Step 1 Compute Observer Markov Parameters

Observer input/output description \( y = \bar{Y} Y \)

\( \bar{Y} \) \( m \times l \)
\( Y \) \( m \times [(r+m) p + r] \)

Solve for \( \bar{Y} = y V T \) \( [VV^T]^{-1} \)

Matrix multiply \( y V T \) \( m \times [(r+m) p + r] \)

Matrix multiply \( V V^T \) \( m \times [(r+m) p + r] \)

Matrix inverse \( m \times [(r+m) p + r] \)

Step 2 Recover System and Observer Gain M.P.

(M.P. = Markov Parameters)

\( (H_i = \) Hankel Matrix)

Number of block columns in \( H_i \)

\( d = \) ceiling(\( X m p / r \)) \( 184 \)

System M.P. \( Y_0, Y_1, Y_2, \ldots, Y_{\text{amp}} \)

\( Y_k \) \( m \times r \)

Number System M.P. \( Y_{\text{amp}} = Y_{\text{amp}} \)

\( \text{amp} = d + p \) \( 192 \)

Observer Gain M.P. \( Y^o_0, Y^o_1, Y^o_2, \ldots, Y^o_{\text{amp}} \)

\( Y^o_k \) \( m \times m \)

Form \( P_k = [Y_k \ Y^o_k] \)

Recursive summation required

\( \text{numsum} = \) \( 28 (1 + 2 + 3 + \ldots + p-1) \)

Matrix multiply

TOTAL FOR OKID \( 1.5E+08 \) flop
APPENDIX C. COMPUTATIONS

Step 3  System Realization with ERA c

\( n = \text{order of realized system} \)
separate \( P_k \) into \( Y_k \) and \( Y_k^* \)
form \( H_0 \) using \( Y_k \)
singular value decomposition of \( H_0 \)
use all singular values \( n = m_p \)

\[
\begin{align*}
H_0 & \quad m_p \times d_r \\
n & \quad m_p \\
= & \quad n \\
\text{form } H_1 \text{ using } Y_k \\
\text{matrix multiply (A)} & \quad n^3 + n^3 + n (d_r n) + d_r n^2 \\
\text{matrix multiply (B)} & \quad n (n r) \\
\text{matrix multiply (C)} & \quad m (n n) \\
\text{form } D \text{ from } Y \\
\end{align*}
\]

Step 4  Realize Observer Gain Matrix and Rescale

construct observability matrix \( P \)
matrix multiply
observer gain matrix \( G = [P]^{-1} Y^o \)

\[
\begin{align*}
P & \quad n \times n \\
(\text{matrix multiply}) & \quad (p-1) m n n \\
Y^o & \quad n \times m \\
G & \quad n \times m \\
\text{matrix inverse} & \quad n \times n \\
\text{matrix multiply} & \quad n n m \\
\text{scale matrices } B, C, \text{ and } G & \quad n r + m n + n m \\
\end{align*}
\]

\[
\begin{align*}
\text{TOTAL FOR ERA c} & \quad 1.2E+08 \text{ flop} \\
\text{TOTAL} & \quad 2.7E+08 \text{ flop}
\end{align*}
\]
OKID with ERA C Calculations:

<table>
<thead>
<tr>
<th>Calculation</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of inputs</td>
<td>( r )</td>
<td>3</td>
</tr>
<tr>
<td>Number of outputs</td>
<td>( m )</td>
<td>23</td>
</tr>
<tr>
<td>Number of data points per channel</td>
<td>( i )</td>
<td>2400</td>
</tr>
<tr>
<td>Number of block rows in Hankel matrix</td>
<td>( p = q )</td>
<td>8</td>
</tr>
<tr>
<td>( X ) times number of columns as rows in ( H )</td>
<td>( X )</td>
<td>3</td>
</tr>
</tbody>
</table>

Initialize Parameters:

- Scale input data: \( u \), \( r \times l \)
- Scale output data: \( y \), \( m \times l \)
- Matrix multiply: \( r \times l + m \times l \) - 6.2E+04 flop

**Step 1** Compute Observer Markov Parameters

Observer input/output description: \( y = \bar{Y} V \)

- Solve for \( \bar{Y} = y V^T (V V^T)^{-1} \)
  - Matrix multiply: \( V V^T \) - 1.2E+07 flop
  - Matrix inverse: \( (V V^T)^{-1} \) - 1.1E+08 flop

**Step 2** Recover System and Observer Gain M.P.

- (M.P. = Markov Parameters)
- (\( \Omega \) = Hankel Matrix)

- Number of block columns in \( H \): \( d = \text{ceiling}(X m p / (r+m)) \) - 22

- System M.P.: \( Y_{0a}, Y_1, Y_2, \ldots, Y_{d+p} \)
  - \( Y_k \), \( m \times r \)
  - \( Y_{np} \), \( d+p \)
  - \( Y_{np} = \text{amp} \)

- Observer Gain M.P.: \( Y^a_{1}, Y^a_{2}, \ldots, Y^a_{d+p} \)
  - \( P_k \), \( m \times (r+m) \)
  - \( \text{numsum} \)

- Recursive summation required
- Matrix multiply: \( \text{numsum} m [m (r+m)] + (\text{amp}+p) p m [m (r+m)] + p m m r \) - 2.8E+06 flop

TOTAL FOR OKID: 1.3E+08 flop
Step 3  System Realization with ERA C

\( n = \text{order of realized system} \)

- form \( H_0 \) using \( Y_k \)
- singular value decomposition of \( H_0 \)
- use all singular values \( n = m \times p \)

\[
\begin{align*}
H_0 &= m \times p \times d \times (r+m) \\
n &= m \times p \\
\text{TOTAL FOR ERA C} &= 5.8 \times 10^7 \text{ flop}
\end{align*}
\]

Step 4  Rescale Matrices

- scale matrices \( B, C, \) and \( G \)

\[
\begin{align*}
\text{TOTAL} &= n \times r + m \times n + n \times m \\
\text{TOTAL FOR ERA C} &= 9.0 \times 10^3 \text{ flop}
\end{align*}
\]
OKID with ERA/DC c Calculations: [A,B,C,D,G]

(Special Case)

Number of inputs \( r = 3 \)
Number of outputs \( m = 23 \)
Number of data points per channel \( l = 2400 \)
Number of block rows in Hankel matrix \( p=q=8 \)
\( X \times \) number of columns as rows in \( H_j \) \( X=3 \)

Initialize Parameters:

scale input data \( u \)
\( r \times l \)
scale output data \( y \)
\( m \times l \)

matrix multiply \( r l + m l \) \( 6.2E+04 \) flop

Step 1 Compute Observer Markov Parameters

observer input/output description \( y = \bar{Y}V \)
\( y \)
\( m \times l \)
\( \bar{Y} \)
\( m \times [(r+m) p + r] \)
\( V \)
\( [(r+m) p + r] \times l \)

solve for \( \bar{Y} = yV^T (VV^T)^{-1} \)
\( yV^T \)
\( m \times [(r+m) p + r] \)
\( VV^T \)
\( [(r+m) p + r] \times [(r+m) p + r] \)

matrix multiply \( m \times [(r+m) p + r] \)
\( m \times [(r+m) p + r] \)

matrix inverse \( [(r+m) p + r] \times [(r+m) p + r] \)

Step 2 Recover System and Observer Gain M.P.

\( (M.P. = \) Markov Parameters)
\( (H_j = \) Hankel Matrix)

number of block columns in \( H_j \)
\( d = \) ceiling(\( X m p / r \))
\( d = 184 \)

system M.P. \( Y_0, Y_1, Y_2, \ldots, Y_{d+p} \)
\( Y_k \)
\( m \times r \)
\( P_m = \) number System M.P. \( Y_{d+p} = Y_{\text{amp}} \)
\( \text{amp} \)
\( d+p \)
\( 192 \)

observer Gain M.P. \( Y^{a}_0, Y^{a}_1, Y^{a}_2, \ldots, Y^{a}_{d+p} \)
\( Y^{a}_k \)
\( m \times m \)
form \( P_{k} = [Y_k, Y^{a}_k] \)
\( P_k \)
\( m \times (r+m) \)

recursive summation required \( \text{numsum} = 28 \times (1 + 2 + 3 + \ldots + p-1) \times m \times (r+m) \)
\( \text{numsum} m [m (r+m)] + \)
\( (\text{amp}-p) [m (r+m)] + \)
\( p m r \)

TOTAL FOR OKID \( 1.5E+08 \) flop
Step 3 System Realization with ERA/DC c

\( n = \text{order of realized system} \)

- separate \( P_k \) into \( Y_k \) and \( Y_k^\alpha \)
- form \( H_0 \) using \( Y_k \)
- form \( R_Q^{\text{ab}} = H_0 H_0^T \)
- form \( R_1^{\text{ab}} \)
- matrix multiply
- singular value decomposition of \( R_Q^{\text{ab}} \)
- use all singular values \( n = mp \)
- matrix multiply (A)
- matrix multiply (B)
- matrix multiply (C)
- form \( D \) from \( Y \)

Step 4 Realize Observer Gain Matrix and Rescale

- construct observability matrix \( P \)
- matrix multiply
- observer gain matrix \( G = [P]\^{-1} Y^o \)
- matrix inverse
- matrix multiply
- scale matrices \( B, C, \) and \( G \)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>Time (flop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL FOR ERA/DC c</td>
<td>( 7.0E+07 ) flop</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>( 2.2E+08 ) flop</td>
<td></td>
</tr>
</tbody>
</table>
OKID with ERA/DC C Calculations:
(Special Case)

Number of inputs \( r = 3 \)
Number of outputs \( m = 23 \)
Number of data points per channel \( l = 2400 \)
Number of block rows in Hankel matrix \( p = q = 8 \)
\( X \) times number of columns as rows in \( H_j \) \( X = 3 \)

Initialize Parameters:

scale input data \( u \) \( r \times 1 \)
scale output data \( y \) \( m \times 1 \)
matrix multiply \( r l + m l \) \( 6.2E+04 \) flop

Step 1 Compute Observer Markov Parameters

observer input/output description \( y = \tilde{Y}v \)
matrix multiply \( y \) \( m \times 1 \)
\( \tilde{Y} \) \( m \times [(r+m)p + r] \)
\( \tilde{V} \) \( [(r+m)p + r] \times 1 \)
solve for \( \tilde{Y} = YVT[VVT]^{-1} \)
matrix multiply \( yVT \) \( m \times [(r+m)p + r] \)
\( VVT \) \( [(r+m)p + r] \times [(r+m)p + r] \)
\( [VVT]^{-1} \) \( [(r+m)p + r] \times [(r+m)p + r] \)
matrix inverse \( [VVT]^{-1} \) \( [(r+m)p + r]^{-1} \)
matrix multiply \( m \times ((r+m)p + r)^{-1} \) \( 9.4E+06 \) flop

Step 2 Recover System and Observer Gain M.P.

(M.P. = Markov Parameters)
\( (H_j = \text{Hankel Matrix}) \)

number of block columns in \( H_j \) \( d = \text{ceiling}(X m p / (r+m)) \)
\( d = 22 \)

system M.P. \( Y_{90}, Y_1, Y_2, \ldots, Y_{d+p} \)
number System M.P. \( Y_{d+p} = Y_{\text{amp}} \)
amp \( d+p \)
amp = 30
observer Gain M.P. \( Y_{p0}, Y_{p1}, \ldots, Y_{p_{d+p}} \)
form \( P_k = [Y_k, Y_{k+1}] \)
recursive summation required \( \text{numsum} = 28 (1 + 2 + 3 + \ldots + p-1) \)
matrix multiply \( \text{numsum} m (m (r+m)) + (\text{amp-p}) p m (m (r+m)) + 2.8E+06 \) flop
\( p m m r \)
TOTAL FOR OKID \( 1.3E+08 \) flop
APPENDIX C. COMPUTATIONS

Step 3  System Realization with ERA/DC C

\[(n = \text{order of realized system})\]

form \(H_0\) using \(Y_k\)

\[H_0 \quad m \times d \times (r+m)\]

form \(R_{hh} = H_0 \ H_0^\top\)

\[R_{hh} \quad m \times m \times p\]

form \(R_{hh}\)

matrix multiplication

\[m \times (p+1) \times (r+m) \times m \times (p+1)\]

2.5E+07 flop

singular value decomposition of \(R_{hh}\)

\[4/3 \times m \times p \times (m \times p)^2\]

8.3E+06 flop

use all singular values \(n = m \times p\)

\[n \quad m \times p\]

184

matrix multiply (A)

\[n \times n \times n + n \times n \times n \times n\]

2.5E+07 flop

matrix multiply ([B G])

\[n \times [n \times (r+m)]\]

8.8E+05 flop

separate \([B G]\)

matrix multiply (C)

\[m \times (n \times n)\]

7.6E+05 flop

form D from \(Y\)

Step 4  Rescale Matrices

scale matrices B, C, and G

\[n \times r \times m \times n \times n \times m\]

9.0E+03 flop

TOTAL FOR ERA/DC C

5.9E+07 flop

TOTAL

1.9E+08 flop
**OKID with Fast ERA Calculations:**

Number of inputs \( r = 3 \)

Number of outputs \( m = 23 \)

Number of data points per channel \( l = 2400 \)

Number of block rows in Hankel matrix \( p = q - 1 \)

\( X \) times number of columns as rows in \( H_j \) \( X = 3 \)

Initialize Parameters:

- scale input data \( u \) \( r \times l \)
- scale output data \( y \) \( m \times l \)
- matrix multiply \( r \times l + m \times l \) \( 6.2E+04 \) flop

**Step 1  Compute Observer Markov Parameters**

observer input/output description \( y = \tilde{Y} \tilde{V} \)

\( \tilde{Y} \quad m \times l \)

\( \tilde{V} \quad (l + m) \times l \)

solve for \( \tilde{Y} = y \tilde{V}^T [VV^T]^{-1} \)

\( y \tilde{V}^T \quad m \times [(l + m) \times r + l] \)

matrix multiply \( m \times [(l + m) \times r + l] \) \( 1.2E+07 \) flop

matrix multiply \( VV^T \quad [(l + m) \times r] \times [(l + m) \times r + l] \)

matrix inverse \( [(l + m) \times r] \times [(l + m) \times r + l] \) \( 9.4E+06 \) flop

matrix multiply \( m \times [(l + m) \times r] \) \( 1.0E+06 \) flop

**Step 2  Recover System and Observer Gain M.P.**

(M.P. = Markov Parameters)

(\( H_j \) = Hankel Matrix)

- number of block columns in \( H_j \) \( d = \) ceiling \((X m + p / r)\) \( d = 184 \)

- system M.P. \( Y_{kp}, Y_{k1}, Y_{k2}, \ldots, Y_{kp} \)

- number System M.P. \( Y_{\text{amp}} = Y_{\text{amp}} \)

- observer Gain M.P. \( Y^o_1, Y^o_2, \ldots, Y^o_{\text{amp}} \)

- form \( P_k = [Y_k \quad Y^o_k] \)

- recursive summation required \( \text{numsum} = \)

- matrix multiply \( \text{numsum} m [m (r + m)] + \)

- \( (\text{numsum} - p) \times p m [m (r + m)] \)

- \( p m m r \)

TOTAL FOR OKID \( 1.5E+08 \) flop

\[ 28 (1 + 2 + 3 + \ldots + p - 1) \]
Step 3  System Realization with Fast ERA

\( n = \text{order of realized system} \)

- separate \( P_k \) into \( Y_k \) and \( Y_k^* \)
- form \( H_0 = Y_k \)
- form \( D_0 = H_0 \, H_0^T \)
- matrix multiply
- form upper left partition \( D_{qL} \)
- eigenvalue decomposition of \( D_{qL} \)
- use all eigenvalues \( n = m \, p \)
- form pseudo-inverse of \( P_{q}^{(1)} \)
- matrix inverse
- matrix multiply
- form lower left partition \( D_{qL} \)
- form \( P_{q}^{(2)} \)
- matrix multiply
- matrix multiply (A)
- matrix multiply (B)
- matrix multiply (C)
- form \( D \) from \( \bar{Y} \)

\[
\begin{array}{ll}
H_0 & m \, (p+1) \times d \, r \\
D_0 & m \, (p+1) \times m \, (p+1) \\
D_{qL} & m \, p \times m \, p \\
\end{array}
\]

Step 4  Realize Observer Gain Matrix and Rescale

- construct observability matrix \( P \)
- matrix multiply
- observer gain matrix \( G = [P]^* \, Y^* \)
- matrix inverse
- matrix multiply
- scale matrices B, C, and G

\[
\begin{array}{ll}
P & n \times n \\
(p-1) \, m \, n \, n & 5.5E+06 \text{ flop} \\
Y^* & n \times m \\
G & n \times m \\
n^* & 6.2E+06 \text{ flop} \\
n \, n \, m & 7.8E+05 \text{ flop} \\
n \, r + m \, n + m \, m & 9.0E+03 \text{ flop} \\
\end{array}
\]

TOTAL FOR FAST ERA 7.0E+07 flop
TOTAL 2.2E+08 flop
Appendix D

ERA Variations

D.1 ERA/DC

The Eigensystem Realization Algorithm with Data Correlations (ERA/DC), as discussed by Juang [72], defines a data correlation matrix $R_j^{hh}$ as follows:

$$R_j^{hh} \triangleq H_j H_d^T$$  \hspace{1cm} (D.1)

where $H_j$ is the generalized $qm \times dr$ Hankel matrix defined by Equation (3.31). Using Equations (3.37) and (3.35), $R_j^{hh}$ can be re-written as:

$$R_j^{hh} = P_q A^j Q_d Q_d^T P_q^T = P_q A^j Q_c$$  \hspace{1cm} (D.2)

where $Q_c = Q_d Q_d^T P_q^T$. For the case where $j = 0$, the matrix $R_0^{hh}$ consists of Markov parameter auto-correlations as well as cross-correlations. If the Markov parameters are not correlated, the correlation matrix $R_0^{hh}$ will contain less noise than the Hankel matrix $H_0$ [72].
ERA/DC forms a block correlation Hankel matrix $\mathcal{H}_j$ as follows:

$$
\mathcal{H}_j = \begin{bmatrix}
R_{j}^{hh} & R_{j+\tau}^{hh} & \ldots & R_{j+\zeta\tau}^{hh} \\
R_{j+\tau}^{hh} & R_{j+2\tau}^{hh} & \ldots & R_{j+(\zeta+1)\tau}^{hh} \\
\vdots & \vdots & \ddots & \vdots \\
R_{j+\zeta\tau}^{hh} & R_{j+(\zeta+1)\tau}^{hh} & \ldots & R_{j+2(\zeta+1)\tau}^{hh}
\end{bmatrix}
$$

(D.3)

Substituting Equation (D.2) into the above equation yields:

$$
\mathcal{H}_j = \begin{bmatrix}
P_q \\
P_qA^\tau \\
\vdots \\
P_qA^{\zeta\tau}
\end{bmatrix}
A^j \begin{bmatrix}
Q_c \\
A^\tau Q_c \\
\vdots \\
A^{\zeta\tau} Q_c
\end{bmatrix} = P_\xi A^j Q_\zeta
$$

(D.4)

and

$$
\mathcal{H}_0 = P_\xi Q_\zeta \quad \mathcal{H}_1 = P_\xi A Q_\zeta
$$

(D.5)

The matrices $P_\xi$ and $Q_\zeta$ can be calculated by taking the singular value decomposition (SVD) of $\mathcal{H}_0$:

$$
\mathcal{H}_0 = P_\xi Q_\zeta = R\Sigma S^T
$$

(D.6)

Note that ERA performs an SVD of the $H_0$ matrix while ERA/DC uses the SVD of $\mathcal{H}_0$. $P_\xi$ and $Q_\zeta$ are chosen to satisfy Equation (D.6) as follows:

$$
P_\xi = R\Sigma^{1/2} \quad Q_\zeta = \Sigma^{1/2} S^T
$$

(D.7)

The state matrix $A$ is obtained from Equation (D.5) using a least-squares solution:

$$
A = P_\xi^\dagger \mathcal{H}_1 Q_\zeta^\dagger
$$

(D.8)
where the $\dagger$ indicates the pseudo-inverse. Using Equation (D.7), the pseudo-inverse of $P_\xi$ and $Q_\zeta$ can be calculated from:

$$P_\xi^{\dagger} = \Sigma^{-1/2} R^T \quad Q_\zeta^{\dagger} = S \Sigma^{-1/2} \quad (D.9)$$

Substituting Equation (D.9) into Equation (D.8) yields the following form of the state matrix $A$:

$$A = \Sigma^{-1/2} R^T K_1 S \Sigma^{-1/2} \quad (D.10)$$

Using the definition given in Equation (3.35), the controllability matrix $Q_d$ can be computed as:

$$Q_d = P_q^\dagger H_0 \quad (D.11)$$

The observability matrix $P_q$ can be obtained from Equation (D.4) using the definition given by Equation (3.43):

$$P_q = E_\gamma^T P_\xi = E_\gamma^T R \Sigma^{1/2} \quad (D.12)$$

where $\gamma = mq$.

Therefore, the controllability matrix $Q_d$ can be obtained by substituting Equation (D.12) into (D.11):

$$Q_d = [E_\gamma^T R \Sigma^{1/2}]^\dagger H_0 \quad (D.13)$$

By examining the structure of the controllability matrix $Q_d$ in Equation (3.34) and utilizing the definitions given by Equations (D.13) and (3.43), the $B$ matrix may be calculated from:

$$B = [E_\gamma^T R \Sigma^{1/2}]^\dagger H_0 E_r \quad (D.14)$$
Similarly, the C matrix may be obtained from the observability matrix $P_q$ (see Equations (3.34) and (D.12)) as follows:

$$C = E_m^T P_q = E_m^T E_1^T R \Sigma^{1/2}$$

(D.15)

The D matrix is obtained immediately from the system Markov parameters shown in Equation (3.5):

$$D = Y_0$$

(D.16)

**ERA/DC Special Case**

For the special case of ERA/DC, where $\tau = 0$, the block correlation Hankel matrix $H_j$ becomes:

$$H_j = R_j^{hh}$$

(D.17)

Following a similar procedure to ERA, ERA/DC first processes $H_0$ which, using Equations (D.17), (D.1) and (3.35), can be written as follows:

$$H_0 = R_0^{hh} = H_0 H_0^T = P_q Q_d Q_d^T P_q^T = P_q Q_c$$

(D.18)

where $Q_c = Q_d Q_d^T P_q^T$ and $P_q$ and $Q_d$ are the observability and controllability matrices respectively (see Equation (3.34)).

Similarly, $H_1$ can be written as:

$$H_1 = P_q A Q_c$$

(D.19)

ERA/DC performs a singular value decomposition of the block correlation Hankel matrix $H_0$:

$$H_0 = P_q Q_c = R \Sigma S^T$$

(D.20)
where the columns of the matrices $R$ and $S$ are orthonormal and $\Sigma$ is a diagonal matrix containing the singular values. Although the $R$ and $S$ computed for ERA/DC are different than for ERA, the singular values $\Sigma$ of $H_0$ are the squares of the singular values of $H_0$.

$P_q$ and $Q_c$ are chosen to satisfy Equation (D.20) as follows:

$$P_q = R \Sigma^{1/2} \quad Q_c = \Sigma^{1/2} S^T$$  \hspace{1cm} (D.21)

The state matrix $A$ is obtained from Equation (D.19) using a least-squares solution:

$$A = P_q^T H_1 Q_c^T$$  \hspace{1cm} (D.22)

where the $\dagger$ indicates the pseudo-inverse. Using Equation (D.21), the pseudo-inverse of $P_q$ and $Q_c$ can be calculated from:

$$P_q^\dagger = \Sigma^{-1/2} R^T \quad Q_c^\dagger = S \Sigma^{-1/2}$$  \hspace{1cm} (D.23)

Substituting Equation (D.23) into Equation (D.22) yields the following form of the state matrix $A$:

$$A = \Sigma^{-1/2} R^T H_1 S \Sigma^{-1/2}$$  \hspace{1cm} (D.24)

Using the definition given in Equation (3.35), the controllability matrix $Q_d$ can be computed as:

$$Q_d = P_q^T H_0$$  \hspace{1cm} (D.25)

which becomes, upon substitution of Equation (D.23):

$$Q_d = \Sigma^{-1/2} R^T H_0$$  \hspace{1cm} (D.26)
APPENDIX D. ERA VARIATIONS

By examining the structure of the controllability matrix $Q_d$ in Equation (3.34) and utilizing the definitions given by Equations (D.26) and (3.43), the $B$ matrix may be calculated from:

$$B = Q_d E_r = \Sigma^{-1/2} R^T H_0 E_r \quad (D.27)$$

Similarly, the $C$ matrix may be obtained from the observability matrix $P_q$ (see Equations (3.34) and (D.21)) as follows:

$$C = E_m^T P_q = E_m^T R \Sigma^{1/2} \quad (D.28)$$

The $D$ matrix is obtained immediately from the system Markov parameters shown in Equation (3.5):

$$D = Y_0 \quad (D.29)$$

Relationship between ERA/DC and ERA

Juang et al. [75] and Peterson [127] point out that ERA is equivalent to the special case of ERA/DC when $\tau = 0$.

Let the state-space realization for ERA (Equations (3.42), (3.44), and (3.45)) be represented as follows:

$$A_1 = \Sigma_1^{-1/2} R_1^T H_1 S_1 \Sigma_1^{-1/2} \quad (D.30)$$

$$B_1 = \Sigma_1^{1/2} S_1^T E_r \quad (D.31)$$

$$C_1 = E_m^T R_1 \Sigma_1^{1/2} \quad (D.32)$$
Let the state-space realization for the special case of ERA/DC (Equations (D.24), (D.27), and (D.28)) be represented as follows:

\[
\begin{align*}
A_2 &= \Sigma_2^{-1/2} R_2^T \mathcal{H}_1 S_2 \Sigma_2^{-1/2} \\
B_2 &= \Sigma_2^{-1/2} R_2^T H_0 E_r \\
C_2 &= E_m^T R_2 \Sigma_2^{1/2}
\end{align*}
\] (D.33) (D.34) (D.35)

In order to compare ERA/DC and ERA, the following observations are necessary:

- \( S_2 = R_2 \) \( R_1 = R_2 \) \( \Sigma_1 = \Sigma_2^{1/2} \)

- \( \mathcal{H}_1 = H_1 H_0^T \) \( H_0 = R_1 \Sigma_1 S_1^T \) \( H_0^T = S_1 \Sigma_1 R_1^T \)

Using the above observations, the following modifications can be made to the ERA/DC realization:

\[
\begin{align*}
A_2 &= \Sigma_2^{-1/2} R_2^T H_1 H_0^T S_2 \Sigma_2^{-1/2} \\
&= \Sigma_2^{-1/2} R_2^T H_1 S_1 \Sigma_1 R_1^T S_2 \Sigma_2^{-1/2} \\
&= \Sigma_1^{-1/2} R_1^T H_1 S_1 \Sigma_1 R_1^T S_2 \Sigma_2^{-1/2} \\
B_2 &= \Sigma_2^{-1/2} R_2^T R_1 \Sigma_1 S_1^T E_r \\
&= \Sigma_1^{-1/2} R_2^T R_1 \Sigma_1 S_1^T E_r \\
&= \Sigma_1^{-1/2} R_2^T R_1 \Sigma_1 S_1^T E_r \\
C_2 &= E_m^T R_2 \Sigma_1 \\
&= E_m^T R_2 \Sigma_1^{1/2} \Sigma_1^{1/2} \\
&= E_m^T R_2 \Sigma_1^{1/2} \\
&= C_1 T
\end{align*}
\] (D.36)

where \( T = \Sigma_1^{1/2} \). Therefore, the special case of ERA/DC and ERA are identical to within a diagonal coordinate transformation.

Equations (D.29) and (3.46) show that the \( D \) matrices are the same in both ERA/DC and ERA.
APPENDIX D. ERA VARIATIONS

D.2 Fast ERA

The Fast Eigensystem Realization Algorithm (Fast ERA), as described by Peterson [126], is based on the data matrix $D_q$ defined as follows:

$$D_q \triangleq R_0^{hh} = H_0 H_0^T$$  \hspace{1cm} (D.37)

where $H_0$ is the Hankel matrix having a structure defined by Equation (3.31).

Following a similar structure as Equation (3.35), $H_0$ can be written as follows:

$$H_0 = P_q Q_d$$  \hspace{1cm} (D.38)

Substituting Equation (D.38) into (D.37) yields:

$$D_q = H_0 H_0^T = P_q Q_d Q_d^T P_q^T$$  \hspace{1cm} (D.39)

Assuming $Q_d Q_d^T = I$ allows $D_q$ to be written as:

$$D_q = P_q P_q^T$$  \hspace{1cm} (D.40)

Note that this assumption specifies the coordinate basis for the realized state-space matrices. The eigenvalue decomposition of $D_q$ results in:

$$D_q = E \Lambda E^{-1}$$  \hspace{1cm} (D.41)

where $\Lambda$ is a diagonal matrix of the eigenvalues and $E$ is the eigenmatrix of eigenvectors. Comparing Equations (D.40) and (D.41) results in the following definition for $P_q$:

$$P_q = E \Lambda^{1/2}$$  \hspace{1cm} (D.42)
The previous equality can be made due to the structure of $D_q$. In general, for a given matrix $A$ the eigenvectors $E$ of $AA^T$ (or for that matter, $A^T A$) are orthonormal. A matrix $E$ is orthonormal if $E^T E = I$.

The state-space matrices are obtained similar to Q-Markov CovER:

\[
A = P_q^{(1)\dagger} P_q^{(2)} \quad B = P_q^{(1)\dagger} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_{q-1} \end{bmatrix} \quad C = E^T_m P_q^{(1)} \quad \text{(D.43)}
\]

where $\dagger$ is the pseudo-inverse.

The $D$ matrix is obtained immediately from the system Markov parameters shown in Equation (3.5):

\[
D = Y_0 \quad \text{(D.44)}
\]
Relationship between Fast ERA and ERA/DC

Peterson [126] demonstrates that Fast ERA is equivalent to the special case of ERA/DC where \( \tau = 0 \) (and, therefore, Fast ERA is equivalent to ERA to within a diagonal coordinate transformation). To show Fast ERA’s equivalence to the special case of ERA/DC, \( D_q \) should have an additional block row and be partitioned as follows:

\[
D_q = P_q P_q^T = \begin{bmatrix}
P_q^{(1)} \\
P_q
\end{bmatrix}
\begin{bmatrix}
P_q^{(1)T} & P_q^T \\
P_q^{(1)T} & P_q^T
\end{bmatrix}
\tag{D.45}
\]

\[
= \begin{bmatrix}
P_q^{(1)} P_q^{(1)T} & P_q^{(1)} P_q^T \\
P_q P_q^{(1)T} & P_q P_q^T
\end{bmatrix}
\tag{D.46}
\]

where \( P_q = CA^{q-1} \) and different partitions of the observability matrix \( P_q \) are defined in Equation (3.67).

The upper left \( q - 1 \) by \( q - 1 \) partition of \( D_q \) (which is equivalent to \( H_0 \) in the special case of ERA/DC) can be decomposed as follows:

\[
D_q^{UL} = P_q^{(1)} P_q^{(1)T} = E \Lambda E^T
\tag{D.47}
\]

where \( \Lambda \) is redefined as a diagonal matrix of the eigenvalues of the upper left partition of \( D_q \) and \( E \) is the corresponding eigenmatrix of eigenvectors.

Similar to Equation (D.42), \( P_q^{(1)} \) can be defined as:

\[
P_q^{(1)} = E \Lambda^{1/2}
\tag{D.48}
\]
The pseudo-inverse of $P_q^{(1)}$ becomes:

$$P_q^{(1)^\dagger} = \Lambda^{-1/2}E^T$$ \hspace{1cm} (D.49)

The $D_q$ matrix can also be partitioned as follows:

$$D_q = P_q P_q^T = \begin{bmatrix} P_0 \\ P_q^{(2)} \end{bmatrix} \begin{bmatrix} P_q^{(1)^T} & P_q^T \end{bmatrix}$$ \hspace{1cm} (D.50)

$$= \begin{bmatrix} P_0 P_q^{(1)^T} & P_0 P_q^T \\ P_q^{(2)} P_q^{(1)^T} & P_q^{(2)} P_q^T \end{bmatrix}$$ \hspace{1cm} (D.51)

where $P_0 = C$, $P_q = CA^{-1}$ and different partitions of the observability matrix $P_q$ are defined in Equation (3.67).

The lower left partition of $D_q$ (which is equivalent to $H_1$) can be written as follows:

$$D_q^{LL} = P_q^{(2)} P_q^{(1)^T}$$ \hspace{1cm} (D.52)

Using this relationship, $P_q^{(2)}$ can be determined from:

$$P_q^{(2)} = D_q^{LL} P_q^{(1)^T}$$ \hspace{1cm} (D.53)

Equation (D.49) allows the matrix $P_q^{(1)^T}$ to be written as:

$$P_q^{(1)^T} = E\Lambda^{-1/2}$$ \hspace{1cm} (D.54)

Substituting Equations (D.54), (D.53), and (D.49) into Equation (D.43) gives the following realization for the $A$ matrix:

$$A = \Lambda^{-1/2}E^T D_q^{LL} E\Lambda^{-1/2}$$ \hspace{1cm} (D.55)
The realization for $B$ given by Equation (D.43) can be rewritten by substituting in Equations (D.49), (3.43) and (3.32):

$$B = \Lambda^{-1/2}E^T\mathcal{H}_0E_r$$  \hspace{1cm} (D.56)

Substituting Equations (D.48) into Equation (D.43) gives the following realization for $C$:

$$C = E_m^T\mathcal{A}^{1/2}$$  \hspace{1cm} (D.57)

The Fast ERA realizations for $A$, $B$, and $C$ are equivalent to the realizations for the special case of ERA/DC for the following reasons:

- The data matrices $\mathcal{H}_0$ and $D^{UL}$ are identical.
- The data matrices $\mathcal{H}_1$ and $D^{LL}$ are identical.
- Due to the structure of $\mathcal{H}_0$ and $D^{UL}$, the singular value decomposition in Equation (D.20) and the eigenvalue decomposition in Equation (D.47) are related as follows:
  
  * $\Sigma = \Lambda$
  * $R = S = E$
  * $R^T = S^T = E^{-1}$

**Relationship between Fast ERA and Q-Markov CovER**

The derivations of Fast ERA and Q-Markov CovER are very similar. Apparent differences are the data matrices $D_q$, the eigenvalue decomposition (Fast ERA), and the singular value decomposition (Q-Markov CovER). The
relationship between the data matrices for Fast ERA and Q-Markov CovER is now described.

Fast ERA uses the data matrix $D_q$ shown in Equation (D.37). The $D_q$ matrix is composed of a Hankel matrix $H_0$ having $q$ rows and $d$ columns. The resulting data matrix used by Fast ERA is now defined as:

$$D_q^d \triangleq H_0 H_0^T$$  \hspace{1cm} (D.58)

Q-Markov CovER is based on the data matrix $D_q$ shown in Equation (3.64). The data matrix requires covariance parameters which have the deterministic formula given by Equation (3.59). The resulting data matrix used by Q-Markov CovER is now defined as:

$$D_q^\infty \triangleq R_q - H_q H_q^T$$  \hspace{1cm} (D.59)

where $R_q$ and $H_q$ are defined by Equations (3.61) and (3.62).

Peterson and Bullock [127] have observed that $D_q^d$ and $D_q^\infty$ are equivalent only in the limit as $d \to \infty$. For a finite $d$ (the practical case), the Q-Markov CovER data matrix $D_q^\infty$ differs fundamentally from the outer product $D_q^d$.

The difference between $D_q^\infty$ and $D_q^d$ can be demonstrated using a simple example with $d = 2$ and $q = 3$ as follows:
The Toeplitz matrix $R_q$ is:

$$\mathbf{R}_q = \begin{bmatrix} Y_0 Y_0^T + Y_1 Y_1^T + Y_2 Y_2^T & Y_0 Y_1^T + Y_1 Y_2^T + Y_2 Y_3^T & Y_0 Y_2^T + Y_1 Y_3^T + Y_2 Y_4^T \\ Y_1 Y_0^T + Y_2 Y_1^T + Y_3 Y_2^T & Y_0 Y_0^T + Y_1 Y_1^T + Y_2 Y_2^T & Y_0 Y_1^T + Y_1 Y_2^T + Y_2 Y_3^T \\ Y_2 Y_0^T + Y_3 Y_1^T + Y_4 Y_2^T & Y_1 Y_0^T + Y_2 Y_1^T + Y_3 Y_2^T & Y_0 Y_0^T + Y_1 Y_1^T + Y_2 Y_2^T \end{bmatrix}$$

where $Y_k$ are the system Markov parameters. The triangular matrices $\mathbf{H}_q$ and $\mathbf{H}_q^T$ are:

$$\mathbf{H}_q = \begin{bmatrix} Y_0 & 0 & 0 \\ Y_1 & Y_0 & 0 \\ Y_2 & Y_1 & Y_0 \end{bmatrix} \quad \mathbf{H}_q^T = \begin{bmatrix} Y_0^T & Y_1^T & Y_2^T \\ 0 & Y_0^T & Y_1^T \\ 0 & 0 & Y_0^T \end{bmatrix}$$

The resulting data matrix $D_q^\infty$ used by Q-Markov CovER is, from Equation (D.59):

$$D_q^\infty = \begin{bmatrix} Y_1 Y_1^T + Y_2 Y_2^T & Y_1 Y_2^T + Y_2 Y_3^T & Y_1 Y_3^T + Y_2 Y_4^T \\ Y_2 Y_1^T + Y_3 Y_2^T & Y_2 Y_2^T & Y_2 Y_3^T \\ Y_3 Y_1^T + Y_4 Y_2^T & Y_3 Y_2^T & 0 \end{bmatrix}$$

If, on the other hand, $D_q^d$ is computed using:

$$H_0 = \begin{bmatrix} Y_1 & Y_2 \\ Y_2 & Y_3 \\ Y_3 & Y_4 \end{bmatrix} \quad H_0^T = \begin{bmatrix} Y_1^T & Y_2^T & Y_3^T \\ Y_2^T & Y_3^T & Y_4^T \end{bmatrix}$$

the resulting data matrix $D_q^d$ is, from Equation (D.58):

$$D_q^d = \begin{bmatrix} Y_1 Y_1^T + Y_2 Y_2^T & Y_1 Y_2^T + Y_2 Y_3^T & Y_1 Y_3^T + Y_2 Y_4^T \\ Y_2 Y_1^T + Y_3 Y_2^T & Y_2 Y_2^T + Y_3 Y_3^T & Y_2 Y_3^T + Y_3 Y_4^T \\ Y_3 Y_1^T + Y_4 Y_2^T & Y_3 Y_2^T + Y_4 Y_3^T & Y_3 Y_3^T + Y_4 Y_4^T \end{bmatrix}$$
APPENDIX D. ERA VARIATIONS

Clearly, \( D_q^\infty \) and \( D_q^d \) are different for a finite value of \( d \).

In fact, for finite \( d \), the data matrix \( D_q^\infty \) may not be as accurate (in the case of lightly damped systems) as the data matrix \( D_q^d \). While the latter depends only on the accuracy of the Markov parameters, the former depends on not only the accuracy of the Markov parameters but also on the accuracy of the Toeplitz matrix \( R_q \). This Toeplitz matrix may not be very accurate for finite \( d \) since it is constructed using the covariance parameters \( R_k \) in Equation (3.59).

Based on simulations and experience with Daisy, the covariance parameters \( R_k \) will be accurate for small values of \( d \) only when the structure is highly damped. In this case, the Markov parameters quickly approach zero and the impulse responses can be captured even for small \( d \). On the other hand, if the structure is lightly damped, the Markov parameters approach zero very slowly and the choice of small \( d \) results in poor approximations for \( R_k \).
Appendix E

SI Software

The present author designs and writes system identification software in Xmath [64] to facilitate the required comparisons and analyses. The OKID module is based on MATLAB software written by Dr. Horta and Dr. Juang of the NASA Spacecraft Dynamics Branch. The Q-Markov CovER module is based on MATLAB software provided by Dr. Skelton of the Space Systems Control Laboratory at Purdue University.

The Xmath SI software implements the algorithms selected in § 2.3 including the variations of OKID and ERA shown in Figure 4.1.

E.1 Running the Software

The software is executed by typing si at the Xmath command prompt in the main directory. The graphical user interface and modular code make the program user friendly. A description of each variable is included within the code.
The main window is shown in Figure E.1. There are several parameters which must be entered before the SI process can begin. These include:

- The number of system inputs and outputs.
- The sample time.
- The observer order $p$ which is related to the number of block rows used in the data matrix.
- The number of columns in the data matrix is equal to $X$ times the number of rows in the data matrix.

The parameter $p$ is also related to the size of the identified state-space matrices. Section 3.2 explains that $p$ should be chosen such that $mp = n$ where $m$ is the number of system outputs and $n$ is the order of the realized state-space system. As the integer $p$ increases, the realized model order increases.

The main window allows the user to specify whether or not the SI experiment(s) started from rest. The default value for this selection assumes that only one experiment is used for SI and that it started from rest. If more than one experiment is used, then this should be indicated by selecting no. At this point, a pop-up window appears showing all experiments currently in memory. The user should select those experiments which did not start from rest and press the $OK$ button.
Figure E.1: Main System Identification Window
Note that OKID is required for ERA (c C), ERA/DC, the special case of ERA/DC (c C) described in § 4.1, Fast ERA and Q-Markov CovER. ORSE and Subspace do not require OKID.

E.2 Software Modes

The software can be executed in three modes: interactive, non-interactive and batch. Interactive mode allows the user to control all parameters at each step in the process while non-interactive mode lets the computer allocate appropriate values for certain parameters. The batch mode performs background SI and is especially useful when many analyses are required on a given data set. It is not selectable from the main menu.

Interactive Mode

When the interactive mode is selected, windows appear during the SI process which allow the user to:

- select either concurrent identification of the observer gain or the consecutive method (both approaches are described in § 4.1).

- load in previously saved observer Markov parameters. When the program questions whether or not OKID has been run before, it offers three different choices. If Yes is selected, then the program assumes that the previous execution of OKID used the same parameters displayed on the main menu. If the number of Markov parameters needs to be changed, then select Markov. If any other parameters have been
changed since the previous run, then no should be selected and the program reprocesses all data.

- change the number of system Markov parameters. By default, OKID chooses the number of system Markov parameters based on the size of the Hankel matrix (specified by \( p \) and \( X \) in the main menu).

- iteratively select the model order based on the singular value decomposition results. Section 5.1.1 demonstrated that singular values of the data matrix can be used as a form of model order reduction. Large singular values correspond to an identified mode which makes a significant contribution to the impulse response history. Small singular values, on the other hand, are often due to the effects of noise and should be removed.

When the user is first prompted for the desired model order, the largest model order possible (given the observer order \( p \)) is displayed. The model order prompt remains on the screen to allow the user to efficiently test different model orders without reprocessing all of the data.

- trim the final model by selecting the modes to be removed. Once a desired model order has been selected, a window appears displaying the modal parameters in order of increasing frequency. The user can automatically truncate all unstable modes by pressing the Remove Unstable Modes button. Modes can be selected with the mouse and then removed from the model by pressing the Remove Selected Modes button. Section E.6 explains how the modes are isolated and removed.
The left mouse button in conjunction with the Shift key allows a consecutive group of modes to be selected. Select the first mode in the group and then, while pressing the Shift key, select the last desired mode. All modes in between will be selected. Multiple modes can be individually selected using the left mouse button in conjunction with the Control key. Once all undesirable modes have been removed, the Finished button is pressed and the realized model can be examined using the Plot menu discussed in § E.3.

Non-interactive Mode

In non-interactive mode only the first two windows used in the interactive mode appear. The user must select either concurrent identification of the observer gain or the consecutive method. Then the user may indicate whether OKID has been run before to reduce the number of computations required when SI analyses are repeated.

Once these options have been selected, the software automatically determines the number of system Markov parameters based on the observer order \( p \) and integer \( X \) entered in the main window. The program then automatically truncates the singular values below a given threshold to select the final system order. As discussed in § 5.1.1, if the contribution of a singular value is less than \( 10^{-10} \) of the sum of all singular values then it is considered negligible and is truncated.
Batch Mode

The SI software can be run in the background to perform large SI analyses where several parameters are to be varied and/or several methods are to be used. In order to initialize this process, the script file batch.msc must be modified to set the appropriate parameters and conditions. To begin the background process go to the main directory and, at the UNIX prompt, type:

```
nohup xmath -tty < begin.ms > OUTPUT.FILE
```

where OUTPUT.FILE is the name of a log file where all output is directed. The realized models are saved in separate files.

E.3 Menus

The following sections describe the different menus available to the user from the main window.

File Menu

The File menu allows the user to load the system input and output matrices from a data file using the Load Input Data and Load Output Data options. Alternatively, the input and output matrices can be created from Xmath variables by selecting Create Input Data or Create Output Data. Previously saved full- and reduced-order realized systems can be retrieved for further
processing by selecting Load Full Order System or Load Reduced Order System. If the user wishes to reload the data into memory use the Restore I/O Data option. This menu also allows the user to Exit the program.

The input/output data to be imported with the Load Input Data or Load Output Data option must be an Xmath.xmd file created with the Xmath save command. The variable names being loaded must be matrices called main.input.data or main.output.data. The rows in the data represent the time history and the columns correspond to different input or output channels. Note that more than one experiment can be used to identify a system (§ 4.1.2). If this feature is desired, other experimental data can be added as additional columns in the input and output data matrices.

**Technique Menu**

The Technique menu begins the SI process using the above user-defined parameters. If Algorithm is selected, then the software skips the OKID step, assuming that the system Markov parameters are available, and the selected algorithm is implemented. If system Markov parameters are unavailable, then the OKID option should be selected. Note that Subspace and ORSE algorithms are also initiated using the Technique menu.
Reduce Menu

The Reduce menu is used for model order reduction once the software has finished realizing the full-order system. The different options available represent different Xmath functions used for model order reduction. If the non-interactive mode is desired, then the option exists to select a model reduction technique and have the realized model be automatically reduced to a desired order. The Manual option in this menu allows the user to manually remove modes from any system currently in memory. Section E.6 explains how the modes are isolated and removed. The Balmoore_ext. option implements an extension to the balmoore algorithm developed by the present author (see § 5.2.4).

Stabilize Menu

The Stabilize menu allows the user to, given any unstable system, force the discrete system eigenvalues to have magnitude less than one. This stabilizing process is described in § 5.2.3.

Plot Menu

The Plot menu is used once the software has finished identifying the system matrices. It allows the user to plot input and output data. Comparisons between the desired system output and calculated system output can be performed as shown in Figure E.2. If multiple experiments are used, the
experiment number can be selected. The system can be simulated with or without the observer and error plots are also available. Once satisfied with the selections, the Begin Simulation button should be pressed. After the simulation is complete, different outputs can be examined and compared. Portions of the data can be observed by specifying the start and end data points. Pressing the Prediction Error button causes the model prediction error for each output to be plotted (see Equation (3.54)).

The eigenvalues of the actual or realized discrete-time system can be displayed for comparative purposes. If the target eigenvalues of the continuous system are known, then they can be loaded into memory using the Plot menu. The target eigenvalues must be saved as an Xmath .xmd file created with the Xmath save command. The variable being loaded must be an Xmath Parameter-Dependent-Matrix (PDM) and named main.system.pdm.

### E.4 Software Directories and Partitions

The software is organized into the following directories:

**data** This directory contains the input and output data saved in an Xmath .xmd file format. It also contains the .pdm file used to compare the realized and target eigenvalues.

**file** This directory contains Xmath command files .msc which load the input and output data to be used in the SI process.

**main** This directory contains the main Xmath command file called si.msc.
Figure E.2: Output Comparison Windows
This file controls the sequence of events when the SI software is executed. The directory also contains the resource file $Si$ which defines the windows required for the graphical user interface (GUI). Xmath should be initially run in this directory. This directory contains the files $\text{begin.ms}$ and $\text{batch.msc}$ which are used to run the SI software in batch mode. Any files saved during the execution of the SI process will be stored in this directory.

$\text{plot}$ This directory contains all the functions used by the $\text{plot}$ menu option. These functions are used after the completion of the SI process.

$\text{okid}$ This directory contains the functions associated with the OKID algorithm.

$\text{fast-era}$ This directory contains the functions necessary for the Fast ERA algorithm.

$C$ This directory contains the C language singular value decomposition routine used by the Xmath SI software.

$qmc$ This directory contains the functions required for the Q-Markov CovER algorithm.

$orse$ This directory contains the functions required for the ORSE algorithm.

$\text{subspace}$ This directory contains the functions required for the Subspace algorithm.

All variables are placed in Xmath $\text{partitions}$ as follows:
**main** This is the default partition and contains the realized system matrices: 
\[ A_d, B_d, C, D, G \]

**data** This partition contains the input/output data used by the SI software.

**parameters** This partition contains all parameters used by the SI software including:

- number of system inputs
- number of system outputs
- sample time
- observer order \( p \)
- the parameter \( X \) related to the number of columns in the data matrix
- number of system Markov parameters
- system Markov parameters

**plot** This partition contains all plot parameters used in conjunction with the plot menu.

**interact** This partition contains all variables associated with user interactions and selections during the SI process.

### E.5 Modal Parameters

During model order selection, the damping and frequencies are displayed to help the user eliminate undesirable identified modes.
The eigenvalues of the continuous-time system $\lambda_c$ are related to the eigenvalues of the discrete-time system $\lambda_d$ as shown in Equation (5.3). The eigenvalues of a system exhibiting viscously damped oscillatory motion have the following form:

$$\lambda_c = -\zeta \omega_n \pm \omega_n \sqrt{1 - \zeta^2} \ j \quad (E.1)$$

where $\omega_n$ is the natural frequency and $\zeta$ is the damping ratio. The natural frequency $\omega_n$, in radians per second, is determined by taking the magnitude of $\lambda_c$. Dividing by $2\pi$ converts the frequency to Hz. Each damping ratio $\zeta$ is calculated by taking the negative of the real part of the eigenvalue and dividing it by $\omega_n$.

**E.6 Mode Removal**

In order to reduce the size of the realized model, the software allows unstable modes to be removed (Remove Unstable Modes button), or user-selected modes to be removed (Remove Selected Modes button).

The modes are isolated by converting the realized state-space matrices into modal coordinates. Vegte [162] explains how to perform this coordinate transformation.
If the original state-space system is of the form:

\[
\dot{x} = Ax + Bu
\]
\[
y = Cx
\]  
and the coordinate transformation is \( x = Tz \), then the transformed system is of the form:

\[
\dot{z} = \hat{A}z + \hat{B}u
\]
\[
y = \hat{C}z
\]  
where \( \hat{A} = T^{-1}AT \), \( \hat{B} = T^{-1}B \) and \( \hat{C} = CT \).

To convert to modal coordinates, the transformation, \( T = U \) which makes \( \hat{A} = U^{-1}AU = \Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n) \) is desired. The elements \( \lambda_i \) of \( \Lambda \) are the eigenvalues.

Once in modal coordinates, it is possible to remove undesirable modes by eliminating the corresponding rows and columns of the state-space matrices.
Appendix F

Cubic Smoothing Spline

Results
Figure F.1: Directly identifying models of order $n = 46$: hub roll
Figure F.2: Directly identifying models of order $n = 46$: hub pitch
Plot A: OKID and Fast ERA with $p=2$, $n=46$

Plot B: OKID with $p=10$, Fast ERA with $q=2$, $n=46$

Plot C: OKID with $p=10$, Cubic Spline, Fast ERA with $q=2$, $n=46$

Figure F.3: Directly identifying models of order $n = 46$: hub yaw
Figure F.4: Directly identifying models of order $n = 46$: rib 1 OC
Figure F.5: Directly identifying models of order $n = 46$: rib 1 IC
Figure F.6: Directly identifying models of order $n = 46$: rib 2 OC
Plot A: OKID and Fast ERA with $p=2$, $n=46$

Plot B: OKID with $p=10$, Fast ERA with $q=2$, $n=46$

Plot C: OKID with $p=10$, Cubic Spline, Fast ERA with $q=2$, $n=46$

Figure F.7: Directly identifying models of order $n=46$: rib 2 IC
Figure F.8: Directly identifying models of order $n = 46$: rib 3 OC
Figure F.9: Directly identifying models of order $n = 46$: rib 3 IC
Figure F.10: Directly identifying models of order $n = 46$: rib 4 OC
Figure F.11: Directly identifying models of order $n = 46$: rib 4 IC
Figure F.12: Directly identifying models of order $n = 46$: rib 5 OC
Figure F.13: Directly identifying models of order $n = 46$: rib 5 IC
Figure F.14: Directly identifying models of order $n = 46$: rib 6 OC
Figure F.15: Directly identifying models of order $n = 46$: rib 6 IC
Figure F.16: Directly identifying models of order $n = 46$: rib 7 OC
Figure F.17: Directly identifying models of order $n = 46$: rib 7 IC
Figure F.18: Directly identifying models of order $n = 46$: rib 8 OC
Figure F.19: Directly identifying models of order $n = 46$: rib 8 IC
Figure F.20: Directly identifying models of order $n = 46$: rib 9 OC
Figure F.21: Directly identifying models of order $n = 46$: rib 9 IC
Figure F.22: Directly identifying models of order \( n = 46 \): rib 10 OC
Figure F.23: Directly identifying models of order $n = 46$: rib 10 IC
Bibliography


BIBLIOGRAPHY

811046, SEA Aerospace Congress and Exposition, Anaheim, California, 1981.


[107] Longman, R.W., Lew, J.-S., Juang, J.-N., *Comparison of Candidate Methods to Distinguish Noise Modes from System Modes in Struc-


[118] Nguyen, P., Selection of Controller Design Methods and System Identification Methods for DICE, Memorandum No. DICE-PN-7, University of Toronto Institute for Aerospace Studies, April 30, 1996.


