UNIVERSITY OF CALIFORNIA, SAN DIEGO

Realization-Based System Identification with Applications

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Daniel N. Miller

Committee in charge:

Professor Raymond A. de Callafon, Chair
Professor Thomas R. Bewley
Professor Robert R. Bitmead
Professor Joel P. Conte
Professor Michael D. Todd

2012

The dissertation of Daniel N. Miller is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2012

DEDICATION

To Karen.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ABBREVIATIONS

AAW	Active Aeroelastic Wing
AFRL	Air Force Research Laboratories
ARX	Autoregressive Exogenous Input
ERA	Eigensystem Realization Algorithm
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LED	Light-emitting diode
LMI	Linear matrix inequality
MOESP	Multivariable Output Error State Space
N4SID	Numerical Methods for Subspace Identification
NASA	National Aeronautics and Space Administration
OE	Output Error
PBSID	Predictor-Based Subspace Identification
PI-MOESP	Past-Input Multivariable Output Error State Space
PO-MOESP	Past-Output Multivariable Output Error State Space
PSD	Power Spectral Density
SBR	Step-Based Realization
SVD	Singular-Value Decomposition

SETS

$\mathbb{R}$	Real numbers
$\mathbb{C}$	Complex numbers
$\mathbb{Z}$	Integers

VARIABLES AND SIGNALS

$u(t)$	Input with discrete index $t \in \mathbb{Z}$
$y(t)$	Output with discrete index $t \in \mathbb{Z}$
$R_x(\tau)$	Auto-covariance function of the signal $x(t)$, $\tau \in \mathbb{Z}$
$R_{xy}(\tau)$	Cross-covariance function of the signal $x(t)$ with the signal $y(t)$, $\tau \in \mathbb{Z}$
$\mathcal{C}$	Controllability matrix
$\mathcal{C}_i$	Extended controllability matrix with i block columns
$\mathcal{O}$	Observability matrix
$\mathcal{O}_i$	Extended observability matrix with i block rows
δ_a	Differential aileron position
δ_a^c	Differential aileron command
δ_r	Composite rudder position
δ_r^c	Composite rudder command

NORMS, OPERATORS, FUNCTIONS, AND INDEXING

$\|\|\cdot\|\|_2$	Euclidian norm of a vector or induced norm of a matrix
$\|\|\cdot\|\|_F$	Frobenius norm of a matrix
$E(\cdot)$	Expectation of a parameter
$\lambda(A)$	Spectrum of the matrix A
$\otimes$	Kronecker product
$\text{vec}(A)$	Vectorization that stacks columns of A sequentially
$A_{(i:j,l:k)}$	The sub-matrix of A with rows i to j and columns l to k
$\text{Re}(\cdot)$	Real part of a scalar, vector, or matrix
$\text{Im}(\cdot)$	Imaginary part of a scalar, vector, or matrix
$\text{trace}(\cdot)$	Trace of a matrix

ACKNOWLEDGEMENTS

I would like to thank my advisor, Raymond A. de Callafon, for his support and guidance. I would also like to thank my committee members, professors Thomas Bewley, Robert R. Bitmead, Joel P. Conte, and Michael D. Todd.

Martin J. Brenner of NASA Dryden Flight Research Center was a pleasure to work with and provided the dataset for NASA's Active Aeroelastic Wing F/A-18. Jie Zeng of ZONA graciously provided aeroelastic simulation data for application development and provided valuable feedback on the software developed to analyze the AAW data.

I would like to thank Seth Lacy of the Air Force Research Laboratories for hosting me as a Space Scholar at Kirtland AFB, NM for the summer of 2010, as well as Vit Babuska and Thomas Paez (ret.) of Sandia National Laboratories for their valuable collaborations. I learned a great deal from their valuable advice and friendship.

Jeff Hulett and James McLaughlin of Vektrex Corporation of San Diego provided a fascinating research problem that led to the constrained step-based realization algorithm presented in this paper. They also provided the LED thermal response dataset used to demonstrate the method.

I would like to thank the UCSD librarians, as well as the librarians of North Carolina State University for scanning and emailing me a copy of Kung [38].

General Atomics Aeronautical Systems supported my tuition for the fall of 2007 when I began graduate school as a part-time M.S. student. I thank my colleagues in industry for their guidance and encouragement to pursue a graduate degree. My experiences there made me the engineer I am today.

Friends, colleagues, and everyone else at UCSD made the long, tiring hours spent on campus bearable; there are too many names to list. I also thank the many fine breweries of San Diego, and Southwest Airlines for flying me and my wife to and from Oakland on a weekly basis.

The results included in this dissertation have been published in the following articles:

D.N. Miller and R.A. de Callafon, "Identification of Linear Discrete-Time Filters

via Realization," to appear in *Linear Algebra*. H. Yasser, Ed. InTech: 2012.

D.N. Miller, J. Hulett, J. McLaughlin, and R.A. de Callafon, "Parametric Estimation of Step Responses Applied to Power Electronics," submitted to *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 2012.

D.N. Miller and R.A. de Callafon, "Subspace Identification with Eigenvalue Constraints," submitted to *Automatica*, 2012.

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," to appear in *AIAA Journal of Guidance, Control and Dynamics*, 2012.

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "Eigenvalue Constraints for Realization-Based Identification," to appear, *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, MN: August 2012.

D.N. Miller and R.A. de Callafon, "Step-Based Realization with Eigenvalue Constraints," to appear, *16th IFAC Symposium on System Identification*, Brussels, Belgium: July 2012.

D.N. Miller and R.A. de Callafon, "Efficient Identification of Input Dynamics for Correlation Function-Based Subspace Identification," *Proc. of the 18th World Congress of the International Federation of Automatic Control*, Milan, Italy: August 2011.

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," *Proc. of the AIAA Atmospheric Flight Mechanics Conference*, Portland, OR: August 2011.

D.N. Miller and R.A. de Callafon, "Subspace Identification Using Dynamic Invariance in Shifted Time-Domain Data," *Proc. of the 49th IEEE Conference on Decision and Control*, Atlanta, GA: December 2010.

D.N. Miller and R.A. de Callafon, "Subspace Identification from Classical Realization Methods," *Proc. of the 15th IFAC Symposium on System Identification*, Saint-Malo, France: July 2009.

The dissertation author was the primary investigator and author of these publications.

VITA

2000-2001	Student Research Assistant Lockheed Martin Space Systems Palo Alto, CA
2005	B. S. in Mechanical Engineering University of California, San Diego
2005-2006	Process Engineer Solid State Stamping, Inc. Temecula, CA
2006-2009	Software Engineer, Flight Controls Group General Atomics Aeronautical Systems Inc. San Diego, CA
2009	Gordon Engineering Leadership Center Scholar
2009-2010	Graduate Teaching Assistant University of California, San Diego
2010	M. S. in Mechanical Engineering University of California, San Diego
2010	Air Force Space Scholar Air Force Research Laboratories, Space Vehicles Directorate Albuquerque, NM
2011	Gordon Engineering Leadership Center Fellow
2010-2012	Graduate Student Research Assistant University of California, San Diego
2012	Ph.D. in Mechanical Engineering University of California, San Diego

PUBLICATIONS

BOOK CHAPTERS

B1. D.N. Miller and R.A. de Callafon, "Identification of Linear Discrete-Time Filters via Realization" to appear in *Linear Algebra* (working title). H. Yasser, Ed. InTech: 2012.

JOURNAL ARTICLES

J1. D.N. Miller, J. Hulett, J. McLaughlin, and R.A. de Callafon, "Thermal Dynamical Identification of LEDs by Step-Based Realization and Convex Optimization," submitted to *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 2012.

J2. D.N. Miller and R.A. de Callafon, "Subspace Identification with Eigenvalue Constraints," to appear in *Automatica*, 2012.

J3. D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," to appear in *AIAA Journal of Guidance, Control and Dynamics*, 2012.

CONFERENCE PROCEEDINGS

C1. D.N. Miller, R.A. de Callafon, and M.J. Brenner, "Eigenvalue Constraints for Realization-Based Identification," to appear, *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, MN: August 2012.

C2. D.N. Miller and R.A. de Callafon, "Step-Based Realization with Eigenvalue Constraints," to appear, *16th IFAC Symposium on System Identification*, Brussels, Belgium: July 2012.

C3. D.N. Miller and R.A. de Callafon, "Efficient Identification of Input Dynamics for Correlation Function-Based Subspace Identification," *Proc. of the 18th World Congress of the International Federation of Automatic Control*, Milan, Italy: August 2011.

C4. D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," *Proc. of the AIAA Atmospheric Flight Mechanics Conference*, Portland, OR: August 2011.

C5. T.L. Paez, S.L. Lacy, V. Babuska, and D.N. Miller, "Improved Stochastic Process Models for Linear Structure Behavior," *Proc. of the 2011 American Controls Conference*, San Francisco, CA: June 2011.

C6. D.N. Miller and R.A. de Callafon, "Subspace Identification Using Dynamic Invariance in Shifted Time-Domain Data," *Proc. of the 49th IEEE Conference on Decision and Control*, Atlanta, GA: December 2010.

C7. R.A. de Callafon, D.N. Miller, J. Zeng, and M.J. Brenner, "Covariance Function Realization Algorithms for Aeroelastic Dynamic Modeling," *Proc. of the AIAA Atmospheric Flight Mechanics Conference*, Toronto, Canada: August 3, 2010.

C8. R.A. de Callafon, D.N. Miller, J. Zeng, and M.J. Brenner, "Step-Based Experiment Design and System Identification for Aeroelastic Dynamic Modeling," *Proc. of the AIAA Atmospheric Flight Mechanics Conference*, Chicago, IL: August 12, 2009.

C9. D.N. Miller and R.A. de Callafon, "Subspace Identification from Classical Realization Methods," *Proc. of the 15th IFAC Symposium on System Identification*, Saint-Malo, France: July 2009.

C10. N. Delson, T. Hanak, K. Loewke, and D.N. Miller, "Modeling and implementation of McKibben actuators for a hopping robot," *Proc. of the International Conference on Advanced Robotics*, Seattle, WA: July 2005.

ABSTRACT OF THE DISSERTATION

Realization-Based System Identification with Applications

by

Daniel N. Miller

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2012

Professor Raymond A. de Callafon, Chair

The identification of dynamic system behavior from experimentally measured or computationally simulated data is fundamental to the fields of control system design, modal analysis, and defect detection. In this dissertation, methods for system identification are developed based on classical linear system realization theory. The common methods of state-space realization from a measured, discrete-time impulse response are generalized to the following additional types of experiments: measured step responses, arbitrary sets of input-output data, and estimated cross-covariance functions of input-output data. The methods are particularly well suited to systems with large input and/or output dimension, for which classical system identification methods based on maximum likelihood estimation may fail due to their reliance on non-convex optimizations.

The realization-based methods by themselves require a finite number of linear algebraic operations. Because these methods implicitly optimize cost functions that are linear in state-space parameters, they may be augmented with convex constraints to form convex optimization problems. Several common behavioral constraints are translated into eigenvalue constraints stated as linear matrix inequalities, and the realization-based methods are converted into semidefinite programming problems. Some additional constraints on transient and steady-state behavior are derived and incorporated into a quadratic program, which is solved following the semidefinite program.

The newly developed realization-based methods are applied to two experiments: the aeroelastic response of a fighter aircraft and the transient thermal behavior of a light-emitting diode. The algorithms for each experiment are implemented in two freely available software packages.

1 Introduction

This first chapter introduces realization theory and its relationship to the identification problem as a historical narrative, beginning with Kronecker's work on the construction of rational functions from infinite series, proceeding to Ho and Kalman's reinterpretation of the theorem in a state-space linear systems framework, and concluding with Kung's realization algorithm based on the singular-value decomposition of a system Hankel matrix.

It is assumed that the reader is somewhat familiar with linear system realization theory in places. Chapter 2 provides a contemporary introduction to the subject. Readers uninterested in the historical development of the theory may skip ahead to the preliminary definitions in Section 1.2; the rest of the dissertation is self-contained.

1.1 Kronecker and Kalman: A Brief History of Realization Theory

With the creation of analysis in the mid 1800s, mathematics progressed beyond the traditional methods of algebraic construction that had defined the field for centuries. Infinite series were gradually replacing polynomials of finite order as the center of mathematical thought. The rigorous study of the convergence of series could be used to prove the existence of transcendental numbers, such as e and π, which were impossible to define as the roots of polynomials. At the same time, the new discipline of set theory was changing mathematics from the study of numbers into the study of abstract objects.

As with any change, there were those who resisted. The idea of infinity was something, literally, unnatural. No man personified the old school of constructionist mathematics better than Leopold Kronecker (1823 - 1891). Kronecker detested this new math [43]. To him, a number existed if and only if it could be constructed in a *finite* sequence of operations. Demonstrating that an *infinite* series converged to something did not prove that something existed because, in his view, completing the infinite was a paradox. Kronecker was also notoriously difficult to work with [43]. This was likely due to his dogmatic mathematical philosophy, illustrated by his famous quote regarding the divine origin of integers:

> God created the integers, all else is the work of man.

Less often quoted is Karl Weierstrass's response [16]:

> Kronecker puts his authority behind the opinion that everything that has been done up to now on the foundations of function theory are sins before the Lord ... Truly, it is sad and fills me with bitter pain ...

Kronecker was hardly alone in his opinions regarding the inferiority of analysis. Dirichlet also believed that all of mathematics could somehow be derived from the natural numbers, and Gauss argued that completed infinite series be excluded from mathematics entirely [18]. However Kronecker was perhaps the most vocal of these critics. The following quote from Kronecker, as retold by Weierstrass, suggests that Kronecker was hardly shy with his convictions [16]:

> If enough years and strength remain to me, I will myself show the mathematical world that arithmetic can lead the way not only for geometry but also for analysis, and surely the more rigorous way. If I can no longer do it, then those who come after me will ... and they will also recognize the falsity of all the conclusions with which the so-called analysis now works.

Kronecker was a man of his word, and much of his later work is on the construction of the finite from the infinite. Though he died long before the publication of Voltera's work on convolution integrals and likely never heard the term "impulse response," in his attempts to reformulate the foundation of analysis, he instead formulated the foundation of linear system realization theory.

Constructing the Finite from the Infinite

Kronecker was a pioneer of the use of determinants in linear algebra, and he was able to prove many interesting theorems from the determinants of structured matrices. In 1881, he published "Zur Theorie der Elimination Einer Variablen aus Zwei Algebraischen Gleichungen" (The Theory of Elimination of Variables from Two Algebraic Equations), a treatise on the elimination of common roots in polynomials. The only well known theorem from this work states that the terms of an infinite series are linearly dependent on each other if and only if the series is the expansion of a rational function. Kronecker proposed testing for linear dependence by examining the determinants of a structured matrix constructed from the terms of the series. This can also be used to test the coprimeness of polynomials.

As determinants have faded from the mathematical spotlight, due in large part to their numerical difficulties, so have many of Kronecker's theorems faded from textbooks.[1] A full proof of Kronecker's theorem appears in Appendix A. An important quality of the proof is that it is done *by construction*. Given a series with linearly dependent elements, formulas for computing the coefficients of the numerator and denominator polynomials of a rational function in a finite number of steps are provided in the proof itself. Kronecker considered this to be the only true way of proving existence. He did not reject all other styles of proof outright, he simply viewed them as being incomplete [16]. This does not mean he considered irrational numbers to be "fake." He argued that they instead be defined in an algebraic fashion; for instance $\sqrt{2}$ is defined as a root of the polynomial $x^2 - 2$ and not by somehow "completing the infinite," which to him was a contradiction. Being able to compute $\sqrt{2}$ to an arbitrary precision was very useful, but did not constitute a definition.

It is tempting to dismiss Kronecker's philosophy of finite constructionism entirely, given its apparent mathematical deficiencies. After all, e is an important number and needs a proper definition. But abstract existence is very different

[1]The only place the author was able to find even part of Kronecker's results from this paper in English was in [20], which is itself a translation of a Russian text. For those fluent in 19th-century German mathematics, Kronecker's original 79-page "Werk" is available online from the University of Michigan Historical Math Collection.

from physical existence. No computer calculates e with an infinite number of operations, and most engineers do not design machines that cannot be built. To the practicing engineer, Kronecker's philosophy has an intuitive appeal. Though modern mathematicians may complete the infinite without a second thought, engineers build things, and building things ultimately requires a constructionist philosophy not unlike Kronecker's. Engineers must agree with at least some of Kronecker's philosophy and, as he once wrote in a letter to Cantor, "take haven in the refuge of actual mathematics." [18] We do not have the luxury of completing the infinite whenever we choose.

The Rise of State-Space System Theory

Kronecker argued that set theory was essentially an unnecessary development, and that polynomials and the algebraic framework they inspire should be the foundation of mathematical thought [17]. Of course by the early 20th century, most mathematicians had rejected this idea, and today it seems almost heretical. However, engineers working in control systems and filter design through the 1950s continued to rely on the same polynomial methods for the analysis of differential equations developed over a century earlier. Before the arrival of modern computers, the most effective means of analyzing linear dynamic systems was through the Laplace transform. Ordinary differential equations could be expressed in terms of an auxiliary variable s and treated as fractions of polynomials, or transfer functions. This had the distinct advantage of allowing the behavior of dynamic systems to be analyzed by hand calculations. The restriction of transfer functions to input-output behavior also allowed for simpler experimental validation of dynamic models and readily extended to the frequency domain, which was often the primary domain of concern for engineers working with early electronics.

As time and technology progressed, however, the advantages of transfer-function methods became limitations. Transfer functions do not allow for the analysis of transient system behavior due to arbitrary initial conditions, the order of fractional polynomials that can be analyzed by hand is significantly restricted, and multivariable systems analysis is extremely cumbersome. Furthermore, the

restriction to input-output descriptions hides internal system dynamics, and many results before 1960 concerning the stability of composite systems were either misleading or incorrect [21].

State-space methods of system analysis were developed to address these limitations. By representing linear systems as systems of first-order differential equations, digital computers were able to provide optimal solutions to the problems of multivariable estimation and control. Many of the engineering problems of the day were in the areas of electronics and aerospace engineering and were naturally described in a state-space framework. State-space representations are also inherently causal while transfer functions may not be. But perhaps most importantly, by providing a more complete description of a system's internal dynamics, researchers could begin to derive necessary and sufficient conditions for system stability that did not suffer from the difficulties of pole-zero cancelations one encountered with transfer functions.

Controllability, Observability, and the Emergence of the Realization Problem

By 1960, the problem of optimal regulation based on quadratic criteria for linear systems had been proposed and solved in a number of special cases. Unfortunately, the lack of generality in the existing solutions meant that the problem of determining when solutions were possible was still an open question. Controllability and observability had yet to be rigorously defined and were often inadvertently assumed satisfied in regulation and estimation problems. While researchers were aware that open- and closed-loop control would fail in certain circumstances, tests that were both sufficient and necessary to guarantee the existence of a stabilizing controller/observer for a dynamic system had yet to be discovered.

It should surprise no one familiar with the field of control systems that it was Rudolf Kalman who, in his one of many breakthrough papers "On the General Theory of Control Systems [36]," presented the first conclusive tests for linear controllability and observability, their implications for quadratic regulation and estimation, and their relationship to classical Wiener filter design. Kalman's

results on controllability, observability, canonical forms, and optimal estimation and control would create the foundation for modern linear systems theory and become one of the most celebrated cannons in 20th-century research.

When Kalman published his first results in Kalman [36], the momentum of control systems research was at the time — and perhaps always has been — directed towards the pursuit of a "pure" theory of control akin to Shannon's theory of information. Kalman, however, was ultimately interested in application, and argued that the creation of such a theory was impossible until questions of "physical realizability" were answered. Though no one would suggest he shared Kronecker's devotion to constructionist algebra, Kalman conjures the spirit of Kronecker in this early paper when declares

> ... only *constructive*[2] methods are employed here, giving some hope of being able to avoid the well-known difficulty of Shannon's theory: methods of proof which are impractical for actually constructing practical solutions.

Kalman's constructionism, however, was not motivated by mathematical dogma, but by an awareness of the needs of practicing engineers. Kalman imagined what theory would be needed next to aid the progress of *existing* technology. If the design of dynamic systems was rooted in causal, linear, Newtonian physics, then these systems ought to be the focus of control systems research. Published with this early paper are two critiques: one arguing that his theory does not generalize sufficiently to the nonlinear case to be significant, and one arguing that his theory should be expanded to address what is now called identifiability. Kalman is sympathetic only to the latter in his response.

The familiar controllability and observability Grammians are absent in this first paper, though rank tests for time-invariant systems are implied. It was later that year, when Kalman published "Contributions to the Theory of Optimal Control [33]," that the controllability matrix was defined; it was shown that the rank of this matrix must be equal to the order of the system if there exists a control signal that can drive each state to 0 in an arbitrary finite time. This paper also

[2] Emphasis appears in the original.

introduces the Riccati differential equation as the means for solving the linear-quadratic-regulator problem.

Also that year, the prolific Kalman published perhaps his most famous work, "A New Approach to Linear Filtering and Prediction Problems [31]," which presented the first formulation of the now ubiquitous Kalman Filter. Kalman described the filter as a more practical approach to solving the Wiener filtering problem, which is the problem of optimally detecting a random stationary signal. While existing algorithms for the design of Wiener filters specified the optimal filter in terms of its impulse response, Kalman's filter was specified in terms of state-space equations which could be directly implemented on a digital computer.

In "Canonical Structure of Linear Dynamical Systems [32]," Kalman presented his landmark results on canonical forms. Though mostly remembered for proving that there exists a state basis for which the controllable/uncontrollable and observable/unobservable components of a linear system may be separated, included in this paper are the following observations: (i) at most, only the controllable and observable parts of a linear system may be identified from known input and output, and (ii) the impulse response function of a linear system may be separated into the product of a mapping from the state to the output and a mapping from the input to the state.

Over the next decade, Kalman and many others would repeat the sentiment that control design be studied in a linear state-space framework, since, unlike the differential operators of the Laplace domain, a state-space model was guaranteed to be causal and not hide potential instability, while transfer functions had no such guarantees. In [35], Kalman argued that

> the difficulty is due to insufficient appreciation of the concept of a dynamical system. Control theory is supposed to deal with physical systems, and not merely with mathematical objects such as a differential equation or a transfer function. We must therefore pay careful attention to the relationship between physical systems and their representation via differential equations, transfer functions, etc.

Kalman and others did not entertain the notion, however, that new control design methods would be adopted by practicing engineers on theoretical merit

alone. Elmer Gilbert summarized the situation in [21]:

> Since differential equations offer a safer basis for describing multivariable systems it is valid to ask why transfer-function matrices should be used at all. The answer is that frequency domain design procedures and the smaller size of [transfer-function martrices] often make computations more manageable.

A divide was quickly forming between control systems theory and control systems practice, and tools to address the differences between transfer-function and state-space descriptions of system were needed if state-space methods were to see widespread adoption.

Notable early attempts to solve the problem of moving between state-space and transfer-function representations include Gilbert [21] and Kalman [35]. In [35], Kalman defined the following problem:

> Given an (experimentally observed) impulse response matrix,[3] how can we identify the linear dynamical system which generated it? ... We propose to call any such system a *realization* of the given impulse response.

Kalman's use of the word "realization" was likely meant to connect these new ideas with the "realizability theory" of McMillan [47], whose extensive research the factorization of transfer function matrices in the previous decade laid the foundation for the field today known as network synthesis. Over the next few years, Kalman would modify his definition several times to more closely mirror McMillan's work.

Solving the Realization Problem

Early results on merging the input-output and state-space descriptions of systems focused primarily on the recently developed concept of canonical forms. In [35], Kalman proved that transfer functions and impulse responses only completely describe *minimal*[4] systems — systems that are both controllable and observable, which was central to the problem of moving between the two representations. Formulas for converting a minimal state-space system to a matrix of transfer functions

[3] By "impulse response matrix", Kalman was referring to convolution matrix as a function of time, not a Hankel matrix of impulse-response coefficients.

[4] He actually used the word "irreducible," but the term "minimal" has since become more common in this context.

first appear in Gilbert [21], which shows that cancellation of unstable controllable modes in cascade systems does not stabilize a system, something that is unclear in the transfer function representation. Both Gilbert [21] and Kalman [35] provide methods for the construction of state-space representations: Gilbert's requires awkward partial-fraction expansion of matrix polynomials, and Kalman's requires an equally awkward factorization of polynomial matrices.

The previous decade, McMillan had shown that the minimum number of energy storage elements necessary to construct a circuit from a given transfer function matrix was equal to the degree of the least common denominator polynomial of the transfer function matrix [47]. This idea was clearly somehow linked to "minimal" dimension of state-space systems and was conjectured by Gilbert in [21], but it was not until [34] that Kalman proved the two numbers were in fact equal. This paper also contains a rather complex method for constructing minimal representations using partial fraction expansions to fill in the entries of a state-space system in Jordan canonical form.

Interestingly, in [35], in which he first defines the realization problem, Kalman cites Volume 1 of a two-volume translation of Gantmacher's *Theory of Matrices*[5] [19]. Kronecker's theorem for the reduction rational functions via Hankel matrices appears in Volume 2 [20]. It was left to Kalman's graduate student, B.Y. Ho,[6] to finish reading the series and discover that Kronecker had solved the realization problem 70 years earlier. The following year, Ho and Kalman presented their eponymous algorithm for the construction of state-space systems from Hankel matrices of impulse response coefficients [27]. The proof of the method can

[5] Gantmacher published his original Russian text in 1953. Two different translations of the text were released by two separate publishers in the same year in 1959. Interscience released a one-volume edition, while Chelsea released a two-volume edition. Though larger and more expensive, the Chelsea version was far more popular, likely due to the translator's correction of a number of mistakes in the original work, while the Interscience version did not survive its first run (reviews of both appear in [46]).

Gantmacher [20] may be the first, if not only, appearance of Kronecker's theorem in English outside loosely translated lecture notes. References to "Kronecker's theorem" or even "Kronecker's well-known theorem" are commonly made without citation in the literature. After thorough research, [20] remains the only publication other than the original German that the author has been able to find an actual proof. This compelled him to include it in a soon-to-appear textbook (B1 in Publications).

[6] Not to be confused with Yu-Chi "Larry" Ho, another colleague of Kalman's.

be found in Chapter 2 and is effectively a restatement of Kronecker's theorem in state-space form. This also appears to be the first instance of impulse response coefficients being referred to as "Markov parameters," which mirrors the usage of the word in Gantmacher [20]. The introduction to [27] states "this is probably the simplest method for computing a realization that will ever be found." They were almost right.

Hankel-Matrix Methods

The Ho-Kalman algorithm in its original form [27] bares only partial resemblance to what appears in textbooks today. Though it could be considered constructive, it provides no explicit method to factor the Hankel matrix of Markov parameters. That the system Hankel matrix is the product of the observability matrix and controllability matrix appears to have been entirely overlooked. Instead, block-companion matrices are used to demonstrate what is now called the "shift-invariance" of the matrix. Shortly later, in research apparently developed independently of Ho and Kalman [27], Youla and Tissi [84] showed that any appropriately dimensioned factorization of the Hankel matrix results in a controllability and observability matrix for some state basis, though it was in the context of network synthesis problems and did not generalize well to arbitrary linear systems. Silverman [64] extended some of the ideas in [27] to time-varying systems. Silverman [63] gives a complete restatement of Kronecker's theorem in a state-space framework, which is sometimes referred to as "Silverman's Algorithm." Chen and Mital [11] later reformulate the Ho-Kalman algorithm to be more computationally friendly.

The almost trivial shift-invariant property of the Hankel matrix for discrete-time systems most commonly seen today — and seen in (2.4) of the following chapter — seems to have gone without discovery, or at least explicit statement, for several years after Ho's initial publication. This is likely due to the focus on continuous-time systems by researchers of the era. Although digital computer were become more common in control applications, most controllers were still constructed from linear circuits. Ho's original formulation is generic enough to

apply to both continuous- and discrete-time systems. In the preceding works cited, only Silverman [63] addresses the discrete-time case specifically. It is not until Ackermann and Bucy [1], which modifies Ho's method to produce a realization in "Bucy canonical form[7]," that the discrete time shift-invariant property appears explicitly.

The first appearance of what today is most often referred to as either the Ho-Kalman algorithm or Kung's algorithm (discussed later) actually appears in a 1.5-column note in the "Technical Notes and Correspondence" section of a 1973 issue of the IEEE Transactions on Automatic Control. In Zeiger and McEwen [85], it is suggested that the Ho-Kalman algorithm be solved with the singular-value decomposition (SVD) from noisy discrete-time Markov parameters. This is the first appearance of the SVD in this context, and it can possibly be considered the first realization-based or subspace identification method, as well as the first statement of what is now called the "Hankel norm" of a linear system.

Though Zeiger and McEwen [85] did first propose using the SVD, a thorough analysis of the method did not appear until 1978 in Kung [38][8]. In this paper, Kung derives an algorithm that that is the common ancestor of subspace identification, the Eigensystem Realization Algorithm, and Hankel-opperator theoretic approaches to model reduction. Kung's work after [38] has focused almost entirely on model-reduction, but the contribution of this paper to the field of system identification should not be understated.

A history of the development of realization-based and subspace-based system identification methods would require another chapter. The historical development of these methods is better documented than that of realization theory and can be found in several books on the subject, such as Juang [29], Van Overschee

[7] If this term is unfamiliar, it may be because it seems to only appear in publications that list Bucy as an author.

[8] This is possibly the most cited yet least read paper in the field of control systems. As of writing, Google Scholar lists it as having 374 citations, however it only appears in the proceedings of an IEEE conference that IEEE has no record of, no digital copies exist, and according to WorldCat, only 5 libraries in the world have copies of the proceedings: two in the US, two in Canada, and one in Germany. There exist papers that present "new" results entirely contained in [38] and yet also cite it as a reference. The author is indebted to the librarians of North Carolina State University for scanning and emailing him a copy.

and De Moor [76], and Verhaegen and Verdult [78]. Instead, we end with Kung. The method Kung proposed is derived in the next chapter, and in each subsequent chapter, we will generalize it to alternative forms of measured data beyond the system impulse response.

1.2 Preliminaries

This section presents definitions and notations used throughout the rest of the dissertation.

1.2.1 Notation

The set of all real numbers will be denoted by $\mathbb{R}$ and all complex numbers by $\mathbb{C}$. An n-dimensional vector x of real numbers will be defined as $x \in \mathbb{R}^n$, and an m-dimensional vector y of complex numbers as $y \in \mathbb{C}^m$. The k-th element of a vector x will be denoted x_k. Matrices of real numbers with n rows and m columns will be defined as $X \in \mathbb{R}^{n \times m}$ and of complex numbers as $Y \in \mathbb{C}^{n \times m}$.

The submatrix of a matrix X taken from the i-th row to the k-th row and the m-th column to the n-th column will be denoted using MATLAB-style subscripts as $M_{(i:k,m:n)}$. The matrix I_k is the identity matrix with k rows and columns. An n-by-m matrix of entirely zeros will be denoted $0_{n \times m}$ and an n-by-m matrix of entirely ones will be denoted $1_{n \times m}$.

Signals will always be discrete-time and indexed with an integer index t unless otherwise stated. The value of a signal x at time t will be denoted $x(t)$. An n-dimensional signal will be defined as $x(t) \in \mathbb{R}^n$. Some signals will be matrix valued, such as covariance function; an n-by-m-signal will be defined as $X(t) \in \mathbb{R}^{n \times m}$.

1.2.2 Linear, Time-Invariant Systems

The identification methods presented herein are restricted to discrete-time, linear, time-invariant (LTI) systems. A discrete-time LTI system has several equivalent representations. The most common are fractional representations and state-

space representations. By defining a forward-time-shift operator q so that

$$qu(t) = u(t+1),$$

the relationship between the input $u(t) \in \mathbb{R}$ and the output $y(t) \in \mathbb{R}$ may be defined in terms of a rational function of q as

$$y(t) = G(q)u(t),$$

where

$$G(q) = \frac{b(q)}{a(q)} = \frac{b_m q^m + b_{m-1}q^{m-1} + \cdots + b_1 q + b_0}{q^n + a_{n-1}q^{n-1} + \cdots + a_1 q + a_0}. \tag{1.1}$$

$G(q)$ must be *proper* , that is, $m \leq n$, to ensure causality. If $G(q)$ is *strictly proper*, that is, $m < n$, then the system has no feed-through term. If $a(q)$ has all roots inside the unit circle, the system is *stable*, and a bounded $u(t)$ results in a bounded $y(t)$. If $b(q)$ and $a(q)$ have no common roots, then $G(q)$ is *coprime*. Fractional representations are not limited to single-input-single-output systems. Multi-input-multi-output systems with vector-valued $y(t)$ and $u(t)$ can be defined as a matrix of fractional operators.

The other common representation for LTI systems is the state-space representation in which a set of matrices (A, B, C, D) defines the input-output relationship

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1.2}$$

where $u(t) \in \mathbb{R}^{n_u}$ is the input signal, $y(t) \in \mathbb{R}^{n_y}$ is the output signal, and $x(t) \in \mathbb{R}^n$ is the system state. The system is stable if A has all eigenvalues inside the unit circle.

The system (1.2) is *controllable* if the state $x(t)$ may be achieve an arbitrary value with a proper selection of $u(t)$ in n time steps. This is true if and only if the *controllability matrix*

$$\mathcal{C}_n = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

has rank n. The system is *observable* if the initial state $x(0)$ may be determined from n observations of the output. This is true if and only if the *observability*

matrix

$$\mathcal{O}_n = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has rank n. Systems that are both controllable and observable are *minimal* and have the property that the state dimension n cannot be reduced [10]. This is closely related to the idea of coprimeness for the fractional operator $G(q)$. The state-space representation is guaranteed to be causal, and thus proper, and so it is called a *realizable* representation of a system since it can always be constructed in reality. It is strictly proper if $D = 0$. In this dissertation, all systems will be assumed minimal unless otherwise stated.

A discrete-time LTI system also has a representation of a convolution with an infinite series

$$y(t) = \sum_{k=0}^{\infty} G(k)u(t-k) + v(t) \tag{1.3}$$

where $G(k) \in \mathbb{R}^{n_y \times n_u}$ are the system Markov parameters. To avoid confusion with the representation (1.1), $G(\cdot)$ should be assumed to be indexing the Markov parameter sequence if any argument other than q is given. Because the system description (1.3) cannot be represented with a finite number of parameters, it is referred to as a *non-parametric* description. This representation is also guaranteed to be causal and thus proper since the summation starts at 0. It is strictly proper if $G(0) = 0$. There are no tests for stability that explicitly use the Markov parameters that do not require knowledge of some sort regarding the entire sequence.

Conversion between the fractional representation (1.1) and the state-space representation (1.2) is straightforward by means of canonical forms [10]. Less clear is the conversion from the Markov parameter sequence. It is easily found that the Markov parameters may be constructed from the state-space parameters as

$$G(k) = \begin{cases} D & k = 0, \\ CA^{k-1}B & k > 0. \end{cases} \tag{1.4}$$

The inverse problem of constructing a state-space representation from a sequence of Markov parameters is more complicated. This is the problem of *realization*.

Before addressing the realization problem more thoroughly, however, we include some definitions concerning stochastic processes.

1.2.3 Stochastic Processes and Quasi-Stationary Signals

A stochastic process is a time series of random variables with an underlying probability distribution. Related to the time-invariance of deterministic processes is the idea of stationarity for stochastic processes. Because system identification concerns both stochastic and deterministic signals, the traditional definition of stationarity must be modified to apply to both.

A signal $s(t) \in \mathbb{R}^{n_s}$, $t \in \mathbb{Z}$ is said to be *quasi-stationary* if it is subject to the two conditions

$$Es(t) = m_s(t), \qquad ||m_s(t)||_2 \leq C \ \forall \ t \in \mathbb{Z}$$

and

$$R_s(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-\tau-1} Es(t+\tau)s(t)^T, \quad \forall \tau \in \mathbb{Z}, \quad ||R_s(\tau)||_2 \leq C,$$

for some $C < \infty$, where E denotes expectation, which is defined to have no effect if $s(t)$ is strictly deterministic. The function $R_s(\tau) : \mathbb{Z} \to \mathbb{R}^{n_s \times n_s}$ is called the autocovariance function of $s(t)$. Similarly, if $w(t) \in \mathbb{R}^{n_w}$ is a second quasi-stationary signal, then the function $R_{sw}(\tau) : \mathbb{Z} \to \mathbb{R}^{n_s \times n_w}$,

$$R_{sw}(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-\tau-1} Es(t+\tau)w(t)^T$$

is called the cross-covariance function of $s(t)$ and $w(t)$. It is zero everywhere if $s(t)$ and $w(t)$ are orthogonal or statistically independent. If only N samples of data are available, estimates of the autocovariance and cross-covariance functions may be calculated as

$$\hat{R}_s(\tau) = \frac{1}{N} \sum_{t=0}^{N-\tau-1} s(t+\tau)s(t)^T$$
$$\hat{R}_{sw}(\tau) = \frac{1}{N} \sum_{t=0}^{N-\tau-1} s(t+\tau)w(t)^T,$$

and are assumed to converge to $R_s(\tau)$ and $R_{sw}(\tau)$, respectively, as $N \to \infty$. These definitions may be applied to both stochastic and deterministic processes. The preceding is adopted from [40] and extended to multivariable signals.

Consider again the state-space representation (1.2), and assume A is stable. If $u(t)$ is quasi-stationary, then $y(t)$ will also be quasi-stationary [40]. If some quasi-stationary signal $\xi(t) \in \mathbb{R}^{n_\xi}$ is correlated with the input, then the cross-covariance signal $R_{y\xi}(\tau) \in \mathbb{R}^{n_y \times n_\xi}$ is the cross-covariance signal $R_{u\xi}(\tau) \in \mathbb{R}^{n_u \times n_\xi}$ filtered through the system dynamics. Thus the covariance functions may be expressed in terms of the state-space matrices (A, B, C, D) as

$$\begin{aligned} R_{x\xi}(\tau+1) &= AR_{x\xi}(\tau) + BR_{u\xi}(\tau) \\ R_{y\xi}(\tau) &= CR_{x\xi}(\tau) + DR_{u\xi}(\tau). \end{aligned} \tag{1.5}$$

In this dissertation, it will always be assumed that signals are quasi-stationary and zero-mean unless stated otherwise. Note that there is no distinction between covariance functions and correlation functions for zero-mean signals.

1.3 Acknowledgements

The historical narrative of this chapter is original. The formulation of realization theory in the context of rational functions was presented in

D.N. Miller and R.A. de Callafon, "Identification of Linear Discrete-Time Filters via Realization," to appear in *Linear Algebra* (working title). H. Yasser, Ed. InTech: 2012.

2 Realization from an Impulse Response

For a linear system, a sequence of output data measured over some finite period of time may be expressed as the linear combination of the past input and the input measured over that same period. For a finite-dimensional system, the mapping from past input to future output is a finite-rank linear operator, and the effect of the past input may be stored in a finite-dimensional vector defined as the system state.

The central idea of realization theory is to factor this mapping from past input to future output into two parts: a map from the input to the state, and another from the state to the output. This factorization provides a complete description of the system dynamics and guarantees the representation is both causal and finite-dimensional; thus it can be physically constructed, or *realized*.

Realization-based identification refers to system identification methods that construct system models by identifying the mapping from past input to future output and constructing a state-space representation via a rank-reducing factorization. The non-deterministic nature of identification requires that both these steps be carefully considered to guarantee consistent estimates.

This chapter presents the central theory behind realization-based system identification. The unifying quality of the various methods presented in this dissertation is that the state dynamics of a discrete-time LTI system may by found by examining single time-shifts in sequences of data. Some definitions from the previous chapter may be repeated for convenience and clarity.

2.1 Properties of the System Hankel Matrix

The realization problem begins with the construction of structured data matrices. We first examine data matrices that define the relationship between infinite-dimensional vectors of input and output data. These vectors are then concatenated to form finite-dimensional equations containing state-space parameters.

Suppose for now that the measured input and output data from a noise-free system with minimal state dimension is known perfectly. From the Markov-parameter convolution

$$y(t) = \sum_{k=0}^{\infty} G(k)u(t-k), \tag{2.1}$$

an infinite-dimensional column vector of all output data may be expressed as the product of a block-Toeplitz convolution matrix and an infinite-dimensional column vector of all input data,

$$\left[\begin{array}{c} \vdots \\ y(t-3) \\ y(t-2) \\ y(t-1) \\ \hline y(t) \\ y(t+1) \\ y(t+2) \\ \vdots \end{array}\right] = \left[\begin{array}{cccc|cccc} \ddots & & & & & & \cdots & 0 \\ \cdots & G(0) & & & & & & \vdots \\ \cdots & G(1) & G(0) & & & & & \\ \cdots & G(2) & G(1) & G(0) & & & & \\ \hline \cdots & G(3) & G(2) & G(1) & G(0) & & & \\ \cdots & G(4) & G(3) & G(2) & G(1) & G(0) & & \\ \cdots & G(5) & G(4) & G(3) & G(2) & G(1) & G(0) & \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right] \left[\begin{array}{c} \vdots \\ u(t-3) \\ u(t-2) \\ u(t-1) \\ \hline u(t) \\ u(t+1) \\ u(t+2) \\ \vdots \end{array}\right],$$

where the matrices are partitioned so that the past, that is, everything before the current time t, has been separated from the present and future. The present and future output, that is, $y(t)$ for $t \geq 0$, may be separated into past and future components as

$$y_f = \begin{bmatrix} G(1) & G(2) & G(3) & \cdots \\ G(2) & G(3) & G(4) & \cdots \\ G(3) & G(4) & G(5) & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} u_p + \begin{bmatrix} G(0) & & & \\ G(1) & G(0) & & \\ G(2) & G(1) & G(0) & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} u_f,$$

where

$$y_f = \begin{bmatrix} y(t) \\ y(t+1) \\ y(t+2) \\ \vdots \end{bmatrix}, \qquad u_p = \begin{bmatrix} u(t-1) \\ u(t-2) \\ u(t-3) \\ \vdots \end{bmatrix}, \qquad \text{and} \qquad u_f = \begin{bmatrix} u(t) \\ u(t+1) \\ u(t+2) \\ \vdots \end{bmatrix}.$$

The matrix

$$H = \begin{bmatrix} G(1) & G(2) & G(3) & \cdots \\ G(2) & G(3) & G(4) & \cdots \\ G(3) & G(4) & G(5) & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{2.2}$$

is the infinite-dimensional system Hankel matrix. It maps past input to future output, and in some respects its 2-norm $||H||_2$ is well-suited for defining the "gain" of an LTI system. (See, for instance, Zhou, Doyle, and Glover [86]). Substitution of the state-space parameters from the relationship

$$G(k) = \begin{cases} D & k = 0 \\ CA^{k-1}B & k > 0 \end{cases}$$

reveals that (2.2) is the product of the infinite-dimensional extended observability matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix}$$

and the infinite-dimensional extended controllability matrix [1]

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots \end{bmatrix}.$$

[1] Often Γ and Ω are used in the literature to denote the extended observability and extended controllability matrices, respectively. This appears to date back to Rissanen [61] and related papers that reinterpret Kalman's results in an algebraic framework. In these papers, Ω is the set of all homomorphisms mapping the nonpositive integers to the input vector space (the past and present input) and Γ is the set of all homomorphisms mapping all positive integers to the output vector space (the future output). Thus Ω defines the behavior of the past input and Γ defines the behavior of the future output. The linear space $\Omega \oplus \Gamma$ is then a ring with convolution as the product and the unit pulse as the identity.

so that

$$H = \mathcal{O}\mathcal{C}.$$

Because $\text{rank}(\mathcal{O}) = \text{rank}(\mathcal{C}) = n$ for a minimal system,

$$\text{rank}(H) = n.$$

Additionally, if the indices of H are shifted forward by 1 to form an infinite-dimensional Hankel matrix

$$H' = \begin{bmatrix} G(2) & G(3) & G(4) & \cdots \\ G(3) & G(4) & G(5) & \cdots \\ G(4) & G(5) & G(6) & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}, \tag{2.3}$$

then substitution of (1.4) reveals

$$H' = \mathcal{O}A\mathcal{C}. \tag{2.4}$$

Generally, we will only consider a finite "slice" of H, denoted

$$H_k = \begin{bmatrix} G(1) & G(2) & \cdots & G(l) \\ G(2) & G(3) & \cdots & G(l+1) \\ \vdots & \vdots & & \vdots \\ G(k) & G(k+1) & \cdots & G(k+l-1) \end{bmatrix} \tag{2.5}$$

which is the product of

$$\mathcal{O}_k = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{k-1} \end{bmatrix} \qquad \text{and} \qquad \mathcal{C}_l = \begin{bmatrix} B & AB & \cdots & A^{l-1}B \end{bmatrix}.$$

A finite-dimensional H'_k is defined similarly:

$$H'_k = \begin{bmatrix} G(2) & G(3) & \cdots & G(l+1) \\ G(3) & G(4) & \cdots & G(l+2) \\ \vdots & \vdots & & \vdots \\ G(k+1) & G(k+2) & \cdots & G(k+l) \end{bmatrix} \tag{2.6}$$

The index l is omitted from the subscripts of H_k and H'_k because the column dimension is less significant in the later identification procedures.

If H_k is known exactly, then *any* factorization

$$H_k = \mathcal{O}_k \mathcal{C}_l$$

with valid dimensions will result in an $\mathcal{O}_k$ and $\mathcal{C}_l$ for some arbitrary state basis. If H'_k is also known exactly, the parameter A in the same basis as $\mathcal{O}_k$ and $\mathcal{C}_l$ may be found from

$$A = \mathcal{O}_k^\dagger H'_k \mathcal{C}_l^\dagger, \tag{2.7}$$

where $(\cdot)^\dagger$ is the Moore-Penrose pseudoinverse. Then with C taken from the top n_y rows of $\mathcal{O}_k$, B taken from the first n_u columns of C_l, and $D = G(0)$, a complete and minimal state-space realization may be found from a deterministic sequence of Markov parameters. Only $2n+1$ Markov parameters are needed to realize a state-space system: the first parameter $G(0)$ for D, the next $2n$ for H, and $G(2n+1)$ for H'. The state-basis of the resulting realization will of course depend on the factorization used to find $\mathcal{O}_k$ and $\mathcal{C}_l$.

At times only an estimate of the extended observability matrix $\mathcal{O}_k$ is available. In these cases, A can be estimated from the shift-invariance of $\mathcal{O}_k$ alone as follows: Let $\mathcal{O}_{2|k}$ denote block rows 2 through k of the infinite-dimensional extended observability matrix, so that

$$\mathcal{O}_{2|k} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{k-1} \end{bmatrix}.$$

Because

$$\mathcal{O}_{2|k} A = \mathcal{O}_{k-1},$$

A may be found from

$$A = \mathcal{O}_{2|k}^\dagger \mathcal{O}_{k-1}. \tag{2.8}$$

This relationship is the basis of the Eigensystem Realization Algorithm (ERA) [30] and the Multivariable Output-Error State-sPace (MOESP) methods [78]. A similar

method may be used to solve for A by shifting the columns of the controllability matrix [38].

2.2 Estimation of the System Hankel Matrix

The earliest realization-based identification methods factor estimates of H_k directly where H_k is assumed to have rank n. Several ways to estimate H_k exist. The most obvious is to apply a unit impulse on $u(t)$ at time $t = 0$. Then $G(k) = y(k)$, and H_k may be constructed directly from measured output data. In practice, however, many systems cannot be sufficiently excited from an impulse without exciting undesired nonlinearities in the response, and even small amounts of noise may severely degrade the quality of the estimate. Another possibility is to deconvolve the output with the input, but this will amplify noise beyond usable levels in most cases.

Alternatively, the Markov parameter sequence is the inverse discrete Fourier transform of the frequency-response function of the system [30], so that given the Fourier transforms $\mathcal{F}[u(t)] = U(e^{i\omega})$ and $\mathcal{F}[y(t)] = Y(e^{i\omega})$,

$$G(k) = \mathcal{F}^{-1}\left[\frac{Y(e^{i\omega})}{U(e^{i\omega})}\right].$$

If the frequency response measurement is accurate enough, it may be used to generate estimates of the Markov parameters which may be used to construct H_k directly. A broadband input excitation signal, however, is required to estimate and invert the frequency response function with sufficient accuracy. These and additional difficulties associated with inverting the frequency response are discussed in the introduction of McKelvey, Akçay, and Ljung [44].

We have not yet addressed the effects of noise on the estimate of H_k or on the factorization $\mathcal{O}_k\mathcal{C}_l$. Let $\hat{H}_k$ be the estimate of H_k constructed from estimated Markov parameters. If $\hat{H}_k$ has an error term

$$\hat{H}_k = H_k + E,$$

where E is the result of a stochastic process, then $\hat{H}_k$ will have full rank instead of rank n if $k > n$, and a factorization into valid-dimensioned $\mathcal{O}_k$ and $\mathcal{C}_k$ will only

be possible if $n = k$. Generally either the order of the system is unknown, or we would like to use more than the first $2n + 1$ Markov parameters to estimate the system so that we may have an estimate $\hat{H}_k$ with 2-norm closer to H_k. This requires reducing the rank of $\hat{H}_k$.

2.3 Factorization of the Estimated Hankel Matrix

In this section we study the factorization of an estimated system Hankel matrix $\hat{H}_k$ with $k \geq n$, which has rank greater than the rank of the system n. It is clear that no exact rank-n factorization exists, and so we must first solve a rank-reduction problem prior to factorization.

2.3.1 Rank Reduction of the Estimate

The obvious tool for reducing the rank of matrices is the *singular-value decomposition* (SVD). Estimating the system parameters this way is known as Kung's method [38]. Assume for now that n is known. The SVD of $\hat{H}_k$ is

$$\hat{H}_k = U\Sigma V^T$$

where U and V^T are orthogonal matrices and Σ is a diagonal matrix containing the nonnegative *singular values* σ_i ordered from largest to smallest. The SVD for a matrix is unique and guaranteed to exist, and the number of nonzero singular values of a matrix is equal to its rank [22]. Because U and V^T are orthogonal, it also satisfies

$$\hat{H}_k = \left|\left|U\Sigma V^T\right|\right|_2 = ||\Sigma||_2 = \sigma_1 \tag{2.9}$$

where $||\cdot||_2$ is the induced matrix 2-norm, and

$$\hat{H}_k = \left|\left|U\Sigma V^T\right|\right|_F = ||\Sigma||_F = \left(\sum_i^l \sigma_i^2\right)^{1/2} \tag{2.10}$$

where $||\cdot||_F$ is the Frobenius norm. From (2.9) and (2.10), we can directly see that if the SVD of H_k is partitioned into

$$\hat{H}_k = \begin{bmatrix} U_n & U_s \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_s \end{bmatrix} \begin{bmatrix} V_n^T \\ V_s^T \end{bmatrix},$$

where U_n is the first n columns of U, Σ_n is the upper-left $n \times n$ block of Σ, and V_n^T is the first n rows of V^T, the solution to the rank-reduction problem is [22]

$$Q = \underset{\text{rank}(Q)=n}{\arg\min} \left|\left|Q - \hat{H}_k\right|\right|_2 = \underset{\text{rank}(Q)=n}{\arg\min} \left|\left|Q - \hat{H}_k\right|\right|_F = U_n \Sigma_n V_n^T.$$

Additionally,

$$\left|\left|Q - \hat{H}_k\right|\right|_2 = \sigma_{n+1},$$

which suggests that if the rank of H_k is not known beforehand, it can be determined from examining the nonzero singular values in the deterministic case and from searching for a significant drop-off in singular values if only a noise-corrupted estimate is available.

2.3.2 Choice of State Basis in Factorization

From the rank-n matrix Q, any factorization

$$\hat{H}_k = \hat{\mathcal{O}}_k \hat{\mathcal{C}}_l$$

can be used to estimate $\mathcal{O}_k$ and $\mathcal{C}_l$. The error in the state-space realization, however, will depend on the chosen state basis. Generally we would like to have a state variable with a norm $||x_k||_2$ in between $||u_k||_2$ and $||y_k||_2$. As first proposed in Zeiger and McEwen [85], choosing the factorization

$$\hat{\mathcal{O}}_k = U_n \Sigma_n^{1/2} \qquad \text{and} \qquad \hat{\mathcal{C}}_l = \Sigma_n^{1/2} V_n^T \tag{2.11}$$

results in

$$\left|\left|\hat{\mathcal{O}}_k\right|\right|_2 = \left|\left|\hat{\mathcal{C}}_l\right|\right|_2 = \sqrt{\left|\left|\hat{H}_k\right|\right|_2}, \tag{2.12}$$

and thus, from a functional perspective, the mappings from input to state and state to output will have equal magnitudes, and the scalar entries of the state

vector x_k will have similar magnitudes. State-space realizations that satisfy (2.12) are sometimes called *internally balanced* realizations [10]. (Alternative definitions of a "balanced" realization exist, however, and it is generally wise to verify the definition in each context.)

With $\hat{\mathcal{O}}_k$ and $\hat{\mathcal{C}}_l$ known, an estimate $\hat{A}$ may be calculated by finding an estimate of the shifted Hankel matrix H'_k. If $\hat{H}'_k$ is an estimate of H'_k, then choosing the factorization (2.11) simplifies (2.7) to

$$\begin{aligned}\hat{A} &= \left(\hat{\mathcal{O}}_k\right)^{\dagger} \hat{H}'_k \left(\hat{\mathcal{C}}_l\right)^{\dagger} \\ &= \Sigma_n^{-1/2} U_n^T \hat{H}'_k V_n \Sigma_n^{-1/2}.\end{aligned}$$

By estimating $\hat{B}$ as the first block column of $\hat{\mathcal{C}}_l$, $\hat{C}$ as the first block row of $\hat{\mathcal{O}}_k$, and $\hat{D}$ as $G(0)$, a complete state-space realization $(\hat{A},\ \hat{B},\ \hat{C},\ \hat{D})$ may be identified from estimates of the system Markov parameters.

2.4 Pitfalls of Direct Realization Methods

Realization-based identification methods that generate a system estimate from either a direct estimate of the system Hankel matrix or a Hankel matrix constructed of estimated Markov parameters have numerous difficulties when applied to noisy measurements. Measuring an impulse response directly is often infeasible; high-frequency damping may result in a measurement that has a very brief response before the signal-to-noise ratio becomes prohibitively small, and a unit pulse will often excite high-frequency nonlinearities that degrade the quality of the resulting estimate. "Pre-loading" the system — applying an arbitrary initial condition, usually by bending a mechanical structure — and measuring the free-response when released, can have similar difficulties.

Both these methods suffer from the fact that the input is not quasi-stationary, and the only way to ensure that an estimate converges to its true value is repeat experiments and average the results. This of course assumes that the experiments are repeatable, which is often impractical.

Taking the inverse Fourier transform of the frequency response guarantees that the estimates of the Markov parameters will converge as the dataset grows

only so long as the input is broadband. Generally input signals decay at higher frequencies, and calculation of the frequency response from the spectrum by inversion of the input will amplify high-frequency noise.

We would prefer an identification method that is guaranteed to provide a system estimate that converges to the true system as the amount of data measured increases. This can only occur if the input is quasi-stationary. Unfortunately, constraints on the input signal may make such experiments impossible, and so methods that identify high-quality models without quasi-stationary input are still quite valuable. In the following chapter, we present a realization-based identification procedure that constructs system estimates directly from step-response measurements. The realization-based approach will later be extended to arbitrary input-output data sequences.

2.5 Acknowledgements

This chapter, in part, includes content from the following publication:

D.N. Miller and R.A. de Callafon, "Subspace Identification from Classical Realization Methods," *Proc. of the 15th IFAC Symposium on System Identification*, Saint-Malo, France: July 2009.

3 Realization from a Step Response

Step-response measurements are perhaps the most common measurement used for identification and model validation purposes. The experiments are simple to conduct, and many common physical systems can be adequately approximated by first or second order systems derived from graphical measurements of a step response, so long as the measurement noise is low. When the system dynamics cannot be approximated by low-order systems, however, or when measurement noise is greater, common graphical methods often fail to identify system dynamics with a sufficient degree of accuracy. Additionally, graphical methods are difficult to automate and cannot be used to identify models with more than one output signal.

Despite the popularity of step responses experiments in practice, identification from step-response measurements is often overlooked in system identification literature. It is generally assumed that more complex models require more complex experiments in which convergence can be guaranteed as the length of the experiment increases. While this is often the case, there are many systems for which a step is the only feasible input. As an example, consider the cooking of food. Repeated heating and cooling of food will fundamentally change the nature of the response; once the food is cooked, then cooling will not un-cook it. A persistently exciting input is clearly impossible in this case, yet the transient response to a step-increase in temperature may still be adequately captured by a simple LTI model.

Though modeling the thermal response of food might not be the most mo-

tivational research problem, the difficulties in this example mirror many processes in thermodynamics and manufacturing. Additionally, preliminary step-response experiments conducted prior to a rigorous identification procedure are useful for experimental design purposes, since even in situations with poor signal-to-noise ratios, a step response is usually sufficient for determining the dominant fundamental frequencies of the system [24].

For some processes, the primary target of identification is not the input-output behavior, but the time constants that contribute to various lags in the step response, sometimes referred to as the *time-constant spectrum*. Such methods are often used for defect detection where the change in time-constants of a step-response implies a manufacturing error. Time-constant spectrum methods are mostly non-parametric. They are discussed further in Chapter 8.

This chapter presents a method for identifying models from a step response based on an extension of the impulse-response realization procedure presented in the previous chapter. It begins with an overview of existing methods. The step-based realization procedure is then derived for single-input systems. The chapter concludes by extending the method to multi-input systems with separate datasets in which an experiment has been performed for each input.

3.1 Some Background for Step-Based Identification Methods

Although most system identification methods identify discrete-time models, nearly all methods that are explicitly formulated to use step-response measurements produce continuous-time models. Several likely reasons for this exist. Practicing engineers are often more comfortable with continuous-time models, since continuous-time Newtonian physics is the foundation of most engineering disciplines. Additionally, step-responses are most often used to identify simple, low-order models for use with PID controllers, and most PID tuning rules require continuous-time models.

Discrete-time models constructed using unconstrained least-squares meth-

ods may have poles with negative real parts. Because the continuous-time poles depend on the logarithm of the discrete-time poles for a zero-order hold discretization, the discrete-time poles with negative real parts cannot be directly converted to continuous-time poles. Thus, if a continuous-time model is desired as the final result, direct continuous-time identification is often simpler than either constrained discrete-time methods or more sophisticated continuous-time conversions.

As stated, most step-based identification methods are graphical; they are typically based on either measuring tangent angles at various points in the step response curves or measuring the rise time as well as the period and decay rate of any oscillations. This limits the type of models that can be identified to first- or second-order systems with a possible time delay. Other methods identify continuous-time models by estimating the area of various regions in the step-response curve. Both point-based and area-based methods tend to be sensitive to noise, though area-based methods less so. Common graphical approaches may be found in the literature on PID control, notably Aström and Hägglund [4]. Rake [59] extends common graphical methods to inputs beyond step-response measurements, such as pulses or ramps.

Graphical methods typically require a complete response, which can be problematic if the settling time of the system is particularly long. Often, the steady-state value of a process to a step response is known beforehand, however, and this additional knowledge can be used to construct a model from a partial step response. Bi, Cai, Lee, Wang, Hang, and Zhang [7] develop a method for estimation of a first-order system with a time delay by numerically measuring the area between the curve of a step-response and its steady-state value which does not require the system to settle to steady-state.

Non-graphical methods for identification from step-response measurements, including non-parametric methods, are often based on numerically differentiating the step response to estimate the system impulse response. A model may then be identified from the impulse response using a method such as the realization-based procedure previously presented. Although methods based on numerically differentiating measured data are clearly limited in to cases in which noise levels

are very low, they are still popular in some areas such as defect detection [69].

Another reason for a lack of step-based identification procedures is that if the output of a linear system to a known input is solved for as a function of time, the parameters of the system do not appear linearly in the analytical solution. Fitting a parametric model of linear differential equations directly to a measured response will require nonlinear optimization, which is prone to becoming stuck in non-global minima. Although these problems are true of many system identification methods, they are particularly troublesome when relatively small amounts of data are available [40], which is often the case with step-response experiments.

One approach is to repeatedly integrate the response to construct a set of auxiliary signals which have a linearly parameterized input-output relationship, often referred to as moment-based methods. This approach requires either explicitly parameterizing any disturbances in the measurements or use of an instrumental-variable technique to avoid integrating random variables. Details for this approach, including parameterizations for initial conditions and distrubance models first appear in Whitfield and Messali [81], with asymptotic properties derived in Sagara and Zhao [62]. Hwang and Lai [28] extend this method to include the identification of time delays when using inputs with known integrals, such as step responses. A very thorough survey of both graphical and numerical methods for identification from step-responses can be found in Ahmed, Huang, and Shah [2].

Identification methods capable of generating models of higher than second-order from noisy step-response measurements are rather rare. Even rarer are methods which easily handle multiple-input-multiple-output (MIMO) systems and initial conditions. In this chapter, we present method which overcomes these difficulties. It provides possibly high-order models from step-response measurements of multivariable systems without nonlinear optimization or the requirement that the response settle to steady-state. Additionally, separate step responses measured from each input of a multi-input system may be used all at once in the identification procedure to avoid estimating separate models for a MIMO system. In Chapter 8, we add convex constraints to the method to guarantee that the models may be converted to continuous time.

3.2 Step-Based Realization

The method derived in the following is based on the realization-theoretic framework presented earlier, which constructs system estimates from an impulse response. The method was originally derived for single-input-multi-output systems in van Helmont, van der Weiden, and Anneveld [74], which applied the technique to control design for coal-fired water boilers. It appears to have been mostly forgotten until de Callafon, Moaveni, Conte, He, and Udd [15], which used Markov parameter estimates to generalize the method to arbitrary input-output signals. An alternative generalization to input-output data will be presented in Chapter 4. It was re-derived by the author in the context of subspace identification methods in Miller and de Callafon [52]; this alternative interpretation easily extends to multi-input systems and systems which have nonzero initial conditions.

3.2.1 Step-Response Data Matrix Equations

We first derive the method for single-input systems initially at steady-state. Consider again the state-space system (1.2), and suppose for now that the system is single-input with no noise. Let $y(t)$ be the output to a unit step applied from rest at $t = 0$. We wish to express $y(t)$ in terms of the system Hankel matrix H_k in a way that allows us to use the properties of H_k to identify the system. Let $Y_{1|k}$ be a block-Hankel matrix of output data

$$Y_{1|k} = \begin{bmatrix} y(1) & y(2) & \cdots & y(l) \\ y(2) & y(3) & \cdots & y(l+1) \\ \vdots & \vdots & & \vdots \\ y(k) & y(k+1) & \cdots & y(k+l-1) \end{bmatrix} \in \mathbb{R}^{n_y k \times l}. \tag{3.1}$$

This matrix may be expressed in terms of an upper-triangular Toeplitz matrix of past input data

$$U_p = \begin{bmatrix} u(0) & u(1) & \cdots & u(l-1) \\ & u(0) & \cdots & u(l-2) \\ & & \ddots & \vdots \\ & & & u(0) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ & 1 & \cdots & 1 \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix} \in \mathbb{R}^{l \times l},$$

a Hankel matrix of input data, which in this case is entirely 1,

$$U_{1|k} = \begin{bmatrix} u(1) & u(2) & \cdots & u(l) \\ u(2) & u(3) & \cdots & u(l+1) \\ \vdots & \vdots & & \vdots \\ u(k) & u(k+1) & \cdots & u(k+l-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{k \times l},$$

the finite system Hankel matrix $H_k \in \mathbb{R}^{n_y k \times l}$ from (2.5), and a block-Toeplitz matrix of Markov parameters

$$T_{0|k-1} = \begin{bmatrix} G(0) & & & \\ G(1) & G(0) & & \\ \vdots & \vdots & \ddots & \\ G(k-1) & G(k-2) & \cdots & G(0) \end{bmatrix} \in \mathbb{R}^{n_y k \times k},$$

as

$$Y_{1|k} = H_k U_p + T_{0|k-1} U_{1|k}. \tag{3.2}$$

Because $U_{1|k}$ is a matrix entirely of 1, the effect of the left multiplication by $U_{1|k}$ in (3.2) is to add all entries in each row of $T_{0|k-1}$,

$$T_{0|k-1}U_{1|k} = \begin{bmatrix} G(0) & G(0) & \cdots & G(0) \\ G(0)+G(1) & G(0)+G(1) & \cdots & G(0)+G(1) \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{k-1} G(i) & \sum_{i=0}^{k-1} G(i) & \cdots & \sum_{i=0}^{k-1} G(i) \end{bmatrix}.$$

Because $u(t)$ is 1,

$$y(t) = \sum_{k=0}^{t} G(t),$$

and the product $T_{0|k-1}U_{1|k}$ can be expressed in terms of $y(t)$ as

$$M = \begin{bmatrix} y(0) & y(0) & \cdots & y(0) \\ y(1) & y(1) & \cdots & y(1) \\ \vdots & \vdots & & \vdots \\ y(k-1) & y(k-1) & \cdots & y(k-1) \end{bmatrix} = T_{0|k-1}U_{1|k}. \tag{3.3}$$

M may then be used to solve for $H_k U_p$:

$$H_k U_p = Y_{1|k} - M.$$

If $Y_{2|k+1}$ is $Y_{1|k}$ shifted forward by one time step to form

$$Y_{2|k+1} = \begin{bmatrix} y(2) & y(3) & \cdots & y(l+1) \\ y(3) & y(4) & \cdots & y(l+2) \\ \vdots & \vdots & & \vdots \\ y(k+1) & y(k+1) & & y(k+l) \end{bmatrix}, \tag{3.4}$$

this may be expressed in terms of the shifted Hankel matrix H'_k in (2.6) as

$$Y_{2|k+1} = H'_k U_p + T_{1|k} U_{1|k+1} \tag{3.5}$$

where the block-Toeplitz matrix of Markov parameters now has an additional column on the left,

$$T_{1|k} = \begin{bmatrix} G(1) & G(0) & & & \\ G(2) & G(1) & G(0) & & \\ \vdots & \vdots & \vdots & \ddots & \\ G(k) & G(k-1) & G(k-2) & \cdots & G(0) \end{bmatrix}$$

and the "future" input data matrix has an additional row so that

$$U_{1|k+1} = \left[\begin{array}{ccccc} & & U_{1|k} & & \\ \hline u(k+1) & & \cdots & & u(k+l) \end{array} \right].$$

Similarly,

$$M' = T_{1|k} U_{1|k+1} = \begin{bmatrix} y(1) & y(1) & \cdots & y(1) \\ y(2) & y(2) & \cdots & y(2) \\ \vdots & \vdots & & \vdots \\ y(k) & y(k) & \cdots & y(k) \end{bmatrix}. \tag{3.6}$$

Thus M' may be used to solve for $H'_k U_p$ via

$$H'_k U_p = Y_{2|k+1} - M'.$$

As with estimation directly from the system Hankel matrix, in the deterministic case, any appropriately-dimensioned factorization

$$H_k U_p = (\mathcal{O}_k)(\mathcal{C}_l U_p) \tag{3.7}$$

results in a solution

$$A = \mathcal{O}_k^\dagger (H'_k U_p)(\mathcal{C}_l U_p)^\dagger = \mathcal{O}_k^\dagger H'_k \mathcal{C}_l^\dagger$$

when the matrices are constructed from noise-free data. Nondeterministic effects, however, will result in the data matrix equations (3.2) and (3.5) having error terms

$$Y_{1|k} = H_k U_p + T_{0|k-1} U_{1|k} + E \tag{3.8}$$

$$Y_{2|k+1} = H'_k U_p + T_{1|k} U_{1|k+1} + E'. \tag{3.9}$$

The result is that $Y_{1|i}$ will have full rank, and a factorization (3.7) will not exist. As before, we solve this problem by reducing the rank of the data matrices.

3.2.2 Rank Reduction and Factorization

If the output $y(t)$ is corrupted by measurement noise, we have only the approximations

$$M \approx T_{0|k-1} U_{1|k}$$

$$M' \approx T_{1|k} U_{1|k+1}$$

to remove the effects of the input after time $t = 1$. This will cause the estimates

$$H_k U_p \approx Y_{1|k} - M$$

$$H'_k U_p \approx Y_{2|k+1} - M'$$

to have full rank. From the SVD

$$Y_{1|k} - M = \begin{bmatrix} U_n & U_s \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_s \end{bmatrix} \begin{bmatrix} V_n^T \\ V_s^T \end{bmatrix} \tag{3.10}$$

a 2-norm and Frobenius-norm optimal rank-n estimate for $H_k U_p$ is

$$H_k U_p \approx \hat{H}_k U_p = U_n \Sigma_n V_n^T.$$

Since U_p is known, at this point one could try to invert U_p to solve for $\hat{H}_k$. Examining the inverse of U_p, however, reveals

$$U_p^{-1} = \begin{bmatrix} 1 & -1 & & & 0 \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ 0 & & & 1 & -1 \end{bmatrix},$$

and so multiplication on the right by U_p^{-1} is equivalent to numerically differentiating the output data, which will amplify high-frequency noise. Fortunately, removing the effects of U_p is unnecessary. Similar to the factorization used with direct estimates of H_k, we factor $\hat{H}_k U_p$ as

$$\hat{\mathcal{O}}_k = U_n \Sigma_n^{1/2} \qquad \text{and} \qquad \hat{\mathcal{C}}_l U_p = \Sigma_n^{1/2} V_n^T.$$

A suitable estimate for the system dynamics is then

$$\begin{aligned} \hat{A} &= \Sigma_n^{-1/2} U_n^T (Y_{2|k+1} - M') V_n \Sigma_n^{-1/2} \\ &= (\hat{\mathcal{O}}_k)^\dagger (Y_{2|k+1} - M')(\hat{\mathcal{C}}_l U_p)^\dagger \\ &\approx (\hat{\mathcal{O}}_k)^\dagger H'_k U_p (\hat{\mathcal{C}}_l U_p)^\dagger \\ &\approx \mathcal{O}_k^\dagger H'_k \mathcal{C}_l^\dagger, \end{aligned} \tag{3.11}$$

An estimate $\hat{C}$ may subsequently be taken as the first n_y rows of $\hat{\mathcal{O}}_k$.

An interesting property of (3.11) is that given estimates $\hat{\mathcal{O}}_k$ and $\hat{\mathcal{C}}_l U_p$, (3.11) minimizes the cost function

$$J_s(\hat{A}) = \left|\left| \hat{\mathcal{O}}_k \hat{A} (\hat{\mathcal{C}}_l U_p) - Y_{2|k+1} + M' \right|\right|_F. \tag{3.12}$$

This will be used later to form a constrained step-based realization procedure.

Because the first column of U_p is $(1\ 0\ \cdots)^T$, the first column of $\hat{\mathcal{C}}_l U_p$ will provide a possible estimate $\hat{B}$. Then with $\hat{D}$ taken as $y(0)$, a complete state-space realization is found. Because B and D are linear in the output data, however, optimal estimates may be calculated in a linear least-squares problem once $\hat{A}$ and $\hat{C}$ are known. Before formulating the least-squares problem, we first examine the effects of initial conditions on the step-based realization procedure.

3.2.3 Nonzero Initial Conditions

For now, we return to the noise-free case. Nonzero initial conditions may be represented as an additive past-input term U_i so that

$$Y_{1|k} = H_k U_p + \mathcal{O}_k \mathcal{C} U_i + T_{0|k-1} U_{1k}$$

where

$$U_i = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ u(-1) & 0 & 0 & \cdots & 0 \\ u(-2) & u(-1) & 0 & \cdots & 0 \\ u(-3) & u(-2) & u(-1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix}. \tag{3.13}$$

U_i may potentially have infinite row dimension; H_k will have infinite column dimension, and the product $H_k(U_p + U_i)$ will still be finite. From the structure of (3.13), we can see that $(U_p + U_i)$ will still have full column rank unless $u(t) = 1$ for all $t < 0$, that is, unless the system is at steady-state throughout the entire experiment.

To examine what effect the extended past input has on the output data matrix, we instead represent its effects in terms of the state vector $x(t)$:

$$\begin{aligned} H_k(U_p + U_i) &= \mathcal{O}_k \mathcal{C}_l U_p + \mathcal{O}_k \mathcal{C} U_i \\ &= \mathcal{O}_k \begin{bmatrix} x(1) & x(2) & \cdots & x(l) \end{bmatrix} \\ &= \mathcal{O}_k X. \end{aligned}$$

The effect of shifting the Markov parameters in H_k is then

$$\begin{aligned} H'_k(U_p + U_i) &= \mathcal{O}_k A \mathcal{C}_l U_p + \mathcal{O}_k A \mathcal{C} U_i \\ &= \mathcal{O}_k A X, \end{aligned}$$

and the data matrix equations become

$$Y_{1|k} = \mathcal{O}_k X + T_{0|k-1} U_{1|k}$$

and

$$Y_{2|k+1} = \mathcal{O}_k A X + T'_{1|k} U_{1|k+1}.$$

Because the output $y(t)$ is now

$$y(t) = \sum_{k=0}^{t} G(t) + C A^t x(0),$$

M and M' have additive terms

$$M = T_{0|k-1}U_{1|k} + \mathcal{O}_k X_0$$
$$M' = T'_{0|k-1}U_{1|k+1} + \mathcal{O}_k A X_0,$$

where

$$X_0 = \begin{bmatrix} x(0) & x(0) & \cdots & x(0) \end{bmatrix} \in \mathbb{R}^{n \times l}.$$

Subtracting M and M' as is done with zero initial conditions results in

$$Y_{1|k} - M = \mathcal{O}_k X - \mathcal{O}_k X_0 = \mathcal{O}_k (X - X_0)$$
$$Y_{2|k+1} - M' = \mathcal{O}_k A X - \mathcal{O}_k A X_0 = \mathcal{O}_k A (X - X_0).$$

Non-deterministic effects will result in error terms for both equations and require a rank-reducing factorization as before. The SVD (3.10) may then be instead used to factor estimates

$$\hat{\mathcal{O}}_k = U_n \Sigma_n^{1/2} \qquad \text{and} \qquad \hat{X} - \hat{X}_0 = \Sigma_n^{1/2} V_n^T$$

and a suitable estimate for the system dynamics is

$$\begin{aligned} \hat{A} &= \Sigma_n^{1/2} U_n^T (Y_{2|k+1} - M') V_n \Sigma_n^{-1/2} \\ &= (\hat{\mathcal{O}}_k)^\dagger (Y_{2|k+1} - M')(\hat{X} - \hat{X}_0)^\dagger \\ &\approx (\hat{\mathcal{O}}_k)^\dagger (\mathcal{O}_k) A (X - X_0)(\hat{X} - \hat{X}_0)^\dagger \\ &\approx A. \end{aligned} \tag{3.14}$$

Thus in the case of nonzero initial conditions, A may be estimated as before without any changes in the algorithm. C may still be approximated as the first n_y rows of $\hat{\mathcal{O}}_k$, but B can no longer be approximated by the first column of $\Sigma_n^{1/2} V_n^T$, and D from $y(0)$. We can, however, parameterize a linear-least-squares problem to estimate B, D, and an initial condition $x(0)$.

3.3 Estimation of Input Parameters

If A and C are known, then B, D and an initial condition $x(0)$ are linear in the input-output data and may be identified by solving a linear-least-squares

problem. This may be seen from the convolution

$$y(t) = \sum_{k=0}^{\infty} G(k)u(t-k) = \sum_{k=-\infty}^{t-1} CA^{t-k-1}Bu(k) + Du(t) + CA^t x(0). \tag{3.15}$$

For a step-response, this reduces to

$$y(t) = \sum_{k=0}^{t-1} CA^{t-k-1}B + D + CA^t x(0)$$

Thus estimates of B, D, and $x(0)$ may be found by solving the linear-least squares problem

$$\hat{\theta} = \begin{bmatrix} \hat{B} \\ \hat{D} \\ x(0) \end{bmatrix}, \qquad \hat{\theta} = \min_{\theta} \left|\left| y - \phi^T \theta \right|\right|_2, \tag{3.16}$$

where

$$y = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} \in \mathbb{R}^{n_y N} \qquad \text{and} \qquad \phi^T = \begin{bmatrix} 0 & 1 & C \\ C & 1 & CA \\ CA & 1 & CA^2 \\ CA^2 & 1 & CA^3 \\ \vdots & \vdots & \vdots \\ CA^{N-2} & 1 & CA^{N-1} \end{bmatrix}. \tag{3.17}$$

$\hat{\theta}$ has the unique solution

$$\hat{\theta} = (\phi^T)^{\dagger} y. \tag{3.18}$$

Because $\hat{\theta}$ is the solution to a least-squares problem, it will provide more optimal estimates of B and D than would be obtained from the first column of $\hat{\mathcal{O}}_l$ and $y(0)$, respectively. If it is assumed that the system has a known time-delay so that $D = 0$, or that the system has zero initial conditions so that $x(0) = 0$, then the relevant rows of θ and columns of ϕ^T can be removed.

3.4 Extension to Multi-Input Systems

The step-based realization procedure can be trivially extended to multi-input systems if separate experiments are applied for each input; it turns out that,

as with the case of nonzero initial conditions, the generalized realization procedure requires virtually no modification.

If a system has multiple inputs, separate steps on each input are necessary to identify the system from step-response measurements. The step-based realization procedure, however, can be performed on all data at once. Suppose the input $u(t)$ has dimension n_u. Let $Y_{1|k}^{(i)}$ and $Y_{2|k+1}^{(i)}$ be the data matrices (3.1) and (3.4), respectively, constructed from the response of the system to a unit step applied to input i. If we extend the definition of $Y_{1|k}$ and $Y_{2|k+1}$ to

$$Y_{1|k} = \begin{bmatrix} Y_{1|k}^{(1)} & \cdots & Y_{1|k}^{(n_u)} \end{bmatrix} \qquad \text{and} \qquad Y_{2|k} = \begin{bmatrix} Y_{2|k+1}^{(1)} & \cdots & Y_{2|k+1}^{(n_u)} \end{bmatrix}, \tag{3.19}$$

and similarly extend the definitions

$$U_{1|k} = \begin{bmatrix} U_{1|k}^{(1)} & \cdots & U_{1|k}^{(n_u)} \end{bmatrix}, \qquad U_{1|k+1} = \begin{bmatrix} U_{1|k+1}^{(1)} & \cdots & U_{1|k+1}^{(n_u)} \end{bmatrix},$$

and

$$U_p = \begin{bmatrix} U_p^{(1)} & \cdots & U_p^{(n_u)} \end{bmatrix},$$

then these extended matrices still satisfy (3.2) and (3.5). By similarly extending the definitions of M and M' to be the composite matrices

$$M = \begin{bmatrix} M^{(1)} & \cdots & M^{(n_u)} \end{bmatrix} \qquad \text{and} \qquad M' = \begin{bmatrix} M'^{(1)} & \cdots & M'^{(n_u)} \end{bmatrix}, \tag{3.20}$$

we see that

$$M \approx T_{0|k-1} \begin{bmatrix} U_{1|k}^{(1)} & \cdots & U_{1|k}^{(n_u)} \end{bmatrix} \qquad \text{and} \qquad M' \approx T_{1|k} \begin{bmatrix} U_{1|k+1}^{(1)} & \cdots & U_{1|k+1}^{(n_u)} \end{bmatrix},$$

and thus the step-based realization procedure may be performed on multiple step-response data sets. As with the single-input case, nonzero initial conditions require no modifications to the algorithm. With A and C estimated, the individual columns of B and D and an initial condition $x(0)$ can be estimated via linear least squares for each experiment.

3.5 Summary of the Step-Based Realization Procedure

The following steps completely describe the step-based realization algorithm:

1. Construct the block-Hankel data matrices $Y_{1|k}$ in (3.1) and $Y_{2|k+1}$ in (3.4). Concatenate blocks of $Y_{1|k}^{(i)}$ and $Y_{2|k+1}^{(i)}$ as in (3.19) if multi-input data is used.

2. Construct the column-identical matrices M in (3.3) and M' in (3.6). Concatenate blocks of $M^{(i)}$ and $M'^{(i)}$ as in (3.20) if multi-input data is used.

3. Take the SVD of $Y_{1|k} - M$. Examine the singular values to determine the system order if needed.

4. Partition the SVD as in (3.10). Solve for $\hat{A}$ from (3.14) and $\hat{C}$ from the first n_y rows of $U_n \Sigma_n^{1/2}$.

5. From $\hat{A}$ and $\hat{C}$, construct the regressor ϕ^T in (3.17) and solve for $\hat{B}$, $\hat{D}$, and $\hat{x}(0)$ from (3.14), removing corresponding rows of θ and columns of ϕ^T if D or $x(0)$ is known to be 0.

The procedure is demonstrated in the following example, and in Chapter 8.

Example 3.1. Consider the 4$^{\text{th}}$-order system with state-space parameters

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cccc|c} 0.95 & 0.1 & 0 & 0 & -0.01209 \\ -0.1 & 0.95 & 0 & 0 & 0.2429 \\ 0 & 0 & 0.93 & 0.3 & 0 \\ 0 & 0 & -0.3 & 0.93 & 0.4246 \\ \hline 0.4039 & 0.4039 & -0.0955 & -0.2196 & 0 \\ 0.2 & 0.3 & -0.4 & 0.3 & 0 \end{array}\right].$$

The deterministic step response of this system is seen in Figure 3.1. Suppose independent white noise signals of variance $\sigma^2 = 0.003$ are added to each output of the measured data. A sample realization of the measured step response together with the true response is shown in Figure 3.1.

The step-based realization procedure was applied to the measured noisy data. The data matrices were given a block row dimension of $k = 30$. The singular values of $Y_{1|k} - M$ are shown in Figure 3.2. The model order $n = 4$ is clearly visible in the magnitude of the singular values. The response of the system estimate is

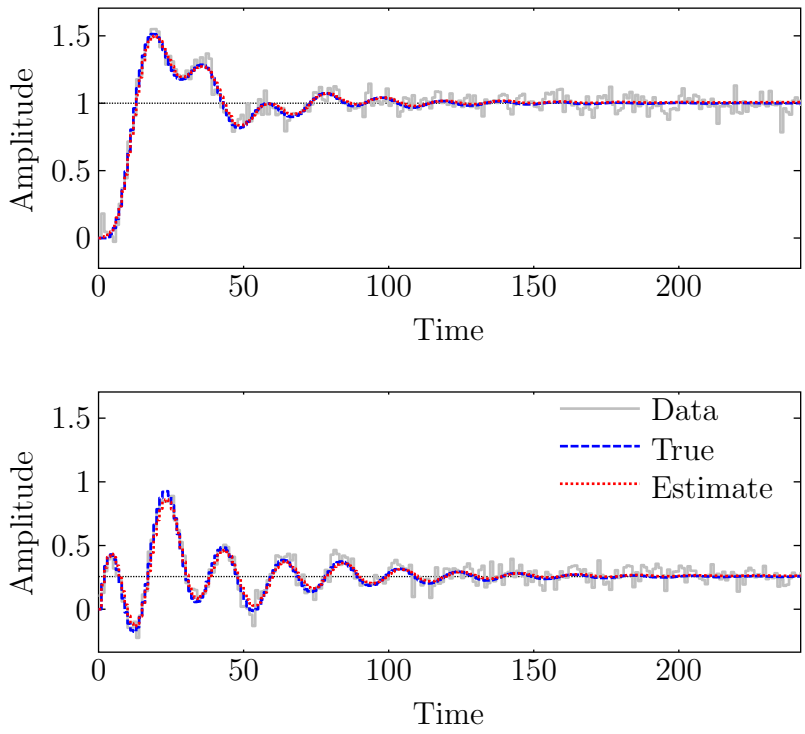

Figure 3.1: Step response for Example 3.1.

shown with the measured and true responses in Figure 3.1. It was assumed that $D = 0$ is known for the least-squares solution of B.

For comparison purposes, the raw numerical derivatives of the noisy and noise-free step responses are shown in Figure 3.3. Methods that rely on numerical differentiation will evidently require sophisticated smoothing techniques.

3.6 Acknowledgements

This chapter, in part, includes content from the following publications:

D.N. Miller and R.A. de Callafon, "Step-Based Realization with Eigenvalue Constraints," to appear, *16th IFAC Symposium on System Identification*, Brussels, Belgium: July 2012.

D.N. Miller and R.A. de Callafon, "Subspace Identification from Classical Realization Methods," *Proc. of the 15th IFAC Symposium on System Identification*,

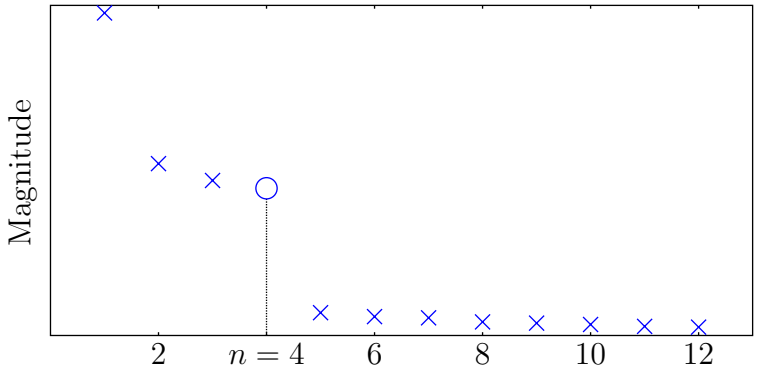

Figure 3.2: Singular values for Example 3.1.

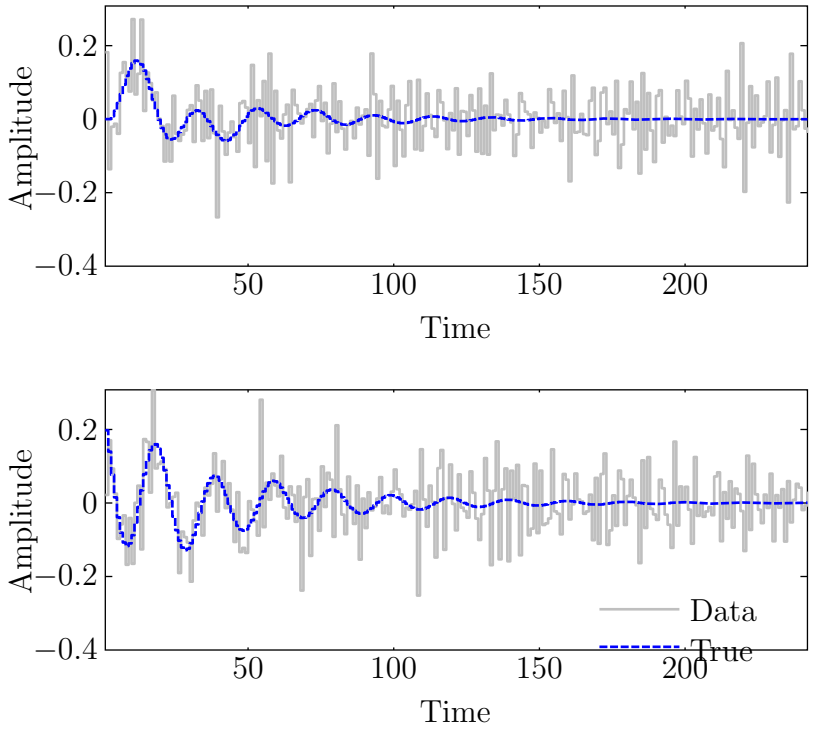

Figure 3.3: Differentiated step response for Example 3.1.

Saint-Malo, France: July 2009.

4 Realization from Input-Output Data

Although the step-based realization method provides very good estimates even in the case of noise-corrupted measurements, convergence to the true system can only be guaranteed if the step-response is repeatedly measured and the output data averaged. In the following, we extend the realization-based identification method to arbitrary sequences of input-output data. This method is guaranteed to converge to the true system estimate as the length of the dataset increases. The final method can be categorized as a subspace identification method. Differences between the presented method and alternative subspace methods will be discussed in the conclusion of the chapter.

4.1 Input-Output Data Matrix Equations

We begin by reformulating the previous data-matrix equations for the general case. Some definitions from previous sections are repeated here to aid the theoretical development. Let the input-output behavior of a discrete-time, LTI system be described in state-space form with a noise signal $v(t) \in \mathbb{R}^{n_y}$ added to the output:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) + v(t). \end{aligned} \tag{4.1}$$

The noise $v(t)$ may be either white or colored, which includes the case of the noise-generating process sharing eigenvalues with A.

Consider a block-Hankel matrix of r block rows and l columns of measured output data starting with the sample $y(0)$,

$$Y_{0|r-1} = \begin{bmatrix} y(0) & y(1) & \cdots & y(l-1) \\ y(1) & y(2) & \cdots & y(l) \\ \vdots & \vdots & & \vdots \\ y(r-1) & y(r) & \cdots & y(r+l-2) \end{bmatrix} \in \mathbb{R}^{n_y r \times l}, \tag{4.2}$$

and a block-Hankel matrix of r block rows and l columns of measured input data starting with the sample $u(0)$,

$$U_{0|r-1} \begin{bmatrix} u(0) & u(1) & \cdots & u(l-1) \\ u(1) & u(2) & \cdots & u(l) \\ \vdots & \vdots & & \vdots \\ u(r-1) & u(r) & \cdots & u(r+l-2) \end{bmatrix} \in \mathbb{R}^{n_u r \times l}. \tag{4.3}$$

These data matrices satisfy

$$Y_{0|r-1} = H_r U_p + T_{0|r-1} U_{0|r-1} + V_{0|r-1}, \tag{4.4}$$

where

$$H_r = \begin{bmatrix} G(1) & G(2) & G(3) & \cdots \\ G(2) & G(3) & G(4) & \cdots \\ \vdots & \vdots & \vdots & \\ G(r) & G(r+1) & G(r+2) & \cdots \end{bmatrix} \in \mathbb{R}^{n_y r \times \infty},$$

$$U_p = \begin{bmatrix} u(-1) & u(0) & \cdots & u(l-2) \\ u(-2) & u(-1) & \cdots & u(l-3) \\ u(-3) & u(-2) & \cdots & u(l-4) \\ \vdots & \vdots & & \vdots \end{bmatrix} \in \mathbb{R}^{\infty \times l},$$

$$T_{0|r-1} = \begin{bmatrix} G(0) & & & \\ G(1) & G(0) & & \\ \vdots & \vdots & \ddots & \\ G(r-1) & G(r) & \cdots & G(0) \end{bmatrix} \in \mathbb{R}^{n_y r \times n_u r},$$

and $V_{0|r-1}$ is a matrix of noise $v(t)$ with the same dimensions and indices as $Y_{0|r-1}$.

Similarly, a forward-shifted block-Hankel output data matrix

$$Y_{1|r} = \begin{bmatrix} y(1) & y(2) & \cdots & y(l) \\ y(2) & y(3) & \cdots & y(l+1) \\ \vdots & \vdots & & \vdots \\ y(r) & y(r+1) & \cdots & y(r+l-1) \end{bmatrix} \in \mathbb{R}^{n_y r \times l} \tag{4.5}$$

may be expressed in terms of a forward-shifted system Hankel matrix

$$H'_r = \begin{bmatrix} G(2) & G(3) & G(4) & \cdots \\ G(3) & G(4) & G(5) & \cdots \\ \vdots & \vdots & & \vdots \\ G(r+1) & G(r+2) & G(r+3) & \cdots \end{bmatrix} \in \mathbb{R}^{n_y r \times \infty}$$

and a shifted block-Toeplitz matrix

$$\begin{aligned} T_{1|r} &= \begin{bmatrix} G(1) & G(0) & & & \\ G(2) & G(1) & G(0) & & \\ \vdots & \vdots & \vdots & \ddots & \\ G(r) & G(r-1) & G(r-2) & \cdots & G(0) \end{bmatrix} \\ &= \left[\begin{array}{c|c} G(1) & \\ \vdots & T_{0|r-1} \\ G(r) & \end{array}\right] \in \mathbb{R}^{n_y r \times n_u (r+1)} \end{aligned}$$

as

$$Y_{1|r} = H'_r U_p + T_{1|r} U_{0|r} + V_{1|r}, \tag{4.6}$$

in which

$$U_{0|r} = \left[\begin{array}{c} U_{0|r-1} \\ \hline u(r) \quad \cdots \quad u(r+l-1) \end{array}\right] \in \mathbb{R}^{n_u (r+1) \times l} \tag{4.7}$$

and $V_{1|r}$ is a block-Hankel matrix of noise terms $v(t)$ with the same indices as $Y_{1|r}$.

An important quality of (4.4) and (4.6) is that the index r in $U_{0|r-1}$ may be replaced with any index $\bar{r} \geq r$ without invalidating the equations; causality and dimensional consistency are retained by appending '0's to the right-hand side of

$T_{0|r-1}$. Thus $U_{0|r-1}$ may be replaced with $U_{0|r}$ and $T_{0|r-1}$ with

$$T'_{0|r-1} = \left[\begin{array}{c|c} & 0 \\ T_{0|r-1} & \vdots \\ & 0 \end{array}\right].$$

in (4.4) so that both equations contain the same input matrix:

$$Y_{0|r-1} = H_r U_p + \underbrace{T'_{0|r-1} U_{0|r}}_{=T_{0|r-1}U_{0|r-1}} + V_{0|r-1}$$

This will have important consequences when forming projection matrices to remove $U_{0|r}$ in Section 4.2.

The column dimensions of H_r and H'_r and the row dimension of U_p may be infinite in the above equation, but their product is finite. To express (4.4) and (4.6) with terms of finite dimensions, we note that a matrix of system states satisfies

$$X = \begin{bmatrix} x(0) & x(1) & \cdots & x(l-1) \end{bmatrix} = \mathcal{C} U_p \in \mathbb{R}^{n \times l},$$

where $\mathcal{C}$ has infinite column dimension. Hence (4.4) and (4.6) may be alternatively stated as

$$Y_{0|r-1} = \mathcal{O}_r X + T'_{0|r-1} U_{0|r} + V_{0|r-1} \tag{4.8}$$

and

$$Y_{1|r} = \mathcal{O}_r A X + T_{1|r} U_{0|r} + V_{1|r}. \tag{4.9}$$

In this case the shift-invariance of H_r is seen explicitly. Readers familiar with traditional subspace identification methods will recognize that (4.8) with $U_{0|r-1}$ in place of $U_{0|r}$ is the fundamental "data equation" employed by most subspace algorithms.

As with step-based realization, our goal is to remove the effects of $T'_{0|r-1}$ and $T_{1|r}$ from the row spaces of $Y_{0|r-1}$ and $Y_{1|r}$, respectively, and then perform the realization-type procedure on the results. We could try a tactic similar to the approximations used in the step-based method. Suppose we have measurements of $u(t)$ and $y(t)$ for s samples prior to $t = 0$. Given the block reflector matrix

$$R = \begin{bmatrix} I_{n_u} & & 0 \\ & \ddots & \\ 0 & & I_{n_u} \end{bmatrix}$$

and some "past horizon" index $s > l$, we can find a relationship

$$\begin{bmatrix} Y_{-s|-1} \\ Y_{0|r-1} \end{bmatrix} = \begin{bmatrix} T_{0|r-1} & \\ H_r R & T_{0|r-1} \end{bmatrix} \begin{bmatrix} U_{-s|-1} \\ U_{0|r-1} \end{bmatrix} + \mathcal{O}_{r+s} X_{-s}, \tag{4.10}$$

where

$$Y_{-s|-1} = \begin{bmatrix} y(-s) & y(-s+1) & \cdots & y(-s+l-1) \\ y(-s+1) & y(-s+2) & \cdots & y(-s+l) \\ \vdots & \vdots & & \vdots \\ y(-1) & y(0) & \cdots & y(r+l-1) \end{bmatrix},$$

$U_{-s|-1}$ is defined similarly, and

$$X_{-s} = \begin{bmatrix} x(-s) & \cdots & x(-s+l-1) \end{bmatrix}$$

Assuming $U_{-s|-1}$ has full row rank, from

$$Y_{-s|-1} = T_{0|r-1} U_{-s|-1} + \mathcal{O}_s X_{-s},$$

we find

$$Y_{-s|-1} U_{-s|-1}^{\dagger} U_{0|r} = T'_{0|r-1} U_{0|r} + \mathcal{O}_s X_{-s} U_{-s|-1}^{\dagger} U_{0|r},$$

and thus, in the deterministic case, subtraction of (4.8) by $Y_{-s|-1} U_{-s|-1}^{\dagger} U_{0|r-1}$ will remove the effects of $T_{0|r-1}$ from the row space of $Y_{0|r-1}$. In fact, this does indeed reduce to the step-based method when used with step-response data if $s = r$ and $Y_{0|r-1}$ instead starts with sample $y(1)$. Thus a potential candidate for the realization-based procedure are the pair of matrices

$$\begin{aligned} H_r U_p &\approx Y_{0|r-1} - Y_{-s|-1} U_{-s|-1}^{\dagger} U_{0|r} \\ H'_r U_p &\approx Y_{1|r} - Y_{-s+1|0} U_{-s+1|0}^{\dagger} U_{0|r}. \end{aligned}$$

While this will provide exact results in the deterministic case, the right-hand multiplication of $U_{-s|-1}^{\dagger}$ will tend to amplify noise for arbitrary input sequences and degrade the resulting estimate. Fortunately, when $U_{0|r}$ has full row rank, we can incorporate the use of projection matrices to remove the row spaces of $T_{0|r-1}$ and $T_{1|r}$ in a least-squares sense.

4.2 Null Space Projection of the Data Matrices

To remove the effects of future input on the row space of $Y_{0|r-1}$ and $Y_{1|r}$, we construct null-space projectors of the input. This projection has some optimality properties that make it well suited to experiments with input sequences that result in a $U_{0|r}$ with full row rank. The projection can be interpreted in two ways: a geometric projection and a Gram-Schmidt type orthogonalization of the data. We first present the projection as a solution to a least-squares problem, then discuss both interpretations.

4.2.1 Least-Squares Solution of the Extended Observability Matrix Row-Space

For the step-based method, we used knowledge of $U_{0|r-1}$ to estimate the product $T_{0|r-1}U_{0|r-1}$. For the case of arbitrary input and output data, however, we can find an optimal estimate of $T_{0|r-1}$ when $U_{0|r}$ has full row rank. For the unshifted data-matrix equation (4.8), we define optimality by solving the following least-squares problem [79]:

$$\min_{\hat{T}'_{0|r-1}} \left|\left| Y_{0|r-1} - \hat{T}'_{0|r-1}U_{0|r} \right|\right|_F . \tag{4.11}$$

If $U_{0|r}$ has full row rank, then this has the analytical solution

$$\hat{T}'_{0|r-1} = Y_{0|r-1}U^{\dagger}_{0|r}.$$

Thus a suitable estimate for the row space of $\mathcal{O}_r$ may be found from

$$\begin{aligned} Y_{0|r-1} - \hat{T}'_{0|r-1}U_{0|r} &= Y_{0|r-1} - Y_{0|r-1}U^{\dagger}_{0|r}U_{0|r} \\ &= Y_{0|r-1}\left(I_l - U^T_{0|r}(U_{0|r}U^T_{0|r})^{-1}U_{0|r}\right) \\ &= Y_{0|r-1}\Pi_{U^{\perp}} \\ &= \mathcal{O}_r X \Pi_{U^{\perp}} + V_{0|r-1}\Pi_{U^{\perp}}. \end{aligned}$$

The matrix

$$\Pi_{U^{\perp}} = I_l - U^T_{0|r}(U_{0|r}U^T_{0|r})^{-1}U_{0|r} \tag{4.12}$$

is the projector for the null-space of $U_{0|r}$ and satisfies the property

$$U_{0|r}\Pi_{U^\perp} = 0.$$

For the shifted data-matrix equation (4.9), we solve the similar problem

$$\min_{\hat{T}_{1|r}} \left|\left| Y_{1|r} - \hat{T}_{1|r} U_{0|r} \right|\right|_F .$$

This has the solution

$$\hat{T}_{1|r} = Y_{1|r} U_{0|r}^\dagger,$$

and thus the estimate used to find $\mathcal{O}_r A$ can be taken from

$$\begin{aligned} Y_{1|r} - \hat{T}_{1|r} U_{0|r} &= Y_{1|r} - Y_{1|r} U_{0|r}^\dagger U_{0|r} \\ &= Y_{1|r} \left(I_l - U_{0|r}^T (U_{0|r} U_{0|r})^{-1} U_{0|r} \right) \\ &= Y_{1|r} \Pi_{U^\perp} \\ &= \mathcal{O}_r A X \Pi_{U^\perp} + V_{1|r} \Pi_{U^\perp}. \end{aligned}$$

It is very important that the *same* projector matrix is used for both equations so that the product $X\Pi_{U^\perp}$ appears in both equations. An estimate of $X\Pi_{U^\perp}$ from the unshifted equation will be used to estimate A from the shifted equation.

Before discussing various interpretations of the solution of (4.11), scrutinizing the equation more closely reveals that a "more optimal" estimate of $T'_{0|r-1}$ would account for the effects of $\mathcal{O}_r X$ on the output data, so that by instead minimizing

$$\min_{\hat{T}'_{0|r-1}} \left|\left| Y_{0|r-1} - \mathcal{O}_r X - \hat{T}'_{0|r-1} U_{0|r} \right|\right|_F = \min_{\hat{T}'_{0|r-1}} \left|\left| T'_{0|r-1} U_{0|r} - \hat{T}'_{0|r-1} U_{0|r} + V_{0|r-1} \right|\right|_F ,$$

the residuals would be limited to the noise term only. The category of subspace identification methods known as "Predictor-Based Subspace Identification" (PBSID) methods compute a preliminary estimate of $\mathcal{O}_r X$ for just such a purpose [13].

4.2.2 Geometric Interpretation

From now on we will always assume that the input data matrix $U_{0|r}$ has full row rank. We have seen that the optimal-in-a-sense method to remove the effects

of $T'_{0|r-1}$ and $T_{1|r}$ from the row spaces of $Y_{0|r-1}$ and $Y_{1|r}$, respectively is equivalent to multiplication on the right by the projector matrix $\Pi_{U^\perp}$:

$$Y_{0|r-1}\Pi_{U^\perp} = \mathcal{O}_r X\Pi_{U^\perp} + V_{0|r-1}\Pi_{U^\perp} \tag{4.13}$$

$$Y_{1|r}\Pi_{U^\perp} = \mathcal{O}_r AX\Pi_{U^\perp} + V_{1|r}\Pi_{U^\perp}. \tag{4.14}$$

Geometrically, this a *null-space projection* onto the null space of $U_{0|r}$. This projection is guaranteed to separate the output from the future input. This also has the effect of projecting the states X onto the null space of $U_{0|r}$ as well, and thus we see why it is necessary to use the same projection for both unshifted and shifted equations. Projector matrices such as $\Pi_{U^\perp}$ have many interesting properties, an overview of which can be found in Appendix B.1.

Because $U_{0|r}$ has full row rank, the dimension of the null space of $U_{0|r}$ is the difference between its columns and its rows [65]:

$$\dim(\text{null}(U_{0|r})) = l - r,$$

and thus $\text{rank}(\Pi_{U^\perp}) = l - r$. A necessary condition to preserve the rank of $\mathcal{O}_r X$ and $\mathcal{O}_r AX$ when multiplying by $\Pi_{U^\perp}$ is that $\Pi_{U^\perp}$ have rank n. Hence

$$l \geq n + r$$

is a necessary condition to preserve the rank of $\mathcal{O}_r X$ and $\mathcal{O}_r AX$. This provides a necessary condition on the dimensions of the data matrices. We must also satisfy

$$\text{rank}(X\Pi_{U^\perp}) = n,$$

so that the state dimension is preserved under the projection. This may safely be assumed true for nearly all input signals. For an extensive analysis of both necessary and sufficient conditions to preserve the rank of $\mathcal{O}_r X$ under a null space projection and the relationship of the conditions to persistency of excitation see Willems, Rapisarda, Markovsky, and De Moor [82].

4.2.3 LQ-Factorization Interpretation

Rather than implementing the projections of (4.13) and (4.14) by explicitly forming projector matrices, the projected matrix products $Y_{0|r-1}\Pi_{U^\perp}$ and $Y_{1|r}\Pi_{U^\perp}$

may be found within LQ-decompositions. The LQ-decomposition of a matrix is the transpose of the QR-decomposition of its transpose, that is,

$$P = LQ^T \Leftrightarrow P^T = QL^T = QR.$$

If we take the LQ decompositions

$$\begin{bmatrix} U_{0|r} \\ Y_{0|r-1} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$$

and

$$\begin{bmatrix} U_{0|r} \\ Y_{1|r} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L'_{21} & L'_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ {Q'_2}^T \end{bmatrix},$$

it can be seen from substitution that

$$Y_{0|r-1}\Pi_{U^\perp} = L_{22}Q_2^T \tag{4.15}$$

$$Y_{1|r}\Pi_{U^\perp} = L'_{22}{Q'_2}^T \tag{4.16}$$

The algorithm may then be performed with $L_{22}Q_2^T$ in place of $Y_{0|r-1}\Pi_{U^\perp}$ and $L'_{22}{Q'_2}^T$ in place of $Y_{1|r}\Pi_{U^\perp}$. This calculation, though less unintuitive, is far more numerically efficient.

In addition to being more numerically efficient, the LQ-decomposition has an interesting interpretation by itself. Because the individual columns of $\begin{bmatrix} U_{0|r} \\ Y_{0|r-1} \end{bmatrix}$ and $\begin{bmatrix} U_{0|r} \\ Y_{1|r} \end{bmatrix}$ are pairs of input-output sequences, the LQ-decomposition is equivalent to a Gram-Schmidt process in which the principle of superposition is exploited to form new pairs of input-output sequences. The columns of the blocks $L_{22}Q_2^T$ and $L'_{22}{Q'_2}^T$ then correspond to free-response output data since the input for these newly formed sequences is 0. This interpretation is primarily due to Katayama [37].

4.3 Classical Subspace Identification Methods

Traditional subspace methods focus on (4.13) only, with the projector (4.12) constructed instead from $U_{0|r-1}$. The system dynamics are then estimated from

some shift-invariant property of the factorization. Though some notable exceptions exist, most subspace identification methods are identical up to this point.

As with the other realization-based methods, we will use the SVD of the data matrices for rank reduction. In the following section, we briefly review the two most common subspace methods and their relationship to the general realization problem. We then present an alternative method that uses the shift invariance found in (4.13) and (4.14) to estimate the system dynamics.

4.3.1 MOESP

If the noise $v(t)$ is white and the output and input data matrices are separated into two parts

$$Y_{0|r-1} = \begin{bmatrix} Y_{0|k-1} \\ Y_{k|r-1} \end{bmatrix} \qquad \text{and} \qquad U_{0|r-1} = \begin{bmatrix} U_{0|k-1} \\ U_{k|r-1} \end{bmatrix},$$

and the projector $\Pi_{U^\perp}$ is defined with $U_{k|r-1}$ in (4.12), then the right-hand multiplication

$$\frac{1}{N} Y_{k|r-1} \Pi_{U^\perp} Y_{0|k-1}^T = \frac{1}{N} \mathcal{O}_r X \Pi_{U^\perp} Y_{0|k-1}^T + \frac{1}{N} V_{k|r-1} \Pi_{U^\perp} Y_{0|k-1}^T$$

will converge with increasing column dimension l to

$$\frac{1}{N} Y_{k|r-1} \Pi_{U^\perp} Y_{0|k-1}^T = \frac{1}{N} \mathcal{O}_r X \Pi_{U^\perp} Y_{0|k-1}^T.$$

From the SVD

$$\frac{1}{N} Y_{k|r-1} \Pi_{U^\perp} Y_{0|k-1}^T = \begin{bmatrix} U_n & U_s \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_s \end{bmatrix} \begin{bmatrix} V_n^T \\ V_s^T \end{bmatrix},$$

it can be shown that U_n converges to the row space of $\mathcal{O}_r$, and thus

$$\hat{\mathcal{O}}_r = U_n$$

will converge to the extended observability matrix corresponding to some state basis as $l \to \infty$. A can then be estimated from

$$\hat{A} = \left(\hat{\mathcal{O}}_r\right)^\dagger_{(1:(r-1)n_y,\ :)} \left(\hat{\mathcal{O}}_r\right)_{(n_y+1:rn_y,\ :)} \tag{4.17}$$

where the subscripts denote MATLAB-style indexing. This is known as Past-Output-MOESP (PO-MOESP).

If $v(t)$ is colored, then the right-hand multiplication

$$\frac{1}{N}Y_{k|r-1}\Pi_{U^\perp}U_{0|k-1}^T = \frac{1}{N}\mathcal{O}_r X\Pi_{U^\perp}U_{0|k-1}^T + \frac{1}{N}V_{k|r-1}\Pi_{U^\perp}U_{0|k-1}^T$$

will converge to

$$\frac{1}{N}Y_{k|r-1}\Pi_{U^\perp}U_{0|k-1}^T = \frac{1}{N}\mathcal{O}_r X\Pi_{U^\perp}U_{0|k-1}^T$$

under some additional assumptions placed on the input. In this case, the SVD

$$\frac{1}{N}Y_{k|r-1}\Pi_{U^\perp}U_{0|k-1}^T = \begin{bmatrix}U_n & U_s\end{bmatrix}\begin{bmatrix}\Sigma_n & 0\\ 0 & \Sigma_s\end{bmatrix}\begin{bmatrix}V_n^T\\ V_s^T\end{bmatrix}$$

has the property that row space of U_n converges to the row space of $\mathcal{O}_r$. This is known as Past-Input-MOESP (PI-MOESP). For both cases, B, D, and a possible initial condition $x(0)$ are solved for in a linear least squares problem, which will be discussed in detail at the end of the chapter. Overviews of the MOESP family of subspace methods may be found in the book Verhaegen and Verdult [78].

4.3.2 N4SID

If the noise $v(t)$ is the result of additive white noise on the state and the output, then the SVD

$$Y_{0|r-1}\Pi_{U^\perp} = \begin{bmatrix}U_n & U_s\end{bmatrix}\begin{bmatrix}\Sigma_n & 0\\ 0 & \Sigma_s\end{bmatrix}\begin{bmatrix}V_n^T\\ V_s^T\end{bmatrix}$$

may be interpreted as the factorization

$$\hat{\mathcal{O}}_r = U_n S_n^{1/2} \qquad\qquad \tilde{X} = S_n^{1/2}V_n^T$$

in which $\tilde{X} = X\Pi_{U^\perp}$ is a bank of Kalman filter states for the free response of the system. The state dynamics may then be estimated using the shift-invariant structure of $\tilde{X}$. In fact, all parameters ($A,\ B,\ C,\ D$) as well as a noise parameter K may be estimated via the least squares problem

$$\min_{\left[\begin{smallmatrix}A & B\\ C & D\end{smallmatrix}\right]}\left\|\begin{bmatrix}\tilde{X}'\\ Y\end{bmatrix} - \begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}\tilde{X}\\ U\end{bmatrix} - \begin{bmatrix}K\\ I\end{bmatrix}E\right\|_F$$

where $\tilde{X}'$ is $\tilde{X}$ shifted forward by one index and E is a column-wise sequence of noise in innovations form. This method is known as N4SID (somehow an acronym for "Numerical Methods for Subspace System Identification.")

Alternatively, (4.17) may be used to estimate the system dynamics and a least-squares problem used to estimate B, D, and K only. This is known as Robust N4SID and is implemented in the MATLAB System Identification Toolbox's "n4sid" function. Details for these methods may be found in the book Van Overschee and De Moor [76].

4.4 Subspace Identification by Shift-Invariance in the Output Data

Both the MOESP and N4SID family of algorithms provide system estimates through the shift-invariance of the factorizations which result from the SVD. In this section, we instead proposed a method which uses the shift-invariance of the output data itself to form a subspace identification method more closely related to classical realization theory.

Let us first consider the noise-free case, beginning with the projected output term

$$Y_{0|r-1}\Pi_{U^\perp} = \mathcal{O}_r X \Pi_{U^\perp}.$$

Assume $\text{rank}(Y_{0|r-1}\Pi_{U^\perp}) = n$. Given some factorization of $\mathcal{O}_r X \Pi_{U^\perp}$ into appropriately dimensioned terms $\mathcal{O}_r$ and $X\Pi_{U^\perp}$, we have

$$\begin{aligned}
\mathcal{O}_r^\dagger Y_{1|r}\Pi_{U^\perp}(X\Pi_{U^\perp})^\dagger &= \mathcal{O}_r^\dagger \mathcal{O}_r A X \Pi_{U^\perp}(X\Pi_{U^\perp})^\dagger \\
&= \mathcal{O}_r^\dagger \mathcal{O}_r A (\mathcal{C}_l U_p \Pi_{U^\perp})(\mathcal{C}_l U_p \Pi_{U^\perp})^\dagger \\
&= \mathcal{O}_r^\dagger H_r' U_p \Pi_{U^\perp}(\mathcal{C}_l U_p \Pi_{U^\perp})^\dagger \\
&= \mathcal{O}_r^\dagger H_r' \mathcal{C}_l^\dagger \\
&= A
\end{aligned}$$

with A in the same state basis as $\mathcal{O}_r$. Our goal is to construct a similar method for use in the non-deterministic case. Adding non-deterministic effects, however, will

again cause $Y_{0|r-1}\Pi_{U^\perp}$ to have full rank, and we must again produce a rank-reduced estimate of $Y_{0|r-1}\Pi_{U^\perp}$. From the SVD

$$Y_{0|r-1}\Pi_{U^\perp} = \begin{bmatrix} U_n & U_s \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_s \end{bmatrix} \begin{bmatrix} V_n^T \\ V_s^T \end{bmatrix},$$

we choose as a factorization

$$\hat{Y}_{0|r-1}\Pi_{U^\perp} = \hat{\mathcal{O}}_r \hat{X}\Pi_{U^\perp}$$

with

$$\hat{\mathcal{O}}_r = U_n\Sigma_n^{1/2} \qquad \text{and} \qquad \hat{X}\Pi_{U^\perp} = \Sigma_n^{1/2}V_n^T. \tag{4.18}$$

We now precisely define the identification problem. Given our estimate $\hat{\mathcal{O}}_r$, we wish to find an optimal solution for the one-step propagation of the state, that is, we wish to solve

$$\min_{\hat{A}} \left|\left| \mathcal{O}_r \hat{A} X\Pi_{U^\perp} - \mathcal{O}_r A X\Pi_{U^\perp} \right|\right|_F.$$

Since we do not know $\mathcal{O}_r$ and $X\Pi_{U^\perp}$ exactly, we use the estimates from the SVD, and since

$$\mathcal{O}_r A X\Pi_{U^\perp} = Y_{1|r}\Pi_{U^\perp} - V_{1|r}\Pi_{U^\perp},$$

we replace $\mathcal{O}_r A X\Pi_{U^\perp}$ with $Y_{1|r}\Pi_{U^\perp}$. The estimate for A is then found by solving

$$\min_{\hat{A}} \left|\left| \hat{\mathcal{O}}_r \hat{A} \hat{X}\Pi_{U^\perp} - Y_{1|r}\Pi_{U^\perp} \right|\right|_F, \tag{4.19}$$

the solution of which is

$$\hat{A} = \hat{\mathcal{O}}_r^\dagger Y_{1|r}\Pi_{U^\perp}(\hat{X}\Pi_{U^\perp})^\dagger. \tag{4.20}$$

Two questions arise for this factorization: do the estimates $\hat{\mathcal{O}}_r$ and $\hat{X}\Pi_{U^\perp}$ converge to true possible values of $\mathcal{O}_r$ and $X\Pi_{U^\perp}$, and does the term $\hat{\mathcal{O}}_r^\dagger V_{1|r}(\hat{X}\Pi_{U^\perp})^\dagger$ vanish?

We answer the second question first. Suppose $\mathcal{O}_r$ and $X\Pi_{U^\perp}$ are known exactly. Then

$$\begin{aligned} \hat{A} &= \hat{\mathcal{O}}_r^\dagger \mathcal{O}_r A X\Pi_{U^\perp}(\hat{X}\Pi_{U^\perp})^\dagger + \hat{\mathcal{O}}_r^\dagger V_{1|r}(\hat{X}\Pi_{U^\perp})^\dagger \\ &= A + \mathcal{O}_r^\dagger V_{1|r}(X\Pi_{U^\perp})^\dagger. \end{aligned} \tag{4.21}$$

Expanding the noise term, we find

$$\begin{aligned}\mathcal{O}_r^\dagger V_{1|r}(X\Pi_{U^\perp})^\dagger &= \mathcal{O}_r^\dagger V_{1|r}(X\Pi_{U^\perp})^T\left((X\Pi_{U^\perp})(X\Pi_{U^\perp})^T\right)^{-1}\\ &= \mathcal{O}_r^\dagger V_{1|r}\Pi_{U^\perp}X^T(\cdots)^{-1}\\ &= \mathcal{O}_r^\dagger\left(V_{1|r}X^T - V_{1|r}U_{0|r}^T\left(U_{0|r}U_{0|r}^T\right)^{-1}X^T\right)(\cdots)^{-1}.\end{aligned}$$

Because the noise and input are independent, the noise and state are also independent; therefore

$$\cancelto{0}{V_{1|r}X^T} - \cancelto{0}{V_{1|r}U_{0|r}^T}\left(U_{0|r}U_{0|r}^T\right)^{-1}X^T$$

as the column dimension l increases. Hence the noise term in (4.21) vanishes.

To answer the first question, suppose $v(t)$ is generated by the LTI system

$$\begin{aligned}x_v(t+1) &= A_v x_v(t) + B_v e_v(t)\\ v(t) &= C_v x_v(t) + D_v e_v(t),\end{aligned}$$

where $e(t)$ is multi-dimensional white noise of some constant variance. The original state-space system (4.1) will then have an equivalent representation

$$\begin{aligned}\begin{bmatrix}x(t+1)\\ x_v(t+1)\end{bmatrix} &= \begin{bmatrix}A & 0\\ 0 & A_v\end{bmatrix}\begin{bmatrix}x(t)\\ x_v(t)\end{bmatrix} + \begin{bmatrix}B & B_v\end{bmatrix}\begin{bmatrix}u(t)\\ e_v(t)\end{bmatrix}\\ y(t) &= \begin{bmatrix}C & C_v\end{bmatrix}\begin{bmatrix}x(t)\\ x_v(t)\end{bmatrix} + \begin{bmatrix}D & D_v\end{bmatrix}\begin{bmatrix}u(t)\\ e_v(t)\end{bmatrix}.\end{aligned} \tag{4.22}$$

This will have a completely controllable and observable representation

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) + Ke(t)\\ y(t) &= Cx(t) + Du(t) + Le(t)\end{aligned} \tag{4.23}$$

with minimal order n if and only if A and A_v have the same eigenvalues. Otherwise, the row space of $\hat{\mathcal{O}}_r$ will contain elements of both CA^k and $C_vA_v^k$, but only the states corresponding to CA^k will be controllable. Hence the estimates $\hat{\mathcal{O}}_r$ and $\hat{X}\Pi_{U^\perp}$ will only converge to true possible values of $\mathcal{O}_r$ and $X\Pi_{U^\perp}$ if $v(t)$ is ether white or generated by a linear process with the same poles as the deterministic system. If the noise process has different poles than the system, the uncontrollable states from the noise process will be mistakenly identified as controllable in the final model. This is demonstrated by the following example.

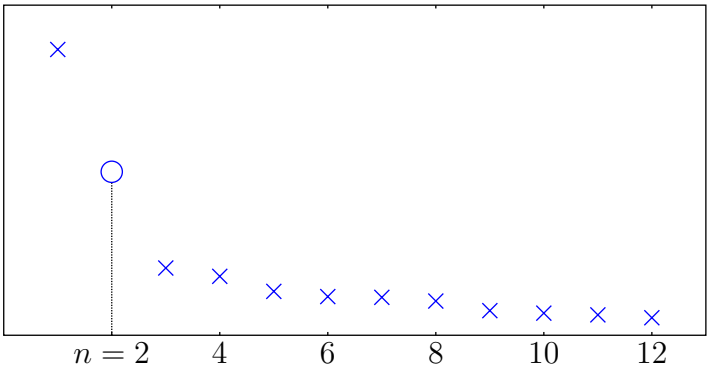

Figure 4.1: Example singular values with white process noise.

Example 4.1. Consider the system with state-space parameters

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} 0.7 & 1 & 0 \\ -0.16 & 0.7 & 1 \\ \hline -0.1247 & 1.2469 & 0 \end{array}\right]$$

where $e(t) \in \mathbb{R}^2$ and $w(t) \in \mathbb{R}$ are independent zero-mean white noise signals with

$$\operatorname{cov}(e(t)) = 0.1 I_2 \qquad \text{and} \qquad \operatorname{var}(w(t)) = 1.$$

Let $r = 12$. The singular values of $Y_{0|r-1}\Pi_{U^\perp}$ are shown in Figure 4.1. It is clear that there is a drop-off after the 2^{nd} singular value, after which the values flatten out, suggesting that the system is indeed 2^{nd}-order.

Now consider an alternative noise process

$$\begin{aligned} x_v(t+1) &= A_v x_v(t) + B_v e_v(t) \\ v(t) &= C_v x_v(t) + D_v e_v(t) \end{aligned}$$

$$\left[\begin{array}{c|c} A_v & B_v \\ \hline C_v & D_v \end{array}\right] = \left[\begin{array}{cc|c} 0.9 & 1 & 0 \\ -0.04 & 0.9 & 0.25 \\ \hline 0.2 & 0 & 0 \end{array}\right]$$

The singular values of the new system are shown in Figure 4.2. There is now a drop off after the 4^{th} singular value, even though the order of the deterministic subsystem has not changed.

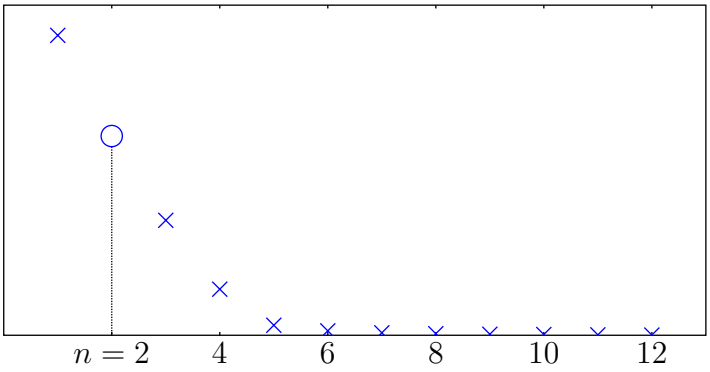

Figure 4.2: Example singular values with colored output noise.

In fact, if a fourth-order A is estimated using (4.17), the eigenvalues from A and A_v will exist in the estimate. Thus all observable states of the combined deterministic and non-deterministic systems are estimated. Unfortunately, there is no way to distinguish between controllable and uncontrollable states, and so the uncontrollable states of the noise-generating will be identified as controllable states of the deterministic system.

From this example, we see that the factorization (4.18) will only result in unbiased estimates of $\mathcal{O}_r$ and $X\Pi_{U^\perp}$ if the system is limited to process noise, that is, if all eigenvalues of A_v are eigenvalues of A. Though it is likely that the estimate of B will result in transmission zeros being placed close to the eigenvalues of A_v, it is impossible for them to cancel exactly.

To ensure that the factorization converges to a valid mapping from past input to future output, we apply an instrumental-variable technique. In the next chapter, we show that this difficulty may be overcome in a simpler, numerically efficient way by instead applying the realization procedure to covariance function estimates.

4.4.1 Obtaining Unbiased Estimates

To obtain an unbiased estimate of the row space of $\mathcal{O}_r$, we employ an auxiliary signal $z(t) \in \mathbb{R}^{n_z}$ that is uncorrelated with the noise signal $v(t)$, but

correlated with $u(t)$; more specifically, $z(t)$ satisfies

$$E\left[v(t)z^T(t+\tau)\right] = 0 \quad \forall\ \tau \tag{4.24}$$

and

$$E\left[u(t)z^T(t+\tau)\right] \neq 0 \quad \text{for some } |\tau| < r. \tag{4.25}$$

Potential candidates for $z(t)$ include a reference signal if experiments are conducted in closed-loop operation or the input signal shifted backward by a sufficient number of samples if experiments are conducted in open-loop operation.

From this signal $z(t)$, construct a block-Hankel data matrix of p rows,

$$Z_{0|p-1} = \begin{bmatrix} z(0) & z(1) & \cdots & z(l-1) \\ z(1) & z(2) & \cdots & z(l) \\ \vdots & \vdots & & \vdots \\ z(p-1) & z(p) & \cdots & z(p+l-2) \end{bmatrix}. \tag{4.26}$$

From (4.24), $Z_{0|p-1}$ has the property

$$\frac{1}{l} V_{0|r-1} Z_{0|p-1}^T \to 0_{rn_y \times pn_z},$$

and therefore

$$\frac{1}{l} V_{0|r-1} \Pi_{U^\perp} Z_{0|p-1}^T = \frac{1}{l} V_{0|r-1} Z_{0|p-1}^T + \frac{1}{l} V_{0|r-1} U_{0|r}^T \left(U_{0|r} U_{0|r}^T\right)^{-1} U_{0|r} Z_{0|p-1}^T \to 0$$

as $l \to \infty$. From (4.25), $Z_{0|p-1}$ also has the property

$$\frac{1}{l} U_{0|r} Z_{0|p-1}^T \neq 0_{(r+1)n_u \times pn_z}.$$

Thus multiplication of (4.13) on the right by $\frac{1}{l} Z_{0|p-1}^T$ results in

$$\begin{aligned} \frac{1}{l} Y_{0|r-1} \Pi_{U^\perp} Z_{0|p-1}^T &= \frac{1}{l} \mathcal{O}_r X \Pi_{U^\perp} Z_{0|p-1}^T + \frac{1}{l} V_{0|r-1} \Pi_{U^\perp} Z_{0|p-1}^T \\ &\to \frac{1}{l} \mathcal{O}_r X \Pi_{U^\perp} Z_{0|p-1}^T, \end{aligned}$$

and the row space of $\frac{1}{l} Y_{0|r-1} \Pi_{U^\perp} Z_{0|p-1}^T$ asymptotically approaches $\mathcal{O}_r$. Though we have established that this alone is sufficient to provide unbiased estimates,

additional de-correlation of the noise may be performed by also multiplying (4.14) on the right by $\frac{1}{l}Z_{0|p-1}^T$ so that

$$\begin{aligned}\frac{1}{l}Y_{1|r}\Pi_{U^\perp}Z_{0|p-1}^T &= \frac{1}{l}\mathcal{O}_r AX\Pi_{U^\perp}Z_{0|p-1}^T + \frac{1}{l}V_{1|r}\Pi_{U^\perp}Z_{0|p-1}^T \\ &\to \frac{1}{l}\mathcal{O}_r AX\Pi_{U^\perp}Z_{0|p-1}^T\end{aligned}$$

as $l \to \infty$. By instead taking the SVD

$$Y_{0|r-1}\Pi_{U^\perp}Z_{0|p-1}^T = \begin{bmatrix}U_n & U_s\end{bmatrix}\begin{bmatrix}\Sigma_n & 0\\ 0 & \Sigma_s\end{bmatrix}\begin{bmatrix}V_n^T\\ V_s^T\end{bmatrix}$$

we solve for $\hat{A}$ as

$$\hat{A} = \Sigma_n^{-1/2}U_n^T Y_{1|r}\Pi_{U^\perp}Z_{0|p-1}^T V_n\Sigma_n^{-1/2}, \tag{4.27}$$

which will converge to the strictly deterministic subsystem, so long as our assumptions on $z(t)$ are valid. Thus by using an external weighting matrix $Z_{0|p-1}$, we can achieve unbiased estimates even when the poles of the noise model are not contained in the deterministic subsystem.

4.5 Asymptotic Properties of the Estimate

In this section, asymptotic properties of the system estimate are derived and analyzed. It is shown that variance the estimate of the row space of $\mathcal{O}_r$ is a function of a block-Toeplitz matrix of cross-covariance function data.

4.5.1 Asymptotic Properties of the Data Matrices

We first analyze properties of the projected and weighted data matrix equations, repeated here:

$$\frac{1}{l}Y_{0|r-1}\Pi_{U^\perp}Z_{0|p-1}^T = \frac{1}{l}\mathcal{O}_r X\Pi_{U^\perp}Z_{0|p-1}^T + \frac{1}{l}V_{0|r-1}\Pi_{U^\perp}Z_{0|p-1}^T \tag{4.28}$$

$$\frac{1}{l}Y_{1|r}\Pi_{U^\perp}Z_{1|p}^T = \frac{1}{l}\mathcal{O}_r AX\Pi_{U^\perp}Z_{1|p}^T + \frac{1}{l}V_{1|r}\Pi_{U^\perp}Z_{1|p}^T. \tag{4.29}$$

The left side of (4.28) expands to

$$\frac{1}{l}Y_{0|r-1}\left(I_l - U_{0|r}^T(U_{0|r}U_{0|r}^T)^{-1}U_{0|r}\right)Z_{0|r-1}^T$$
$$= \frac{1}{l}Y_{0|r-1}Z_{0|r-1}^T - \left(\frac{1}{l}Y_{0|r-1}U_{0|r}^T\right)\left(\frac{1}{l}U_{0|r}U_{0|r}^T\right)^{-1}\left(\frac{1}{l}U_{0|r}Z_{0|r-1}^T\right).$$

From Lemma B.1, the first term on the right-hand side expands to

$$\frac{1}{l}Y_{0|r-1}Z_{0|r-1}^T = \begin{bmatrix} \frac{1}{l}\sum_{k=0}^{l-1} y(k)z^T(k) & \cdots & \frac{1}{l}\sum_{k=0}^{l-1} y(k)z^T(k+r-1) \\ \frac{1}{l}\sum_{k=1}^{l} y(k)z^T(k-1) & \cdots & \frac{1}{l}\sum_{k=1}^{l} y(k)z^T(k+r-2) \\ \vdots & & \vdots \\ \frac{1}{l}\sum_{k=r-1}^{l+r-1} y(k)z^T(k-r+1) & \cdots & \frac{1}{l}\sum_{k=r-1}^{l+r-1} y(k)z^T(k) \end{bmatrix},$$

and taking the expectation as $l \to \infty$ results in the block-Toeplitz matrix

$$E\lim_{l\to\infty}\frac{1}{l}Y_{0|r-1}Z_{0|r-1}^T = \begin{bmatrix} R_{yz}(0) & R_{yz}(-1) & \cdots & R_{yz}(-r+1) \\ R_{yz}(1) & R_{yz}(0) & \cdots & R_{yz}(-r+2) \\ \vdots & \vdots & & \vdots \\ R_{yz}(r-1) & R_{yz}(r-2) & \cdots & R_{yz}(0) \end{bmatrix}.$$

Similarly,

$$E\lim_{l\to\infty}\frac{1}{l}Y_{0|r-1}U_{0|r}^T = \begin{bmatrix} R_{yu}(0) & R_{yu}(-1) & \cdots & R_{yu}(-r) \\ R_{yu}(1) & R_{yu}(0) & \cdots & R_{yu}(-r+1) \\ \vdots & \vdots & & \vdots \\ R_{yu}(r-1) & R_{yu}(r-2) & \cdots & R_{yu}(1) \end{bmatrix},$$

$$E\lim_{l\to\infty}\frac{1}{l}U_{0|r}U_{0|r}^T = \begin{bmatrix} R_u(0) & R_u(-1) & \cdots & R_u(-r) \\ R_u(1) & R_u(0) & \cdots & R_u(-r+1) \\ \vdots & \vdots & & \vdots \\ R_u(r) & R_u(r-1) & \cdots & R_u(0) \end{bmatrix}, \tag{4.30}$$

and

$$E\lim_{l\to\infty}\frac{1}{l}U_{0|r}Z_{0|r-1}^T = \begin{bmatrix} R_{uz}(0) & R_{uz}(-1) & \cdots & R_{uz}(-r+1) \\ R_{uz}(1) & R_{uz}(0) & \cdots & R_{uz}(-r+2) \\ \vdots & & \vdots & \\ R_{uz}(r) & R_{uz}(r-1) & \cdots & R_{uz}(1) \end{bmatrix}$$

In Ljung [40], one test for persistency of excitation of an input signal is that (4.30) be invertible. (This might not be true for some signals, such as a sinusoid.) This is also clearly required for the data matrix equation to converge.

If we choose $Z_{0|r-1} = U_{-r|-1}$, then

$$E \lim_{l\to\infty} \frac{1}{l} Y_{0|r-1} U_{-r|-1}^T = \begin{bmatrix} R_{yu}(r) & R_{yu}(r-1) & \cdots & R_{yu}(1) \\ R_{yu}(r+1) & R_{yu}(r) & \cdots & R_{yu}(2) \\ \vdots & \vdots & & \vdots \\ R_{yu}(2r-1) & R_{yu}(2r-2) & \cdots & R_{yu}(r) \end{bmatrix}$$

and

$$E \lim_{l\to\infty} \frac{1}{l} U_{0|r} U_{-r|-1}^T = \begin{bmatrix} R_u(r) & R_u(r-1) & \cdots & R_u(1) \\ R_u(r+1) & R_u(r) & \cdots & R_u(2) \\ \vdots & \vdots & & \vdots \\ R_u(2r) & R_u(2r-1) & \cdots & R_u(r+1) \end{bmatrix}.$$

In this case, the projected data matrix equation is the Schur complement of the block-Toeplitz matrix

$$E \lim_{l\to\infty} \frac{1}{l} \begin{bmatrix} Y_{0|r-1} \\ U_{0|r} \end{bmatrix} U_{-r|r}^T =$$

$$\left[\begin{array}{ccc|ccc} R_{yu}(r) & \cdots & R_{yu}(1) & R_{yu}(0) & \cdots & R_{yu}(-r) \\ \vdots & & \vdots & \vdots & & \vdots \\ R_{yu}(2r-1) & \cdots & R_{yu}(r) & R_{yu}(r-1) & \cdots & R_{yu}(1) \\ \hline R_u(r) & \cdots & R_u(1) & R_u(0) & \cdots & R_u(-r) \\ \vdots & & \vdots & \vdots & & \vdots \\ R_u(2r) & \cdots & R_u(r+1) & R_u(r) & \cdots & R_u(0) \end{array}\right]$$

4.5.2 Asymptotic Properties of the Extended Observability Matrix

Several results exist in the literature regarding asymptotic properties of the extended observability matrix estimate $\hat{\mathcal{O}}_r$ for different subspace methods. The most useful analyses, i.e. the ones that do not rely on computing decompositions

of matrices that become asymptotically infinite-dimensional, are based on the idea of projecting the estimate $\hat{\mathcal{O}}_r$ onto the null space of the true $\mathcal{O}_r$. Let

$$\Pi_{\mathcal{O}} = \mathcal{O}_r \left(\mathcal{O}_r^T \mathcal{O}_r\right)^{-1} \mathcal{O}_r^T$$

be the orthogonal projector onto the row space of the true $\mathcal{O}_r$ and

$$\Pi_{\mathcal{O}^\perp} = I_{n_y r} - \Pi_{\mathcal{O}}$$

be the projector onto its null space. The estimate $\hat{\mathcal{O}}_r$ may then be decomposed into two parts as

$$\hat{\mathcal{O}}_r = \Pi_{\mathcal{O}} \hat{\mathcal{O}}_r + \Pi_{\mathcal{O}^\perp} \hat{\mathcal{O}}_r.$$

The term $\Pi_{\mathcal{O}^\perp} \hat{\mathcal{O}}_r$ represents the estimation error. In the following derivations, subscripts for the data matrices are omitted to conserve space; hence $Y = Y_{0|r-1}$, $V = V_{0|r-1}$, and $U = U_{0|r}$. From

$$\frac{1}{l} Y \Pi_{U^\perp} Z^T = \underbrace{U_n \Sigma_n^{1/2}}_{\hat{\mathcal{O}}_r} \Sigma_n^{1/2} V_n^T + U_s \Sigma_s V_s^T,$$

multiplication on the right by $V_n \Sigma_n^{-1/2}$ produces

$$\frac{1}{l} Y \Pi_{U^\perp} Z^T V_n \Sigma_n^{-1/2} = \hat{\mathcal{O}}_r.$$

Hence

$$\begin{aligned} \Pi_{\mathcal{O}^\perp} \hat{\mathcal{O}}_r &= \frac{1}{l} \Pi_{\mathcal{O}^\perp} Y \Pi_{U^\perp} Z^T V_n \Sigma_n^{-1/2} \\ &= \frac{1}{l} \Pi_{\mathcal{O}^\perp} \left(\mathcal{O}_r X \Pi_{U^\perp} Z^T + V \Pi_{U^\perp} Z^T\right) V_n \Sigma_n^{-1/2} \\ &= \frac{1}{l} \Pi_{\mathcal{O}^\perp} V \Pi_{U^\perp} Z^T V_n \Sigma_n^{-1/2} \end{aligned} \tag{4.31}$$

Results for variance of $\Pi_{\mathcal{O}^\perp} \hat{\mathcal{O}}_r$ using an alternative definition $\hat{\mathcal{O}}_r = U_n$ may be found in Viberg, Wahlberg, and Ottersten [80]. This was extended by Gustafsson [23] to find optimal right-hand column weightings when the system is constrained to real eigenvalues.

Though results exist in the literature for the variance of (4.31), the variance of the system estimate is, however, determined by the pseudoinverse $\hat{\mathcal{O}}_r^\dagger$ for both

realization-based and MOESP-type algorithms. The only existing analysis for $\hat{\mathcal{O}}_r^\dagger$ assumes that all system eigenvalues area real and that $\hat{\mathcal{O}}_r$ has been modified to correspond to a state basis with diagonal A [80]. In the following, we present a direct computation of the variance of $\hat{\mathcal{O}}_r^\dagger$.

The pseudoinverse of $\hat{\mathcal{O}}_r$ may be found from

$$\begin{aligned} U_n\Sigma_n V_n^T + U_s\Sigma_s V_s^T &= \frac{1}{l}\mathcal{O}_r X\Pi_{U^\perp}Z^T + \frac{1}{l}V\Pi_{U^\perp}Z^T \\ U_n\Sigma_n^{1/2} &= \frac{1}{l}\mathcal{O}_r X\Pi_{U^\perp}Z^T V_n\Sigma_n^{-1/2} + \frac{1}{l}V\Pi_{U^\perp}Z^T V_n\Sigma_n^{-1/2} \\ \Sigma_n^{1/2}U_n^T &= \frac{1}{l}\Sigma_n^{-1/2}V_n^T Z\Pi_{U^\perp}X^T\mathcal{O}_r^T + \frac{1}{l}\Sigma_n^{-1/2}V_n^T Z\Pi_{U^\perp}V^T \\ \hat{\mathcal{O}}_r^\dagger = \Sigma_n^{-1/2}U_n^T &= \frac{1}{l}\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}X^T\mathcal{O}_r^T + \frac{1}{l}\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}V^T. \end{aligned}$$

From this we see that the error in $\hat{\mathcal{O}}_r$ may be found by projecting it on the right onto the null space of $\mathcal{O}_r$. Thus the asymptotic properties of $\hat{\mathcal{O}}_r^\dagger$ may be found by analyzing the term

$$\hat{\mathcal{O}}_r^\dagger\Pi_{\mathcal{O}^\perp} = \frac{1}{l}\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}V^T\Pi_{\mathcal{O}^\perp} \tag{4.32}$$

as $l \to \infty$.

Several conclusions may be drawn from (4.32). Clearly the state basis will have an effect on the overall error of the estimate, since a change of basis $T^{-1}AT$ in (4.32) is equivalent to multiplication on the left by $\Sigma_n^{-1}T\Sigma_n$.

The following theorem provides an expression for the variance of the error in $\hat{\mathcal{O}}_r$.

Theorem 4.1. *Let the total error of the estimate be defined as*

$$\epsilon = \frac{1}{l}\mathrm{vec}\left(\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}V^T\Pi_{\mathcal{O}^\perp}\right).$$

Let

$$\begin{bmatrix} Z \\ U \end{bmatrix} = M = \begin{bmatrix} m(0) & m(1) & \cdots & m(l-1) \\ m(1) & m(2) & \cdots & m(l) \\ \vdots & \vdots & & \vdots \\ m(i-1) & m(i-2) & \cdots & m(i+l-2) \end{bmatrix}.$$

Then

$$Var(\epsilon) = \lim_{l\to\infty}\frac{1}{l}E\left[\frac{1}{l^2}\left(\Pi_{\mathcal{O}^\perp}\otimes V_n\Sigma_n^{-3/2}\right)P\left(\Pi_{\mathcal{O}^\perp}\otimes\Sigma_n^{-3/2}V_n^T\right)\right],$$

where

$$P = \sum_{j=0}^{l-1}\sum_{k=0}^{l-1} P_m(j,\ k) \otimes P_v(j,\ k),$$

$$P_m(j,\ k) = \frac{1}{l}\begin{bmatrix} m(j) \\ m(j+1) \\ \vdots \\ m(j+i-1) \end{bmatrix}\begin{bmatrix} m(k) \\ m(k+1) \\ \cdots \\ m(k+i-1) \end{bmatrix}^T$$

$$P_v(j,\ k) = \frac{1}{l}\begin{bmatrix} v(j) \\ v(j+1) \\ \vdots \\ v(j+r-1) \end{bmatrix}\begin{bmatrix} v(k) \\ v(k+1) \\ \cdots \\ v(k+r-1) \end{bmatrix}^T.$$

Proof. We know $E[\epsilon] = 0$ since $E \lim_{l\to\infty} \frac{1}{l}\left[V\Pi_{U^\perp}Z^T\right] = 0$. Thus

$$\begin{aligned}\mathrm{Var}(\epsilon) &= E \lim_{l\to\infty} \frac{1}{l}[\epsilon\epsilon^T] \\ &= E \lim_{l\to\infty}\left[\frac{1}{l^2}\mathrm{vec}\left(\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}V^T\Pi_{\mathcal{O}^\perp}\right)\mathrm{vec}\left(\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}V^T\Pi_{\mathcal{O}^\perp}\right)^T\right].\end{aligned}$$

From

$$\mathrm{vec}\left(\Sigma_n^{-3/2}V_n^T Z\Pi_{U^\perp}V^T\Pi_{\mathcal{O}^\perp}\right) = \left[\Pi_{\mathcal{O}^\perp}\otimes V_n\Sigma_n^{-3/2}\right]\mathrm{vec}(Z\Pi_{U^\perp}V^T),$$

we have

$$\epsilon\epsilon^T = \frac{1}{l^2}\left[\Pi_{\mathcal{O}^\perp}\otimes V_n\Sigma_n^{-3/2}\right]\mathrm{vec}(Z\Pi_{U^\perp}V^T)\mathrm{vec}(Z\Pi_{U^\perp}V^T)^T\left[\Pi_{\mathcal{O}^\perp}\otimes \Sigma_n^{-3/2}V_n^T\right].$$

Also,

$$\begin{aligned} Z\Pi_{U^\perp}V^T &= Z\left[I_l - U^T(UU^T)^{-1}\right]V^T \\ &= ZV^T - ZU^T(UU)^{-1}V^T \\ &= \begin{bmatrix} I_r & -ZU^T(UU)^{-1}\end{bmatrix}\begin{bmatrix} ZV^T \\ UV^T\end{bmatrix},\end{aligned}$$

resulting in

$$\mathrm{vec}\left(Z\Pi_{U^\perp}V^T\right) = \left(I_r\otimes\begin{bmatrix} I_r & -ZU^T(UU)^{-1}\end{bmatrix}\right)\mathrm{vec}\left(\begin{bmatrix} ZV^T \\ UV^T\end{bmatrix}\right).$$

From Theorem B.1,

$$\text{vec}(MV^T)\text{vec}(MV^T)^T = \sum_{j=0}^{l-1}\sum_{k=0}^{l-1} P_m(j,\ k) \otimes P_v(j,\ k) = P$$

Thus

$$\epsilon\epsilon^T = \frac{1}{l^2}\left(\Pi_{\mathcal{O}^\perp} \otimes V_n\Sigma_n^{-3/2}\right) P \left(\Pi_{\mathcal{O}^\perp} \otimes \Sigma_n^{-3/2}V_n^T\right),$$

which is the required result. □

From this, we can see that variance of the error has some asymptotic dependence on the summation of block-Toeptliz matrices of auto-covariance function data. This results is similar to the one achieved in Gustafsson [23]. Unfortunately, statistical dependence between the noise signal $v(t)$ and the right-hand singular vectors V_n^T prevent this expression from being simplified to a more useful result.

4.6 Least-Squares Estimation of Input Parameters from Input-Output Data

If A and C are known, then B, D, and the initial state $x(0)$ are linear in the input-output data and may be identified by solving a linear-least-squares problem. Writing the output as

$$y(t) = CA^t x(0) + \sum_{k=0}^{t-1} CA^{t-k-1}Bu(k) + Du(t) + v(t) \tag{4.33}$$

and using the Kronecker-product identity

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X)$$

results in factoring out B and D on the right as

$$y(t) = CA^t x(0) + \left(\sum_{k=0}^{t-1} u^T(k) \otimes CA^{t-k-1}\right)\text{vec}(B) + \left(u^T(t) \otimes I_{n_y}\right)\text{vec}(D) + v(t).$$

Thus, given estimates $\hat{A}$ and $\hat{C}$, the identification of B, D, and $x(0)$ from a sequence of N data points may be stated as a linear-least-squares problem

$$\left(\hat{B},\ \hat{D},\ \hat{x}(0)\right) = \underset{B,\ D,\ x(0)}{\arg\min} \left|\left|y - \hat{y}\right|\right|_2, \tag{4.34}$$

where

$$y = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} \in \mathbb{R}^{n_y N}, \qquad \hat{y} = \begin{bmatrix} \hat{y}(0) \\ \hat{y}(1) \\ \vdots \\ \hat{y}(N-1) \end{bmatrix} \in \mathbb{R}^{n_y N},$$

$$\hat{y}(t) = \begin{bmatrix} \phi_{x_0}^T(t) & \phi_B^T(t) & \phi_D^T(t) \end{bmatrix} \begin{bmatrix} x(0) \\ \mathrm{vec}(B) \\ \mathrm{vec}(D) \end{bmatrix} = \phi^T(t)\theta,$$

and

$$\phi_{x_0}^T(t) = \hat{C}\hat{A}^t \qquad \in \mathbb{R}^{n_y \times n}, \tag{4.35}$$

$$\phi_B^T(t) = \sum_{k=0}^{t-1} u^T(k) \otimes \hat{C}\hat{A}^{t-k-1} \qquad \in \mathbb{R}^{n_y \times n_u n}, \tag{4.36}$$

$$\phi_D^T(t) = u^T(t) \otimes I_{n_y} \qquad \in \mathbb{R}^{n_y \times n_u n_y}. \tag{4.37}$$

In practice, a significant computational limitation is the calculation of the regressor $\phi_B^T(t)$. The Kronecker product in (4.36) results in calculation and summation of $O(N^3/2)$ matrices of size $n_y \times n_u n$. Memory limitations will dramatically limit the maximum N that may be used for computing $\phi^T(t)$, particularly for multivariable data of large dimension. As we will see in Chapter 5, this problem is compounded when the method is extended to matrix-valued signals, such as covariance functions.

We now present a numerically efficient method of computing (4.36) by means of computing the state sequence of a dual system. This allows for a dramatic reduction in the memory and computation time required to solve the linear-least-squares problem. We begin by first reformulating the calculation of the regressor for the raw data case and then extend to the method to covariance function estimates. To avoid calculating $\phi_B(t)^T$ from (4.36) explicitly, we show that $\phi_B(t)^T$ may be calculated more efficiently as a set of state sequences of a dual system.

Theorem 4.2. *Block element (i, j) of the transposed regressor $\phi_B(t)$ in (4.36) is equivalent to the state sequence of the system*

$$\phi_B^{(i,j)}(t+1) = \hat{A}^T \phi_B^{(i,j)}(t) + \hat{C}^T \tilde{u}(i, j, t) \tag{4.38}$$

computed with the initial condition $\phi_B(0) = 0_n$*, in which*

$$\tilde{u}(i,j,t) = \begin{bmatrix} 0_{j-1} \\ u_i(t) \\ 0_{n_y - j} \end{bmatrix}.$$

Proof. First, observe that the transpose of (4.36) may be expanded as

$$\begin{aligned}
\phi_B(t) &= \sum_{k=0}^{t-1} \left(u^T(k) \otimes \hat{C}\hat{A}^{t-k-1} \right)^T \\
&= \sum_{k=0}^{t-1} \left[\left(u^T(k) \otimes I_{n_y} \right) \left(I_{n_u} \otimes \hat{C}\hat{A}^{t-k-1} \right) \right]^T \\
&= \sum_{k=0}^{t-1} \left(I_{n_u} \otimes \hat{C}\hat{A}^{t-k-1} \right)^T \left(u^T(k) \otimes I_{n_y} \right)^T \\
&= \sum_{k=0}^{t-1} \left(I_{n_u} \otimes \left(\hat{A}^T \right)^{t-k-1} \hat{C}^T \right) \left(u(k) \otimes I_{n_y} \right).
\end{aligned}$$

The first Kronecker product within the summation expands to the block-diagonal matrix

$$\begin{bmatrix} \left(\hat{A}^T \right)^{t-k-1} \hat{C}^T & 0 & \cdots \\ 0 & \left(\hat{A}^T \right)^{t-k-1} \hat{C}^T & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix},$$

and multiplication on the right by $u(k) \otimes I_{n_y}$ results in

$$\begin{bmatrix} \left(\hat{A}^T \right)^{t-k-1} \hat{C}^T u_1(k) \\ \left(\hat{A}^T \right)^{t-k-1} \hat{C}^T u_2(k) \\ \vdots \end{bmatrix}, \tag{4.39}$$

where $u_i(k)$ is the i-th component of the input signal $u(t)$. Because $u_i(k)$ is scalar,

$u_i(k) \otimes I_{n_y} = u_i(k) I_{n_y}$, and we may reincorporate the summation to find

$$\phi_B(t) = \begin{bmatrix} \sum_{k=0}^{t-1} \left(\hat{A}^T\right)^\eta \hat{C}^T \begin{bmatrix} u_1(k) \\ 0 \\ 0 \\ \vdots \end{bmatrix} & \sum_{k=0}^{t-1} \left(\hat{A}^T\right)^\eta \hat{C}^T \begin{bmatrix} 0 \\ u_1(k) \\ 0 \\ \vdots \end{bmatrix} & \cdots \\ \sum_{k=0}^{t-1} \left(\hat{A}^T\right)^\eta \hat{C}^T \begin{bmatrix} u_2(k) \\ 0 \\ 0 \\ \vdots \end{bmatrix} & \sum_{k=0}^{t-1} \left(\hat{A}^T\right)^\eta \hat{C}^T \begin{bmatrix} 0 \\ u_2(k) \\ 0 \\ \vdots \end{bmatrix} & \cdots \\ \vdots & \vdots & \end{bmatrix}$$

where $\eta = t - k - 1$. This is a convolution operation, similar to (4.33), with $\hat{A}^T$ in place of A, $\hat{C}^T$ in place of B, and I_n in place of C. Hence, $\phi_B^T(t)$ may be calculated as separate convolutions of significantly smaller dimension. □

Typically, the fastest way to compute ϕ_B^T will be to form the $n_u n_y$ input signals $\tilde{u}(i, j, \tau)$ and to compute state-sequences for each using (4.38). Efficient routines for computing state-sequences of linear, time-invariant systems of this type are commonly available in numerical software packages, such as MATLAB. The remaining regressors $\phi_{x_0}^T$ and ϕ_D^T are straightforward and far less expensive to compute.

4.7 Summary of Input-Output Data Realization Procedure

The following steps completely describe the input-output data realization algorithm:

1. Construct the block-Hankel output data matrices $Y_{0|r-1}$ in (4.2) and $Y_{1|r}$ in (4.5) and the block-Hankel input data matrix $U_{0|r}$ in (4.7). Optionally form the block-Hankel instrument matrix $Z_{0|p-1}$ in (4.26) if the output noise is believed to be strongly colored.

2. Compute the projected matrices $Y_{0|r-1}\Pi_{U^\perp}$ and $Y_{1|r}\Pi_{U^\perp}$ via the LQ-decompositions in (4.15) and (4.16).

3. Take the SVD of $Y_{0|r-1}\Pi_{U^\perp}$ or $Y_{0|r-1}\Pi_{U^\perp}Z_{0|p-1}^T$ if weighting is used. If necessary, examine the singular values to determine the system order.

4. Solve for $\hat{A}$ from (4.20) or (4.27) if weighting is used. Take $\hat{C}$ from the first n_y rows of $U_n\ \Sigma_n^{1/2}$.

5. Estimate B, D, and $x(0)$ by solving (4.34).

4.8 Acknowledgements

This chapter, in part, includes content from the following publications:

D.N. Miller and R.A. de Callafon, "Efficient Identification of Input Dynamics for Correlation Function-Based Subspace Identification," *Proc. of the 18th World Congress of the International Federation of Automatic Control*, Milan, Italy: August 2011.

D.N. Miller and R.A. de Callafon, "Subspace Identification Using Dynamic Invariance in Shifted Time-Domain Data," *Proc. of the 49th IEEE Conference on Decision and Control*, Atlanta, GA: December 2010.

D.N. Miller and R.A. de Callafon, "Subspace Identification from Classical Realization Methods," *Proc. of the 15th IFAC Symposium on System Identification*, Saint-Malo, France: July 2009.

5 Realization from Covariance-Function Estimates

The identification framework developed in the previous section may be directly extended to alternative forms of data, notably covariance functions. A covariance-based framework allows for the estimation of unbiased state-space models, regardless of the noise spectrum. Furthermore, matrices filled with covariance function data may remain bounded in size as the number of data samples increases, whereas matrices with raw data would become infinite dimensional, making the identification problem more computationally friendly for large data sets.

5.1 Covariance Functions and State-Space Systems

We begin by formulating covariance-function data matrices in which the input is correlated with an external signal $\xi(t)$. Suppose the discrete-time LTI state-space system

$$x(t+1) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t) + v(t)$$

is perturbed by a stationary noise signal $v(t)$ and subjected to a quasi-stationary input $u(t)$. Let $\xi(t)$ be a quasi-stationary signal correlated with $u(t)$. Then the cross-covariance function

$$R_{u\xi}(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-\tau-1} Eu(t+\tau)\xi^T(t)$$

exists, and the cross-covariance function estimate

$$\hat{R}_{u\xi}(\tau) = \frac{1}{N} \sum_{t=0}^{N-\tau-1} u(t+\tau)\xi^T(t)$$

converges to $R_{u\xi}(\tau)$ with increasing N [40]. If $\xi(t)$ is uncorrelated with the noise signal $v(t)$, then

$$R_{v\xi}(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-\tau-1} Ev(t+\tau)\xi^T(t) = 0 \quad \forall \quad \tau$$

and

$$\hat{R}_{v\xi}(\tau) = \frac{1}{N} \sum_{t=0}^{N-\tau-1} v(t+\tau)\xi^T(t) \to 0$$

with increasing N. Because $u(t)$ is quasi-stationary and $v(t)$ is stationary, the cross-covariance function $R_{y\xi}(\tau)$ exists, and the various cross-covariance functions satisfy the same dynamic relationship as the raw data [40]:

$$\begin{aligned} R_{x\xi}(\tau+1) &= AR_{x\xi}(\tau) + BR_{u\xi}(\tau) \\ R_{y\xi}(\tau) &= CR_{x\xi}(\tau) + DR_{u\xi}(\tau) + \cancelto{0}{R_{v\xi}(\tau)}. \end{aligned} \tag{5.1}$$

Thus the cross-covariance functions exactly describe the strictly deterministic subsystem when $\xi(t)$ and $v(t)$ are uncorrelated. We use this property to identify system estimates that are guaranteed to converge with increasing data samples. Possible candidates for $\xi(t)$ include $u(t)$ when experiments are performed in open-loop and an external reference signal when experiments are performed in closed-loop.

5.2 Covariance-Based Realization

In this section, we directly extend the method of realization from input-output data to covariance function estimates.

5.2.1 Covariance-Based Data Matrix Equations

Suppose the covariance functions $R_{u\xi}(\tau)$ and $R_{y\xi}(\tau)$ are known over the domain of some integer sequence $\tau \in \{\tau_0,\ \tau_1, \ldots, \tau_N\}$ for which $R_{u\xi}(\tau)$ is not 0

everywhere. Let $R^{y\xi}_{\tau_0|\tau_r-1}$ be a block-Hankel matrix

$$R^{y\xi}_{\tau_0|\tau_r-1} = \begin{bmatrix} R_{y\xi}(\tau_0) & R_{y\xi}(\tau_1) & \cdots & R_{y\xi}(\tau_l-1) \\ R_{y\xi}(\tau_1) & R_{y\xi}(\tau_2) & \cdots & R_{y\xi}(\tau_l-2) \\ \vdots & \vdots & & \vdots \\ R_{y\xi}(\tau_r-1) & R_{y\xi}(\tau_r) & \cdots & R_{y\xi}(\tau_r+\tau_l-1) \end{bmatrix} \in \mathbb{R}^{n_y r \times n_\xi l},$$

let $R^{u\xi}_{\tau_0|\tau_r-1}$ be a block-Hankel matrix

$$R^{u\xi}_{\tau_0|\tau_r-1} = \begin{bmatrix} R_{u\xi}(\tau_0) & R_{u\xi}(\tau_1) & \cdots & R_{u\xi}(\tau_l-1) \\ R_{u\xi}(\tau_1) & R_{u\xi}(\tau_2) & \cdots & R_{u\xi}(\tau_l-2) \\ \vdots & \vdots & & \vdots \\ R_{u\xi}(\tau_r-1) & R_{u\xi}(\tau_r) & \cdots & R_{u\xi}(\tau_r+\tau_l-1) \end{bmatrix} \in \mathbb{R}^{n_u r \times n_\xi l},$$

and let $R^{x\xi}_{\tau_0}$ be

$$R^{x\xi}_{\tau_0} = \begin{bmatrix} R_{x\xi}(\tau_0) & R_{x\xi}(\tau_0+1) & \cdots & R_{x\xi}(\tau_l) \end{bmatrix} \in \mathbb{R}^{n \times n_\xi l}.$$

From (5.1) these three matrices satisfy the relationship

$$R^{y\xi}_{\tau_0|\tau_r-1} = \mathcal{O}_r R^{x\xi}_{\tau_0} + T_{0|r-1} R^{u\xi}_{\tau_0|\tau_r-1},$$

in which $\mathcal{O}_r$ is the familiar extended observability matrix and $T_{0|r-1}$ the block-Toeplitz matrix of Markov parameters. As with the raw-data case, an extra block row may be added to $R^{u\xi}_{\tau_0|\tau_r-1}$ to form

$$R^{y\xi}_{\tau_0|\tau_r-1} = \mathcal{O}_r R^{x\xi}_{\tau_0} + T'_{0|r-1} R^{u\xi}_{\tau_0|\tau_r}. \tag{5.2}$$

A shifted data matrix equation can also be formed using covariance function data to result in

$$R^{y\xi}_{\tau_1|\tau_r} = \mathcal{O}_r A R^{x\xi}_{\tau_0} + T_{1|r} R^{u\xi}_{\tau_0|\tau_r}. \tag{5.3}$$

Defining the projector matrix

$$\Pi_{R^\perp_u} = I_{n_u l} - (R^{u\xi}_{\tau_0|\tau_r})^T \left(R^{u\xi}_{\tau_0|\tau_r} (R^{u\xi}_{\tau_0|\tau_r})^T \right)^{-1} R^{u\xi}_{\tau_0|\tau_r},$$

we may isolate the row space of $\mathcal{O}_r$ in (5.2) and (5.3) as

$$R^{y\xi}_{\tau_0|\tau_r-1} \Pi_{R^\perp_u} = \mathcal{O}_r R^{x\xi}_{\tau_0} \Pi_{R^\perp_u}$$
$$R^{y\xi}_{\tau_1|\tau_r} \Pi_{R^\perp_u} = \mathcal{O}_r A R^{x\xi}_{\tau_0} \Pi_{R^\perp_u}.$$

Although nondeterministic effects prevent the covariance functions from being known exactly, the covariance function estimates $\hat{R}_{u\xi}(\tau)$ and $\hat{R}_{y\xi}(\tau)$ will converge to the true values so long as $u(t)$, $y(t)$, and $\xi(t)$ remain quasi-stationary. Therefore we may replace $R^{y\xi}_{\tau_0|\tau_r-1}$ and $R^{u\xi}_{\tau_0|\tau_r}$ with matrices of cross-covariance function estimates to form $\hat{R}^{y\xi}_{\tau_0|\tau_r-1}$ and $\hat{R}^{u\xi}_{\tau_0|\tau_r}$, respectively. Because only the estimates

$$\begin{aligned}\hat{R}^{y\xi}_{\tau_0|\tau_r-1}\Pi_{R_u^\perp} &\approx \mathcal{O}_r R^{x\xi}_{\tau_0}\Pi_{R_u^\perp}\\ \hat{R}^{y\xi}_{\tau_1|\tau_r}\Pi_{R_u^\perp} &\approx \mathcal{O}_r A R^{x\xi}_{\tau_0}\Pi_{R_u^\perp}\end{aligned} \tag{5.4}$$

are available, the matrices will have full rank, and once again the SVD must be employed to factor the mapping from past input to future output.

5.2.2 Factorization of the Estimate

From the SVD

$$\hat{R}^{y\xi}_{\tau_0|\tau_r-1}\Pi_{R_u^\perp} = \begin{bmatrix}U_n & U_s\end{bmatrix}\begin{bmatrix}\Sigma_n & 0\\ 0 & \Sigma_s\end{bmatrix}\begin{bmatrix}V_n^T\\ V_s^T\end{bmatrix}$$

we may again choose

$$\hat{\mathcal{O}}_r = U_n\Sigma_n^{1/2}$$

to estimate the extended observability matrix. The right singular vectors however will not be the states, but the cross-covariance of the states and $\xi(t)$ projected onto the null space of $R_{u\xi}(\tau)$:

$$\hat{R}^{x\xi}_{\tau_0}\Pi_{R_u^\perp} = \Sigma_n^{1/2}V_n^T.$$

From (5.4), we may estimate A as

$$\begin{aligned}\hat{A} &= \hat{\mathcal{O}}_r^\dagger \hat{R}^{y\xi}_{\tau_1|\tau_r}\Pi_{R_u^\perp}\left(\hat{R}^{x\xi}_{\tau_0}\Pi_{R_u^\perp}\right)^\dagger\\ &= \Sigma_n^{-1/2}U_n^T\hat{R}^{y\xi}_{\tau_1|\tau_r}\Pi_{R_u^\perp}V_n\Sigma_n^{-1/2},\end{aligned} \tag{5.5}$$

which is the solution to the minimization problem

$$\min_{\hat{A}}\left|\left|\hat{\mathcal{O}}_r\hat{A}\hat{R}^{x\xi}_{\tau_0}\Pi_{R_u^\perp} - \hat{R}^{y\xi}_{\tau_1|\tau_r}\Pi_{R_u^\perp}\right|\right|_F.$$

5.2.3 Reinterpretation Through Data Matrices

The data matrix equations of covariance function estimates (5.2) and (5.3) can alternatively be written as matrices of raw data. We will demonstrate this for a block Hankel matrix of the estimated cross-covariance functions $\hat{R}_{yu}(\tau)$; the interpretation for other data matrices is straightforward.

From

$$\hat{R}^{yu}_{\tau_0|\tau_r-1} = \begin{bmatrix} \frac{1}{N}\sum_{t=0}^{N-\tau_0-1} y(t+\tau_0)u^T(t) & \cdots & \frac{1}{N}\sum_{t=0}^{N-\tau_l-2} y(t+\tau_l-1)u^T(t) \\ \vdots & & \vdots \\ \frac{1}{N}\sum_{t=0}^{N-\tau_r-1} y(t+\tau_r-1)u^T(t) & \cdots & \frac{1}{N}\sum_{t=0}^{N-\tau_r-\tau_l+2} y(t+\tau_r+\tau_l-2)u^T(t) \end{bmatrix},$$

this can be expanded into the product

$$\hat{R}^{yu}_{\tau_0|\tau_r-1} = Y_{\tau_0|\tau_r-1}\Omega^T$$

where

$$\hat{R}^{yu}_{\tau_0|\tau_r-1} = \begin{bmatrix} y(\tau_0) & y(\tau_1) & \cdots & y(N-\tau_0-1) \\ y(\tau_1) & y(\tau_2) & \cdots & y(N-\tau_0) \\ \vdots & \vdots & & \vdots \\ y(\tau_r-1) & y(\tau_r) & \cdots & \end{bmatrix}$$

and Ω is the block-Toeplitz matrix

$$\Omega = \begin{bmatrix} u(0) & u(1) & u(2) & u(3) & \cdots & u(N-1) \\ 0 & u(0) & u(1) & u(2) & \cdots & u(N-2) \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & u(0) & \cdots & u(N-l) \end{bmatrix}$$

and thus the covariance-function approach is similar to the weighting used in the raw-data case, but with an additional permutation to transform the block-Hankel structure of the weighting data matrix into a block-Toeplitz form. It is *not* equivalent, however, since while the raw-data case relies on the projection to accomplish some of the de-correlation, the projection is applied afterwards in the case of covariance-function estimates.

Of course, when constructing data matrices of covariance function estimates, it is best to compute the covariance function estimates directly, then construct the matrices, rather than use the product of the two matrices above.

5.3 Least-Squares Estimation of Input Parameters from Covariance Functions

An important difference when using covariance-function estimates for identification is that the signals are matrix-valued. This can be problematic for the least-squares estimation of the input parameters, which is often the most computationally expensive step in the identification procedure, since the regressors will dramatically increase in size with increasing input dimension.

To extend the least-squares estimation of B, D, and $x(0)$ to covariance function estimates $\hat{R}_{u\xi}(\tau)$ and $\hat{R}_{y\xi}(\tau)$ calculated over $\tau \in [\tau_0, \tau_N]$, we instead solve the least squares problem

$$\left(\hat{B},\ \hat{D},\ \hat{R}_{xu}(\tau_0)\right) = \underset{B,D,R_{xu}(\tau_0)}{\arg\min} \left|\left|\hat{R}_{y\xi} - \beta\right|\right|_2 \tag{5.6}$$

in which

$$\hat{R}_{y\xi} = \begin{bmatrix} \hat{R}_{y\xi}(\tau_0) \\ \hat{R}_{y\xi}(\tau_0+1) \\ \vdots \\ \hat{R}_{y\xi}(\tau_1) \end{bmatrix} \in \mathbb{R}^{Pn_y \times n_\xi} \qquad \beta = \begin{bmatrix} \beta(\tau_0) \\ \beta(\tau_0+1) \\ \vdots \\ \beta(\tau_1) \end{bmatrix} \in \mathbb{R}^{Pn_y \times n_\xi},$$

where $P = \tau_N - \tau_0 + 1$, and

$$\beta(\tau) = \hat{C}\hat{A}^\tau R_{x\xi}(\tau_0) + \sum_{k=0}^{\tau-1} \hat{C}\hat{A}^{\tau-k-1} B \hat{R}_{u\xi}(k) + D\hat{R}_{u\xi}(\tau).$$

Vectorizing $\beta(\tau)$ and separating the unknown parameters B, D and $R_{x\xi}(\tau_0)$ via the Kronecker product results in

$$\begin{aligned} \operatorname{vec}\left(\beta(\tau)\right) = &\left(I_{n_\xi} \otimes \hat{C}\hat{A}^\tau\right) \operatorname{vec}(R_{x\xi}(\tau_0)) \\ &+ \left(\hat{R}_{u\xi}(\tau)^T \otimes I_{n_y}\right) \operatorname{vec}(D) + \left(\sum_{k=0}^{\tau-1} \hat{R}_{u\xi}(k)^T \otimes \hat{C}\hat{A}^{\tau-k-1}\right) \operatorname{vec}(B), \end{aligned}$$

and we obtain the new regressor

$$\begin{aligned}\operatorname{vec}(\beta(\tau)) &= \begin{bmatrix}\phi_{R_{x\xi}}(\tau)^T & \phi_D(\tau)^T & \phi_B(\tau)^T\end{bmatrix}\begin{bmatrix}\operatorname{vec}(R_{x\xi}(\tau_0)) \\ \operatorname{vec}(D) \\ \operatorname{vec}(B)\end{bmatrix} \\ &= \phi(\tau)^T\theta,\end{aligned}$$

where

$$\phi_{R_{x\xi}}(\tau)^T = I_{n_\xi} \otimes \hat{C}\hat{A}^\tau \in \mathbb{R}^{n_y n_\xi \times n n_\xi} \tag{5.7}$$

$$\phi_D(\tau)^T = \hat{R}_{u\xi}(\tau)^T \otimes I_{n_y} \in \mathbb{R}^{n_y n_\xi \times n_y n_u} \tag{5.8}$$

$$\phi_B(\tau)^T = \sum_{k=0}^{\tau-1} \hat{R}_{u\xi}(k)^T \otimes \hat{C}\hat{A}^{\tau-k-1} \in \mathbb{R}^{n_y n_\xi \times n n_u}. \tag{5.9}$$

Although the computation of covariance function estimates will reduce the number of matrices needed to estimate B from $O(N^3/2)$ to $O\left(P^3/2\right)$, the matrices have grown to size $n_y n_\xi \times n n_u$. Estimating $\hat{R}_{x\xi}(\tau_0)$ now requires P matrices of the same size, and the matrices needed to estimate D have grown to size $n_y n_\xi \times n_y n_u$. The consequence of the growth in size of the regression matrices is that the linear regression problem becomes computationally intractable for multivariable systems with a large number of inputs and outputs.

To alleviate the difficulties caused by the effective increase in dimensionality, we extend Theorem 4.2 to covariance function estimates, which achieves similar savings in memory efficiency.

Theorem 5.1. *Block element (i,j) of the transposed regressor ϕ_B in* (5.9) *may be calculated from state sequences of the system*

$$\phi_B^{(i,j)}(\tau+1) = \hat{A}^T\phi_B(\tau) + \hat{C}^T\tilde{u}(i,j,\tau) \tag{5.10}$$

with the initial condition $\phi_B(\tau_0) = 0_n$, in which

$$\tilde{u}(i,j,\tau) = \begin{bmatrix} 0_{\gamma-1} \\ \hat{R}_{u_\alpha,\xi_\beta}(\tau) \\ 0_{n_y-\gamma}\end{bmatrix} \qquad \begin{aligned}\alpha &= \operatorname{floor}((i-1)/n_u) \\ \beta &= \operatorname{floor}((j-1)/n_y)\,, \\ \gamma &= \operatorname{mod}(j-1, n_y)\end{aligned}$$

where $\operatorname{mod}(r,s) : \mathbb{Z}\times\mathbb{Z} \to \mathbb{Z}$ *is r modulo s, and* $\operatorname{floor}(r) : \mathbb{R} \to \mathbb{Z}$ *is the nearest integer* $\leq r$.

Proof. Expand (5.9) in the same manner as in Theorem 4.2 to find

$$\phi_B(\tau) = \sum_{k=0}^{\tau-1} \left(I_{n_u} \otimes \left(\hat{A}^T\right)^{\tau-k-1} \hat{C}^T \right) \left(R_u(k) \otimes I_{n_y} \right).$$

The term within the summation – similar to (4.39) – becomes

$$\begin{bmatrix} \hat{A}^{T^\eta}\hat{C}^T \left(R_{u_1\xi_1}(k) I_{n_y} \right) & \hat{A}^{T^\eta}\hat{C}^T \left(R_{u_1\xi_2}(k) I_{n_y} \right) & \cdots \\ \hat{A}^{T^\eta}\hat{C}^T \left(R_{u_2\xi_1}(k) I_{n_y} \right) & \hat{A}^{T^\eta}\hat{C}^T \left(R_{u_2\xi_2}(k) I_{n_y} \right) & \cdots \\ \vdots & \ddots & \end{bmatrix}$$

where $\eta = t - k - 1$. The above may then be interpreted as a convolution as in Theorem 4.2. □

5.4 Practical Aspects

In general, for experiments with large amounts of data, identification from covariance-function estimates is highly advantageous due to the decrease int the size of the data-matrices. For smaller data sets, however, the raw-data method may be preferable, since the covariance function estimates are far from converging.

Though identification by means of covariance function estimates provides many advantages, it also complicates the procedure somewhat due to the increase in tunable parameters. In this section, we discuss some of these choices.

5.4.1 Domain of the Covariance Function Estimates

The domain of τ over which $R_{u\xi}(\tau)$ is used must be chosen carefully, as $R_{u\xi}(\tau)$ will likely have a brief maximum at some τ then likely lose amplitude quickly. This of course depends on the bandwidth of the excitation signal used. Assume that $\xi(t) = u(t)$. White noise input will cause $R_u(\tau)$ to converge to an impulse at $\tau = 0$ while colored noise input will have a symmetric $R_u(\tau)$ that rings around $\tau = 0$. As a result, beginning the data matrices with some index $\tau_0 < 0$ will sometimes provide a more accurate estimate.

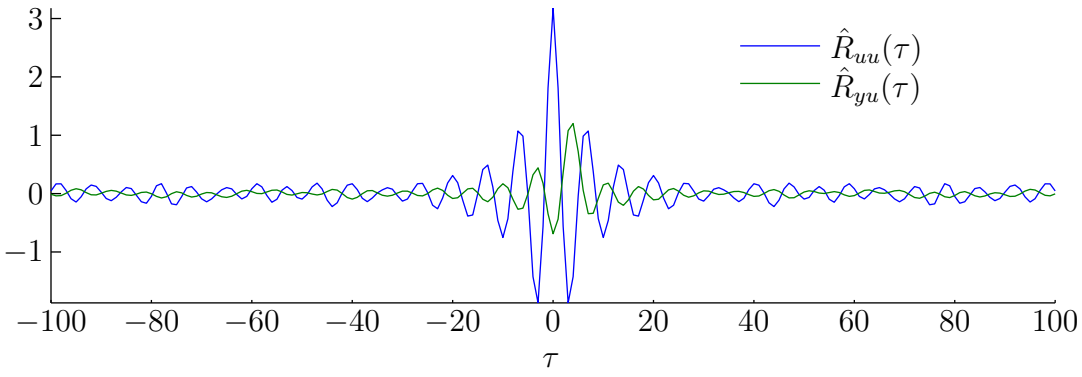

Figure 5.1: Covariance functions for Example 5.1.

Example 5.1. Consider the state-space system described by the parameters

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} 0.7 & 0.3 & 0 \\ -0.3 & 0.7 & 0.8333 \\ \hline 0.72 & 0 & 0 \end{array}\right].$$

Suppose $v(t)$ is generated by white noise with variance $\sigma_v^2 = 10$ filtered through a system with parameters

$$\left[\begin{array}{c|c} A_v & B_v \\ \hline C_v & D_v \end{array}\right] = \left[\begin{array}{cc|c} 0.6 & 0.4 & 0 \\ -0.4 & 0.6 & 1.25 \\ \hline 0.64 & 0 & 0 \end{array}\right]$$

and added to the output, and suppose $u(t)$ is generated by white noise with variance $\sigma_u^2 = 1$ filtered through the system

$$\left[\begin{array}{c|c} A_u & B_u \\ \hline C_u & D_u \end{array}\right] = \left[\begin{array}{cc|c} 0.5 & 0.7 & 0 \\ -0.7 & 0.5 & 1.429 \\ \hline 1 & 0 & 0 \end{array}\right].$$

The input is highly-colored in this case, as shown in Figure 5.1. Note that the input auto-covariance function has reached a significantly large magnitude well before $\tau = 0$.

The covariance-based realization algorithm was applied to 200 datasets of 2000 samples. The algorithm was applied to covariance functions estimated over

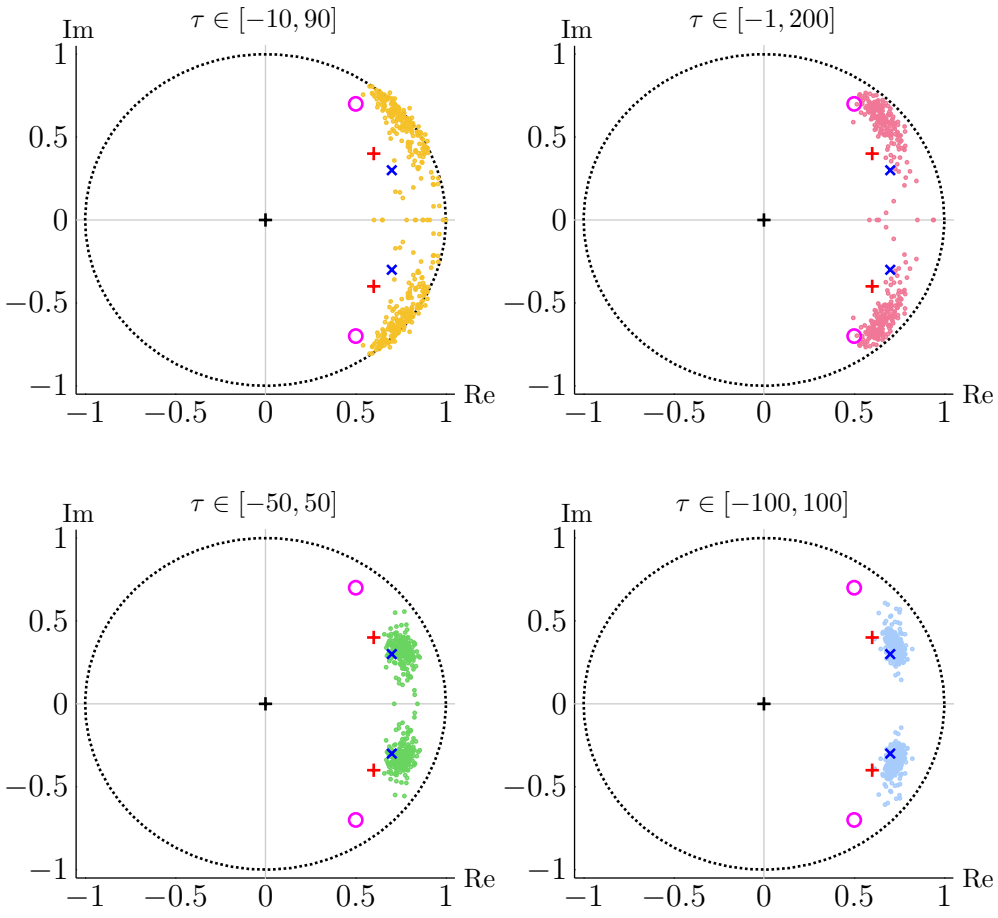

Figure 5.2: Estimated pole locations for Example 5.1.

different domains of τ. In all cases the block-row dimension of the data matrices was $r = 18$. The locations of the estimated poles for various domains of τ together with system, noise, and input filter poles are shown in Figure 5.2.

Clearly, the domain of τ has a profound effect on the accuracy of the estimated model when the input signal is highly colored. Very poor estimates result when the domain of τ does not extend sufficiently backward before $\tau = 0$.

5.4.2 Frequency-Domain Smoothing

An interesting property of cross-covariance functions is that they are equivalent to the Fourier transform of the cross-power spectra; that is, if $\mathcal{F}[\cdot]$ is the

Fourier transform operator, then for two signals $s(t)$ and $w(t)$ with Fourier transforms $S(e^{j\omega})$ and $W(e^{j\omega})$,

$$S(e^{j\omega})W(e^{j\omega})^H = \mathcal{F}\left[R_{sw}(\tau)\right]$$

where H denotes the Hermitian (complex-conjugate transpose). As a result, frequency-domain smoothing [58, for instance] may be trivially applied to covariance function estimates. This will be used later in the example application.

5.5 Summary of Covariance-Function Estimate Realization Procedure

The following steps completely describe the covariance-function estimate realization algorithm:

1. Construct and examine the covariance-function estimates $\hat{R}_{u\xi}(\tau)$ and $\hat{R}_{y\xi}(\tau)$ to determine an appropriate domain of τ.

2. Construct the block-Hankel data matrices $\hat{R}^{y\xi}_{\tau_0|\tau_r-1}$ from (5.2), $\hat{R}^{y\xi}_{\tau_1|\tau_r}$ from (5.3), and $\hat{R}^{u\xi}_{\tau_0|\tau_r}$.

3. Form the projections $R^{y\xi}_{\tau_0|\tau_r-1}\Pi_{R_u^\perp}$ and $R^{y\xi}_{\tau_1|\tau_r}\Pi_{R_u^\perp}$ via the LQ-decomposition as in the raw-data case.

4. Take the SVD of $R^{y\xi}_{\tau_0|\tau_r-1}\Pi_{R_u^\perp}$, and examine the singular values to determine the system order if necessary.

5. Estimate $\hat{A}$ as (5.5) and $\hat{C}$ from the first n_y rows of $U_n\Sigma_n^{1/2}$.

6. Estimate B, D, and $x(0)$ by solving (5.6).

5.6 Acknowledgements

This chapter contains material presented in the following publications:

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," to appear in *AIAA Journal of Guidance, Control and Dynamics*, 2012.

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," *Proc. of the AIAA Atmospheric Flight Mechanics Conference*, Portland, OR: August 2011.

D.N. Miller and R.A. de Callafon, "Subspace Identification Using Dynamic Invariance in Shifted Time-Domain Data," *Proc. of the 49th IEEE Conference on Decision and Control*, Atlanta, GA: December 2010.

D.N. Miller and R.A. de Callafon, "Subspace Identification from Classical Realization Methods," *Proc. of the 15th IFAC Symposium on System Identification*, Saint-Malo, France: July 2009.

6 Application I: Identification of Aeroelastic Dynamics

In the first of two application chapters, we apply the covariance-based realization algorithm to the identification of aeroelastic dynamics from in-flight experimental data. The algorithm is well-suited to such experiments, since it produces unbiased estimates from very large data sets. A detailed background on the problem of aeroelasticity is omitted; a basic treatment of the subject is beyond the scope of this dissertation. Good references for the topic include Clark, Cox, Curtiss, Edwards, Hall, Peters, Scanlan, Simiu, Sisto, and Strganac [14] and Wright and Cooper [83].

6.1 Introduction

Vibrations due to aeroelastic dynamics of aircraft structures, commonly referred to as flutter, have the potential to damage and destroy aircraft in flight if not properly analyzed and suppressed. The current trend in the analysis of ASE dynamics is to derive finite-element and computational-fluid-dynamic models of an airframe at various flight conditions, and to interpolate and extrapolate the damping of flutter modes across the full flight envelope. These computational models are then validated through ground testing and, finally, in-flight testing before the aircraft can be considered operationally safe [9]. Of particular concern to control systems engineers are aero-servo-elastic (ASE) dynamics, which is flutter induced by the aircraft's control surfaces. Unstable ASE dynamics may occur in both closed- and open-loop operation.

In-flight analysis of flutter is inherently difficult due not only to its dangerous nature, but also to the unsteady, turbulent phenomena that induce it. These effects manifest themselves as essentially non-deterministic disturbances, or noise, on acquired data. By nature this noise is colored and correlated across all measured signals; perturbations on control surface positions due to turbulent air flow are inherently correlated with the perturbations measured in stress and acceleration on the airframe. Attempts to analyze the dynamic behavior of the control surfaces on the airframe from data generated from in-flight experiments must take these facts into account to avoid inaccurate conclusions.

Most system identification methods assume that the noise on measured signals is either white, uncorrelated, or both, and are thus ill-suited for identifying ASE dynamics. When dealing with experimental data that does not meet these assumptions, techniques from the analysis of stochastic processes must be incorporated into the identification methods used. Additionally, many system identification methods are based on nonlinear optimizations over cost functions that become extremely non-convex for large, high-dimensional data sets, making them infeasible for ASE analysis, in which many sensors are employed to capture the behavior of the airframe.

Traditional subspace methods [75] have been previously applied to the identification of aeroelastic dynamics using simulated data from an F-16 aircraft and measured data from a V-22 rotocraft in Mehra, Mahmood, and Waissman [48]. Such methods assume strictly deterministic inputs in order to remove the effects of subsequent input on the propagation of the state dynamics and in order to decorrelate the deterministic and non-deterministic subsystems. A subspace-based method for online monitoring of aeroelastic damping was developed and applied to in-flight data by Mevel, Goursat, Benveniste, and Basseville [50]. This method utilized output data only and relied on the auto-covariance of the data to determine when statistically-significant damping of vibration modes dropped below a given threshold but did not identify the input-output behavior of the aeroelastic phenomena and assumed no deterministic control-surface excitation during data acquisition. This method was later extended to include known, strictly-deterministic

inputs in Mevel, Benveniste, Basseville, Goursat, Peeters, Van der Auweraer, and Vecchio [49].

These previous studies all assume disturbances to be white, which in practice is often insufficient. Non-deterministic effects from turbulence, sensor noise, and, in the closed-loop case, control-system feedback will inevitably produce colored noise on the output data. In such cases, either the modes of the estimated system will be biased by the disturbance spectrum [37], or, if the model-order is chosen to be artificially high, the observable modes of the strictly non-deterministic subsystem will be estimated along-side the modes of the deterministic subsystem but be incorrectly identified as controllable [53]. This is particularly problematic if the dynamic model is intended to be used for active flutter suppression, as the control algorithm designed from the derived model will attempt to control the uncontrollable modes. Additionally, treatment of the input as strictly deterministic is only possible if the input measured is actuator commands. In this case, the derived model will include actuator dynamics (such as servomotor dynamics) as well as aeroelastic dynamics. If actuator positions are measured instead, the position measurements will include perturbations which are correlated with the noise on the measured output data, and the effects of the input on state-dynamics cannot be removed with the standard methods of orthogonal projections.

Alternative proposed methods of estimating ASE dynamics include applying frequency-domain total-least-squares by restricting the identification to error-in-variables models in Verboven, Cauberghe, Guillaume, Vanlanduit, and Parloo [77], which allows for the incorporation of colored noise. An approach based on rational orthogonal basis functions incorporated static input and output nonlinearities and addressed the issue of identifying parameter-varying models in Baldelli, Zeng, Lind, and Harris [5]. Neither allows for the presence of correlated noise on both the input and output measurements, and unlike subspace methods, these methods all require *a priori* parameterization of the dynamic system.

In the following section, the COBRA method is applied to the analysis of ASE dynamics of the NASA Active Aeroelastic Wing (AAW) F/A-18. The combination of correlated noise on the input and output measurements of the measured

data and the very large dimensions of the signals makes the COBRA method an appropriate choice for the analysis of ASE dynamics; the method requires no iterative optimization, uses comparatively small matrices compared to other subspace methods, and provides unbiased estimates when the input and output noise are correlated with an external signal uncorrelated with the noise.

6.2 Identification of Aero-Servo-Elastic Dynamics

The COBRA method was applied to in-flight data taken from accelerometer and pressure measurements on-board the AAW F/A-18, located at NASA's Dryden Flight Research Center, which is a fighter aircraft that has been modified for aeroelastic research. The algorithm requires careful selection of the instrument signal $\xi(t)$ to ensure that the system estimate is unbiased. If $\xi(t)$ is chosen incorrectly, the result may be biased by either the noise process or unwanted system dynamics. In the following two examples, the choice of an appropriate $\xi(t)$ is discussed in detail.

6.2.1 Collective Leading-Edge Flap Excitation

Consider the identification of the response from the LEF to the acceleration and pressure sensors. Signal pathways for the system are shown in Figure 6.1, in which G_{lef} is the collective LEF actuator dynamics and G the ASE dynamics of interest. The collective LEF position $u(t)$ is perturbed by a noise signal $v_{\text{lef}}(t)$ that *must* be assumed correlated with the noise $v(t)$ on the acceleration and pressure measurements $y(t)$. The result is that identification directly from $u(t)$ to $y(t)$ will be biased by the cross-spectrum of the two noise signals, regardless of the identification algorithm used, unless steps are taken to de-correlate them from the measured data.

The reference excitation $r(t)$ was chosen to be a minimax crest factor multisine [58] of bandwidth between 3 Hz and 35 Hz. The power-spectral density (PSD)

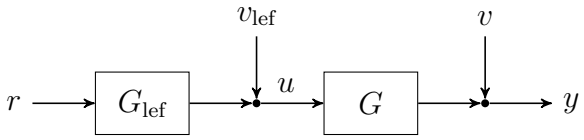

Figure 6.1: Leading-edge flap experiment signal pathways.

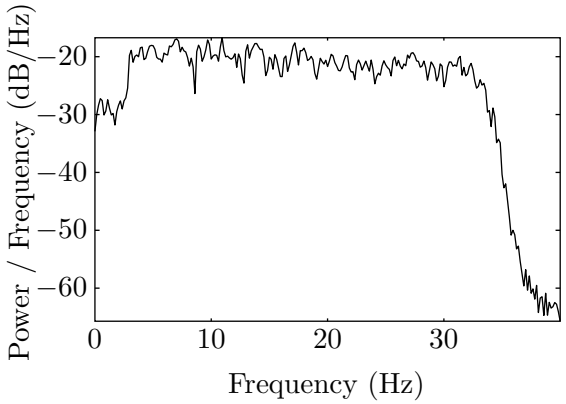

Figure 6.2: Power-spectral density of OBES signal for collective LEF excitation.

of $r(t)$ is shown in Figure 6.2. It can be seen that $r(t)$ closely resembles white noise in the frequency range of interest. The signal $r(t)$ is uncorrelated with either noise signal, since it is deterministic; it may also be treated as quasi-stationary, since as a sum of sinusoids, its autocovariance function exists. Hence the mapping between the cross-covariance functions $R_{yr}(\tau)$ and $R_{ur}(\tau)$ is limited to the dynamics G, and we select $\xi(t) = r(t)$ when analyzing the data.

The cross-covariance estimate $\hat{R}_{ur}(\tau)$ is shown in Figure 6.3. Only the data in which the excitation signal $r(t)$ is nonzero was used to calculate the PSD and cross-covariance functions. The cross-covariance functions were further truncated to $\tau \in [-20, 100]$ after calculation for identification purposes, since, as τ increases, the signal-to-noise ratio of the cross-covariance estimates becomes prohibitively small.

Because 94 signals were available for use in identification, an objective criteria was created to determine which had a sufficiently high signal-to-noise ratio. Only signals which had magnitude-square coherence with $r(t)$ of at least 2/3 av-

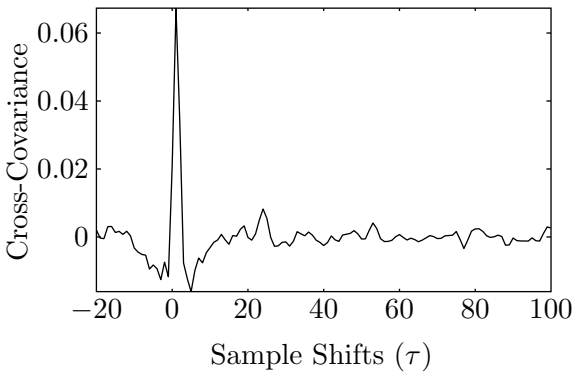

Figure 6.3: Cross-covariance function estimate between collective LEF position (u) and reference (r) for LEF excitation.

eraged over the frequency range 3–35 Hz were selected from the available measurements. The locations of used and unused accelerometers are shown in Figure 6.4. Only the top-front-left pressure sensor was used. Although only 8 total signals were used for identification in this experiment, the collective LEF input is intended to excite neither rigid-body moments nor bending moments on the wing, so the low number of usable signals is expected. Excitation of other surfaces will naturally produce different selections of signals for identification purposes. A sample of signals measured for the experiment is shown in Figure 6.5.

A model was constructed using the method proposed in Chapter 5. The singular values of the matrix are shown in Figure 6.6. The system order was chosen to be $n = 6$, which is just before the magnitude of the singular values appears to flatten out.

Time-domain simulations of the estimated model are shown with the data in Figure 6.7. Cross-covariance estimates from the simulated data and measured data are shown in Figure 6.8. Finally, Bode plots of the the estimated system are compared with spectral estimates (computed from the cross-spectrum of $y(t)$ with $r(t)$ and $u(t)$ with $r(t)$) in Figure 6.9.

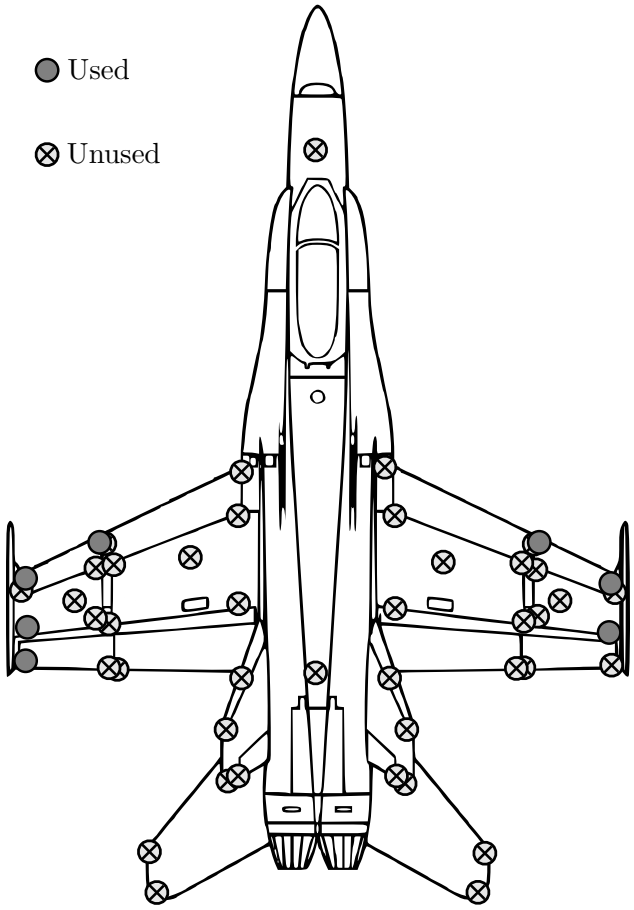

Figure 6.4: Locations of used and unused accelerometers for the collective LEF experiment.

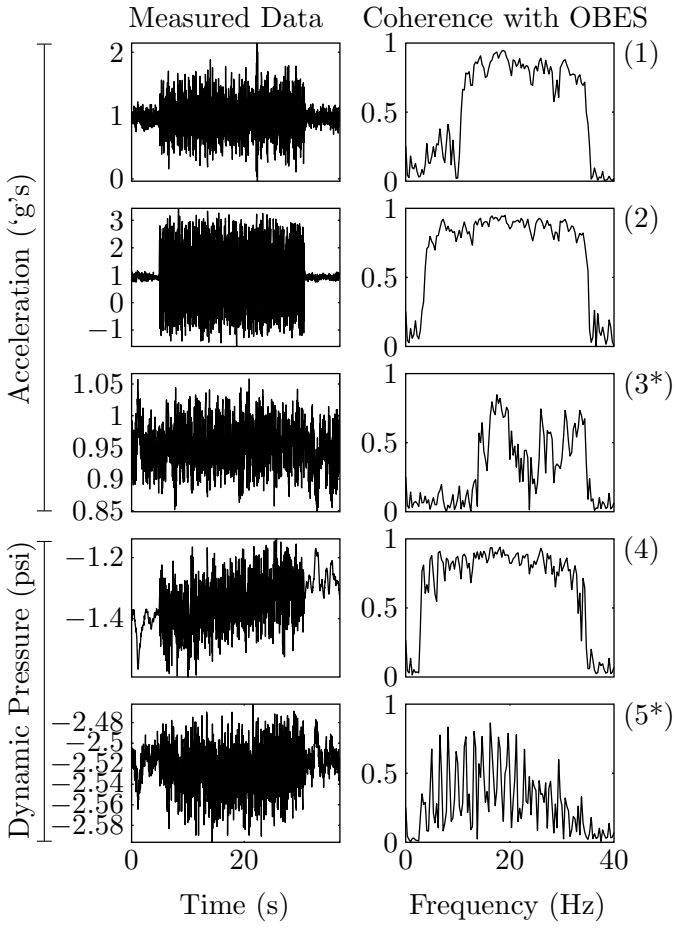

Figure 6.5: Sample of signals measured for the collective leading-edge flap experiment.

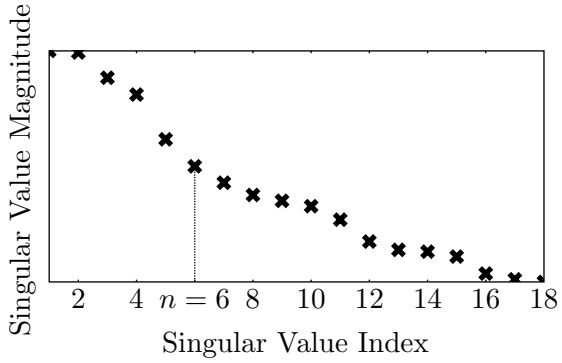

Figure 6.6: Singular values of the projected data matrix for the collective LEF experiment (y-axis in log scale).

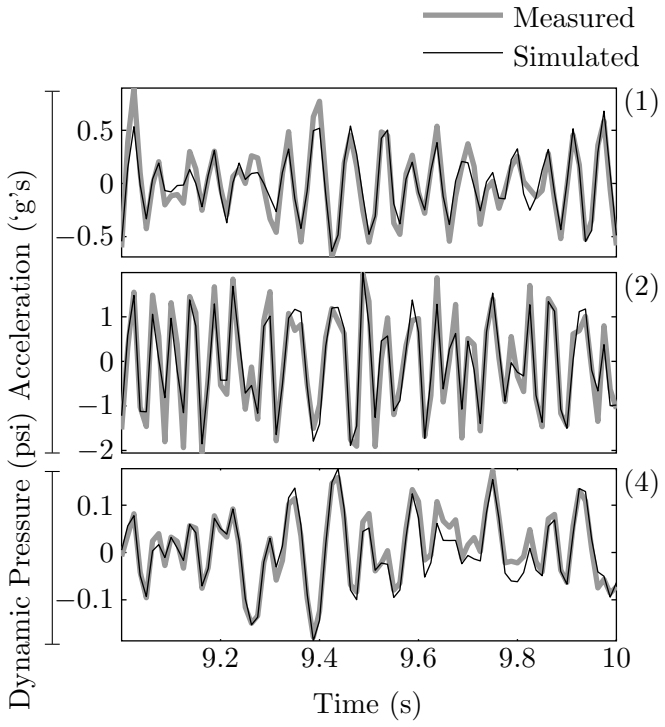

Figure 6.7: Sample of simulation results of the collective LEF experiment.

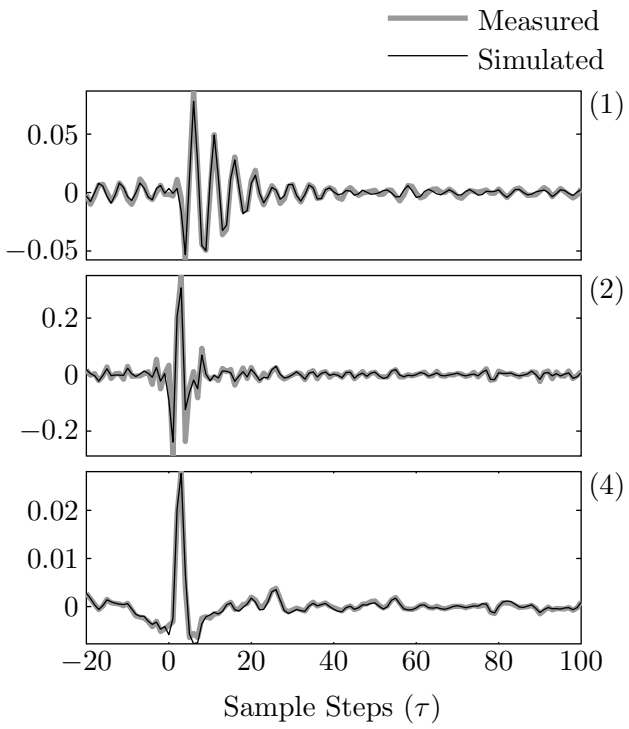

Figure 6.8: Sample of simulation cross-covariance estimates of the collective LEF experiment.

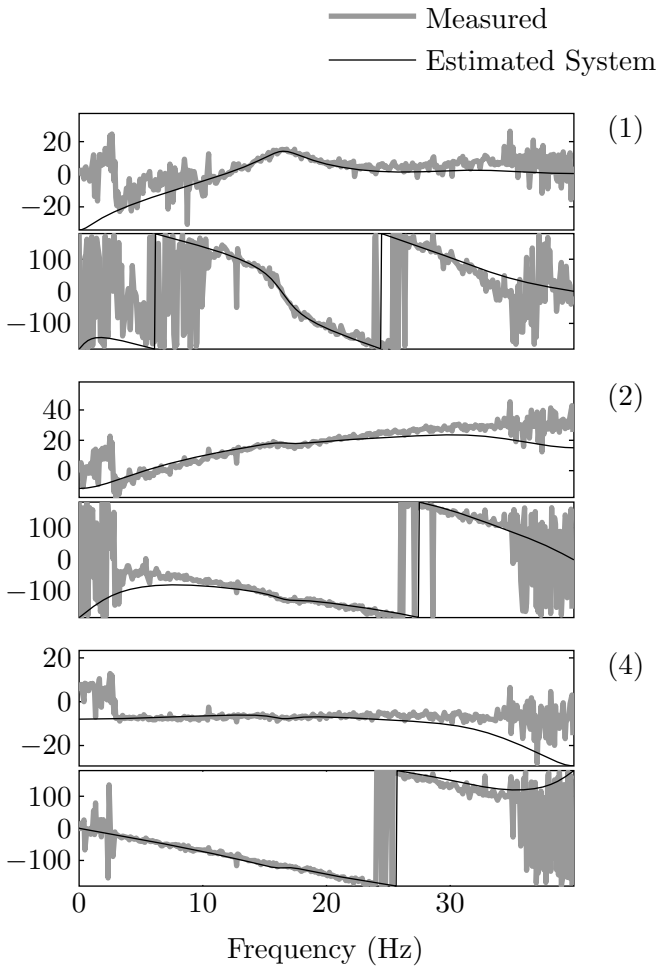

Figure 6.9: Bode plot of the estimated system and spectral estimate of the collective LEF experiment. Magnitude is in dB, phase in degrees.

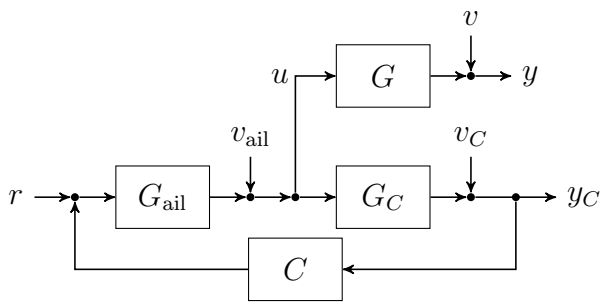

Figure 6.10: Aileron experiment signal pathways.

6.2.2 Differential Aileron Excitation

Next consider the identification from the differential aileron input to the acceleration and pressure sensors. Signal pathways are shown in Figure 6.10. As before, the input $u(t)$ is perturbed by a noise signal $v_{\text{ail}}(t)$ and the output $y(t)$ by a noise signal $v(t)$. Additionally, the system contains a feedback controller C, which augments the excitation $r(t)$ with a differential aileron command. The feedback signals to the control system $y_C(t)$ are the result of both rigid-body and ASE dynamics, represented in a combined system G_C. The feedback $y_C(t)$ also contains a noise signal $v_C(t)$, which must be assumed correlated with $v_{\text{ail}}(t)$ and $v(t)$.

Because $v_C(t)$ appears in $u(t)$ after being filtered through the dynamics of G_C, C and the aileron servo G_{ail}, identification from $u(t)$ to $y(t)$ will provide an estimate biased by the subsystems G_{ail}, G_C, and C in addition to the various cross-spectra of $v(t)$, $v_{\text{ail}}(t)$, and $v_C(t)$. As before, however, the reference $r(t)$ is uncorrelated with the noise signals and may be used as an instrument $\xi(t) = r(t)$ to provide unbiased results.

Sample signals of the differential aileron experimental data are shown in Figure 6.11. The signals shown are (1) lateral acceleration at the nose, (2) acceleration at the right forward wing-tip, (3) axial acceleration at the right outer-wing, (4) acceleration at the right aft wing-tip, (5*) acceleration at the right-aft wing-root, (6*) dynamic pressure at the right top front pressure tap, (7) dynamic pressure at the right top rear pressure tap. The same coherence-based criteria of the LEF

experiment was used to determine which signals were acceptable for identification purposes; signals marked by '*' were designated unacceptable and not used in the identification algorithm. A total of 49 output signals were used for identification purposes for this experiment.

Locations of all used and unused accelerometers are shown in Figure 6.12. Observe that the usable accelerometers are distributed primarily over the wings as one would expect from a differential aileron excitation. The selected accelerometer in the nose measures lateral motion, explaining its high coherence with $r(t)$.

A model was again constructed using the method proposed in Chapter 5. The singular values of the projected data matrix are shown in Figure 6.13. The system order was chosen to be $n = 12$, which is naturally larger than that of the LEF experiment due to the large increase in the output dimension n_y. Additionally, the ailerons have much more inertial excitation than the LEF's, being heavier and a larger geometric proportion of the wings, so more responsiveness is expected overall.

Samples of 5 estimated signal pathways for the 49 used output signals are shown in Figures 6.14 through 6.16. Time-domain simulations are plotted with measured data in Figure 6.14, and comparisons with cross-covariance function estimates in Figure 6.15. The enumeration is the same as in Figure 6.11. Spectral estimates and Bode plots of the estimated system are shown in Figure 6.16.

6.3 Conclusion

We have demonstrated that the COBRA method produces accurate, unbiased, linear models from measured data of large signal dimension (ie. data acquired from many sensors). The convergence of covariance function estimates is used to handle large data sets in both open- and closed-loop experiments, making it well-suited to ASE analysis. The algorithm has been successfully applied to data measured in flight from the NASA Active Aeroelastic Wing F/A-18 for both open-loop and closed-loop experiments.

As a final note, we mention that the algorithm is capable of analyzing data

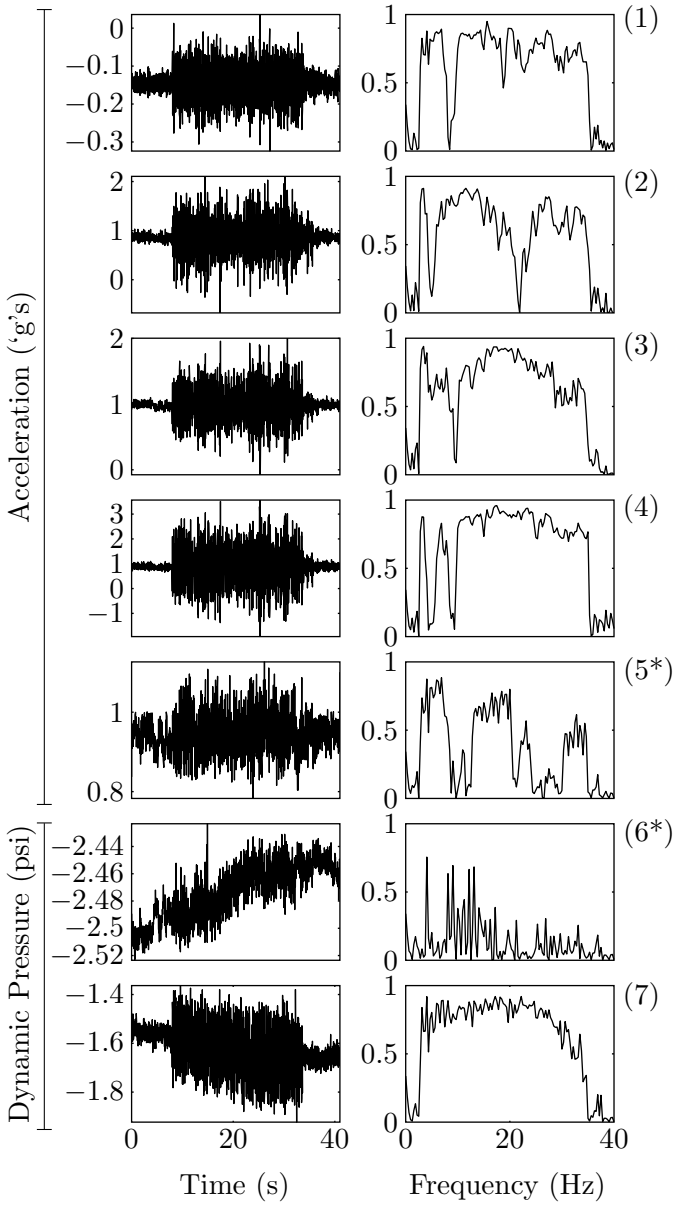

Figure 6.11: Sample of signals used for the differential aileron experiment.

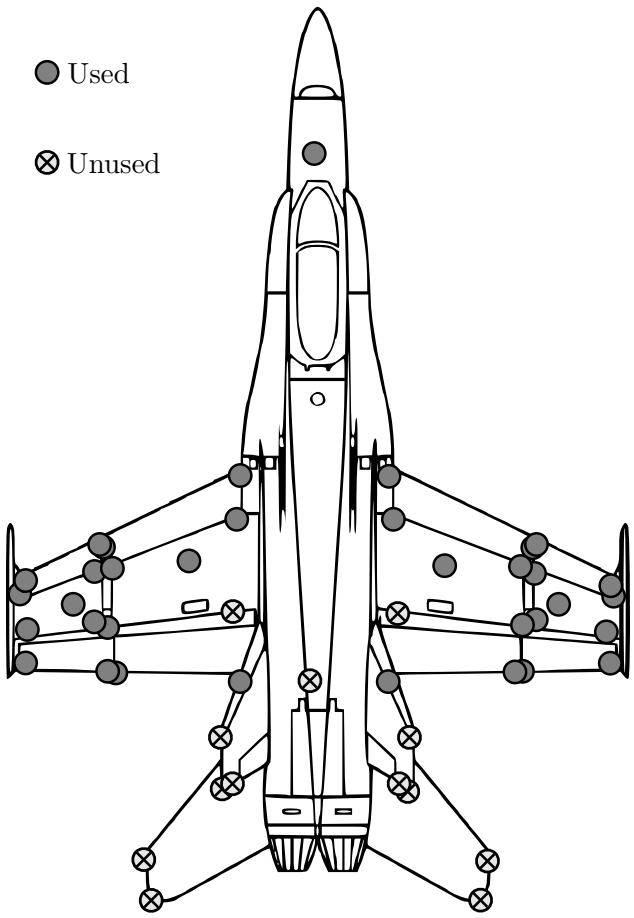

Figure 6.12: Locations of used and unused accelerometers for the differential aileron experiment.

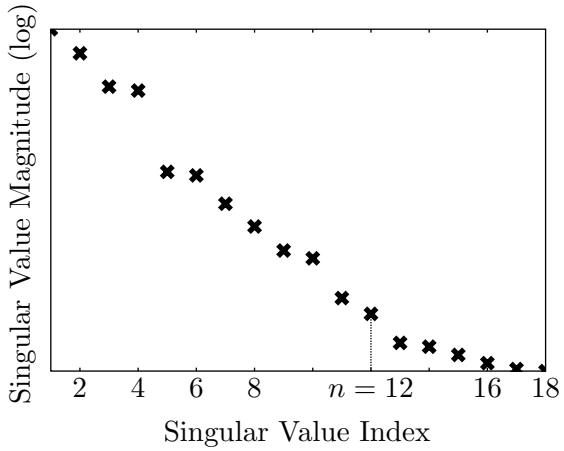

Figure 6.13: Singular values of the projected data matrix for the differential aileron experiment.

from multiple inputs and references and would, in theory, provide similar results were the two experiments combined into a single experiment. Data for such an experiment, however, is currently unavailable to the author.

6.4 Acknowledgements

This chapter, in part, includes content from the following publications:

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," to appear in *AIAA Journal of Guidance, Control and Dynamics*, 2012.

D.N. Miller, R.A. de Callafon, and M.J. Brenner, "A Covariance-Based Realization Algorithm for the Identification of Aeroelastic Dynamics from In-Flight Data," *Proc. of the AIAA Atmospheric Flight Mechanics Conference*, Portland, OR: August 2011.

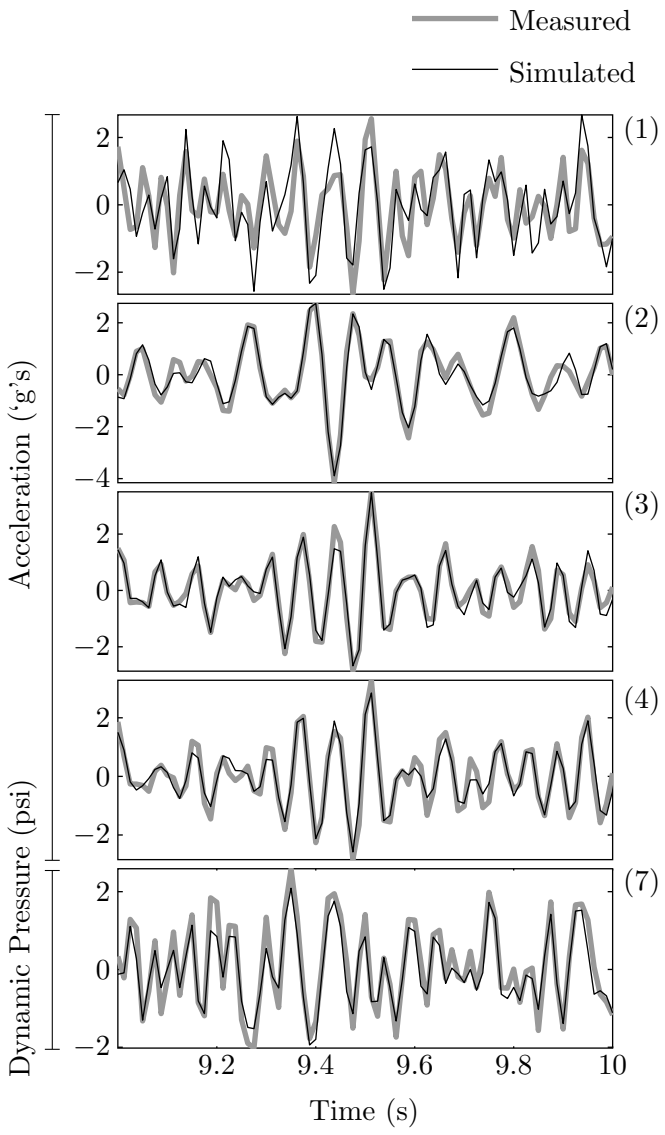

Figure 6.14: Sample of simulation results of the differential aileron experiment.

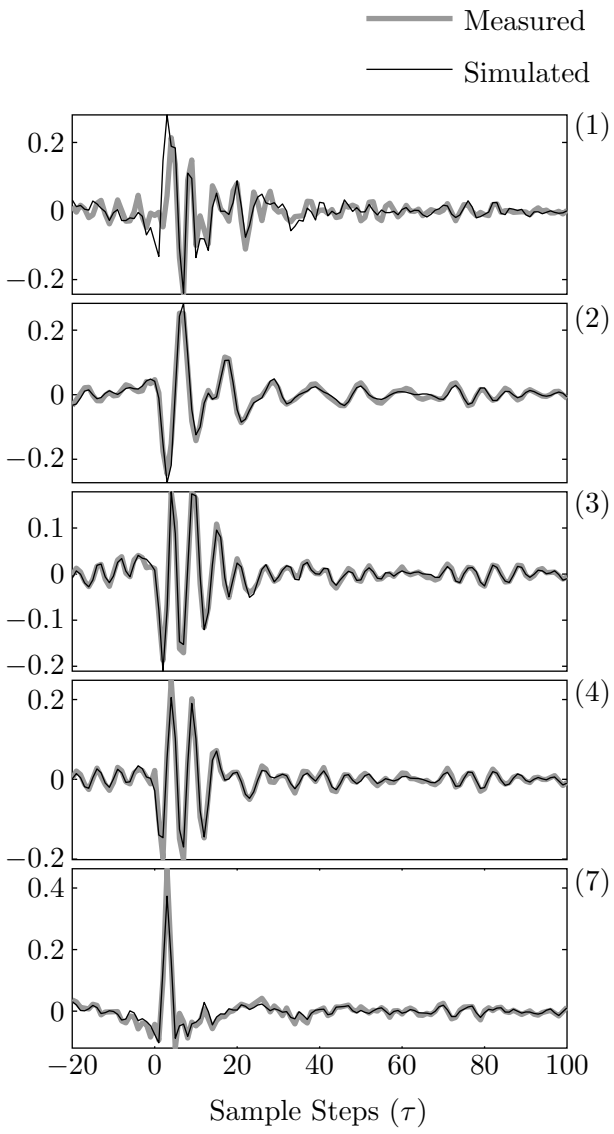

Figure 6.15: Sample of simulation cross-covariance estimates of the differential aileron experiment.

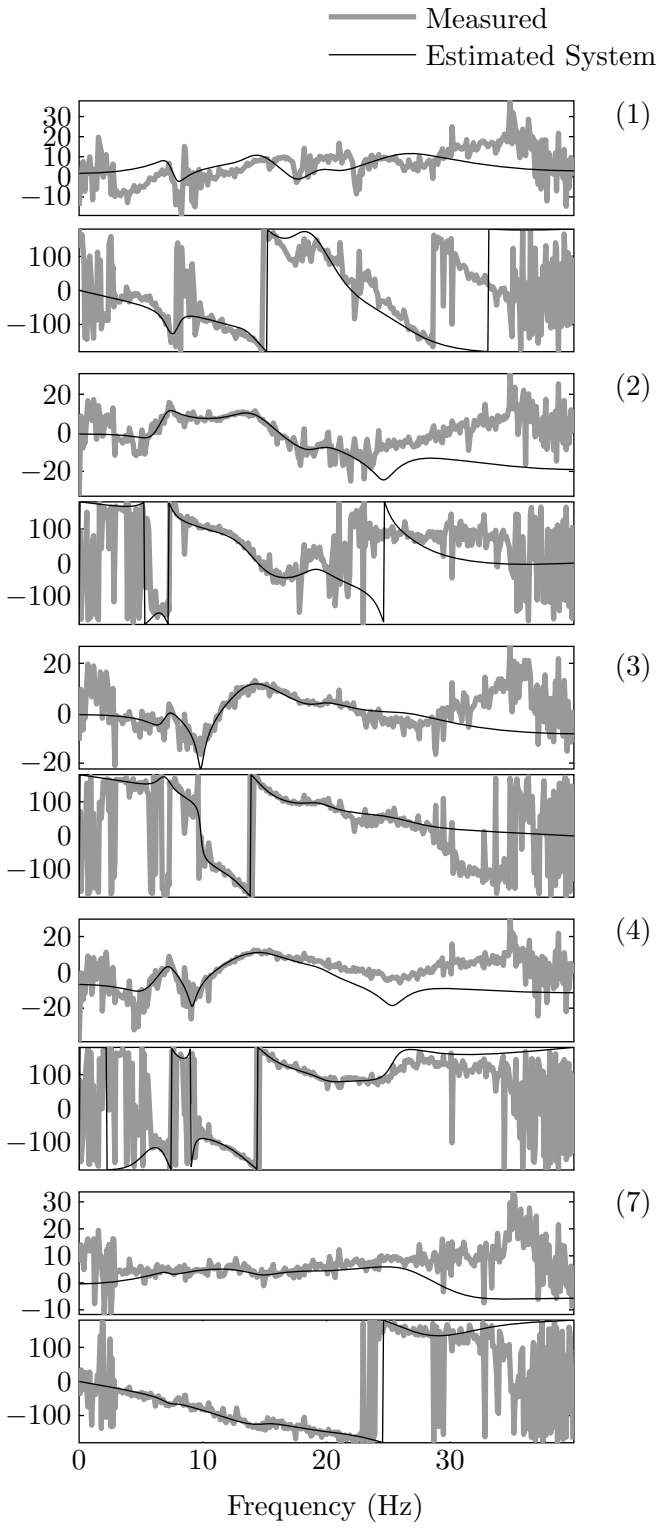

Figure 6.16: Bode plot of the estimated system and spectral estimate of the differential aileron experiment.

7 Incorporating Constraints into the Realization Problem

In this chapter, we augment the realization problem with convex constraints with the goal of producing models that meet *a priori* requirements. Although the constrained solutions can no longer be formulated as a finite sequence of linear algebra operations, they can be solved via convex optimization techniques, which guarantee convergence to a solution. Two types of constraints are introduced: constraints on the location of the system poles in the complex plane, and constraints on time-domain behavior.

7.1 Introduction

When identifying models of systems from measured data, it is often desirable that the identified model behave in agreement with prior knowledge of the system. This is ordinarily limited to basic knowledge of system stability or an assumed model order, but other times this knowledge is derived from first-principles laws that govern the underlying system dynamics. The system identification literature, however, tends to focus on "black-box" modeling approaches that limit the type of constraints that may be incorporated into the identification process.

One possible reason for the lack of constrained identification procedures is that the classical prediction-error framework relies on the optimization of typically non-convex cost functions. Such optimizations are already computationally challenging without adding possibly non-convex constraints. Realization-based identification methods (including subspace identification methods), by contrast, use a

fixed number of linear algebra operations to achieve consistent estimates even in the presence of colored noise. While generally non-optimal in a prediction-error or maximum-likelihood sense, all realization-based methods nonetheless minimize some Frobenius norm, the argument of which is affine in the identified parameters. The inherent convexity of the minimization has so far been largely, though not entirely, unexploited.

Unlike most classical prediction-error identification methods, realization-based methods are capable of identifying unstable systems. While this can be an advantage if the system is in fact unstable, it can become a problem when the system is known to be stable yet the identified model is unstable. The underlying cause of instability in these methods is that the estimate of the extended observability matrix

$$\mathcal{O}_r = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{r-1} \end{bmatrix}$$

does not sufficiently decay in magnitude at lower block rows, implying A has a spectral radius (magnitude of largest eigenvalue) greater than 1. If the spectral radius of A is close to 1, or if non-deterministic effects cause the estimated magnitude of $\mathcal{O}_r$ to grow at lower block rows, then the resulting model will be unstable.

One option to guarantee stable models is to somehow modify the estimate

$$\hat{\mathcal{O}}_r = U_n \Sigma_n^{1/2}$$

directly. As early as Kung [38], it was proven that an estimated $\mathcal{O}_r$ with zeros in its final block row will guarantee a stable estimate when used in realization-based methods. Maciejowski [42] improves the original proof of Kung and proposes a scheme to induce stability by replacing the block row "lost" when shifting $\mathcal{O}_r$ with a block row of zeros. Alternatively, Van Gestel, Suykens, Van Dooren, and De Moor [73] proposes a method by which Σ_n is modified along a similar line of reasoning to guarantee stable estimates.

Examples of identification methods which incorporate convex-optimization techniques include Lacy and Bernstein [39], in which a linear-matrix inequality (LMI) framework is proposed to constrain the eigenvalues of both N4SID- and MOESP-type subspace system estimates to be stable, and Hoagg, Lacy, Erwin, and Bemstein [25] and McKelvey and Moheimani [45], which use similar LMI frameworks to restrict estimates to positive real systems. In Hoagg, Lacy, Erwin, and Bernstein [26], this framework is extended to provide a lower bound on the natural frequencies of the poles of the identified model, creating a convex optimization procedure which restricts the eigenvalues to lie *outside* a convex region of the convex plane; the parameterization used, however eliminates the possibility of also restricting the eigenvalues to lie *within* convex regions of the complex plane, such as the unit circle.

Additional subspace-based methods that incorporate prior knowledge of the system include Okada and Sugie [55], which develops a method for the case in which some of the pole locations of the system are known beforehand. In Trnka and Havlena [72], constraints were developed to fix the steady-state gain of the system and minimize a form of numerical derivative of the system step response. In Alenany, Shang, Soliman, and Ziedan [3], an equality-constrained quadratic program is used to estimate the lower-block-triangular matrix of Markov parameters seen in the data-matrix equations, which also enforces causality in the Markov parameter estimates. The classical realization algorithm was then applied to a block-Hankel matrix constructed from the system Markov parameters. Constraints were also developed to fix the steady-state gain.

In this chapter, we propose a new framework to impose general eigenvalue constraints for subspace identification problems. The eigenvalue constraints are constructed using the concept of LMI regions, first introduced in Chilali and Gahinet [12], which generalize Lyapunov stability to convex regions of the complex plane. The generality of our method allows for the eigenvalues of the estimate to be constrained to any convex region of the complex plane that can be expressed as the intersection of ellipsoids, parabolas, or half-spaces symmetric about the real axis.

Our approach generalizes the methods proposed in Lacy and Bernstein [39] and Hoagg, Lacy, Erwin, and Bemstein [25], which over-constrain the discrete-time Lyapunov inequalities. We also present a stability criteria which over-constrains the discrete-time Lyapunov inequalities, but it is a special case of a more general method, and our constraint has an exact geometric interpretation in the complex plane. In addition to a stability constraint, we also provide constraints that require eigenvalues to have positive real parts and/or zero imaginary parts. We conclude by proposing some additional time-domain constraints for use in identification.

7.2 LMI Regions

An LMI region is a convex region $\mathcal{D}$ of the complex plane, defined in terms of a symmetric matrix α and a square matrix β, as

$$\mathcal{D} = \{z \in \mathbb{C} : f_{\mathcal{D}}(z) \geq 0\} \tag{7.1}$$

where

$$f_{\mathcal{D}}(z) = \alpha + \beta z + \beta^T \bar{z}. \tag{7.2}$$

We will call $f_{\mathcal{D}}(z)$ for a given $\mathcal{D}$ the characteristic function of $\mathcal{D}$. LMI regions generalize Lyapunov notions of stability for continuous and discrete time systems, and the describing-function parameters α and β may be used to form Lyapunov-type inequalities.

LMI regions were first introduced in Chilali and Gahinet [12], and we repeat the central theorem of the paper here for future reference.

Theorem 7.1. *The eigenvalues of a matrix A lie within an LMI region with characteristic function* (7.2) *if and only if there exists a matrix $P \in \mathbb{R}^{n\times n}$ such that*

$$P = P^T > 0, \qquad \mathcal{M}_{\mathcal{D}}(A,\ P) \geq 0 \tag{7.3}$$

in which

$$\mathcal{M}_{\mathcal{D}}(A,\ P) = \alpha \otimes P + \beta \otimes (AP) + \beta^T \otimes (AP)^T. \tag{7.4}$$

The original definition of an LMI region has $<$ in place of $\geq$ for (7.1) and (7.3). We adopt the above definition instead so that our results are straightforward to implement as a semi-definite program and because the real and imaginary axes cannot be parameterized as LMI regions if (7.1) uses a strict inequality. This change does not affect the proofs of Chilali and Gahinet [12], since they are based on the relationship

$$\left(I \otimes v^H\right) \mathcal{M}_{\mathcal{D}}(A,\ P) \left(I \otimes v\right) = \left(v^H P v\right) f_{\mathcal{D}}(\lambda)$$

where λ is an eigenvalue and v is the corresponding left eigenvector of A. Because P is positive definite, the signs of $\mathcal{M}_{\mathcal{D}}$ and $f_{\mathcal{D}}$ need only to be equal, not necessarily negative.

The intersection of two LMI regions $\mathcal{D}_1$ and $\mathcal{D}_2$ is also an LMI region, described by the matrix function

$$f_{\mathcal{D}_1 \cap \mathcal{D}_2}(z) = \begin{bmatrix} f_{\mathcal{D}_1}(z) & 0 \\ 0 & f_{\mathcal{D}_2}(z) \end{bmatrix}. \tag{7.5}$$

Note that in general the $(\alpha,\ \beta)$ pair that describes an LMI region is not unique.

7.2.1 Some LMI Regions Useful for Identification

In the following we derive some LMI regions useful for identification purposes. Of course the user need not be limited by these; LMI regions can be constructed for any convex intersection of half-spaces, ellipsoids, and parabolas symmetric about the real axis. The following regions are straightforward to verify by algebraically solving for the eigenvalues of (7.2).

Discrete-Time Stable Eigenvalues

Stable system estimates are often desirable in the identification problem. Standard subspace methods, however, do not guarantee stability of the identified model. To provide some known degree of stability for the identified models, we may constrain eigenvalues to the disc of radius $1 - \delta_s$.

Proposition 7.1. *The set*

$$\mathcal{S} = \{z \in \mathbb{C} : |z| \leq 1 - \delta_s,\ 0 \leq \delta_s \leq 1\}$$

is equivalent to the LMI region $f_{\mathcal{S}}(z) \geq 0$*,*

$$f_{\mathcal{S}}(z) = (1 - \delta_s) I_2 + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \bar{z}. \tag{7.6}$$

Theorem 7.1 applied to this region with $\delta_s = 0$ results in

$$P > 0, \qquad \text{and} \qquad \begin{bmatrix} P & AP \\ PA^T & P \end{bmatrix} > 0,$$

which, by means of Schur complements, is equivalent to the familiar discrete-time Lyapunov condition

$$P > 0, \qquad \text{and} \qquad P - APA^T > 0.$$

It is also similar, though not identical, to the LMI constraint proposed in Lacy and Bernstein [39]. In (7.6), however, the relaxation parameter δ_s has a specific interpretation in the complex plane.

Eigenvalues with Positive Real Parts

It is also generally desirable to avoid models with poles that have negative real parts. Such systems cannot be transformed to continuous time, since the matrix logarithm is undefined for A with negative real eigenvalues. Consequently, we wish to construct an LMI region that describes the positive right-half plane. This region should also be parameterized so that the region begins some distance away from the imaginary axis.

Proposition 7.2. *The set*

$$\mathcal{P} = \{z \in \mathbb{C} : \mathrm{Re}(z) \geq \delta_p,\ \delta_p \geq 0\}$$

is equivalent to the LMI region $f_{\mathcal{P}}(z) \geq 0$*,*

$$f_{\mathcal{P}}(z) = \delta_p \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} z + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \bar{z}. \tag{7.7}$$

Eigenvalues with Zero Imaginary Parts

If it is known that a process has strictly real eigenvalues (such as with RC circuits, heat transfer, or many orther processes in thermodynamics), then it may be desirable to constrain the eigenvalues of the estimate to the real number line.

Proposition 7.3. *The real number line* $\mathbb{R}$ *is equivalent to the LMI region* $f_{\mathbb{R}}(z) \geq 0$*,*

$$f_{\mathbb{R}}(z) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \bar{z}.$$

This constraint, however, is computationally unfriendly for many numerical optimization procedures, since it is effectively using two inequalities to define an equality, which can create problems for interior-point-based solvers. Instead, we include a parameter to describe an arbitrarily small band around the real axis in the complex plane.

Proposition 7.4. *The set*

$$\mathcal{R} = \{z \in \mathbb{C} : |\mathrm{Im}(z)| \leq \delta_r,\ \delta_r \geq 0\}$$

is equivalent to the LMI region $f_{\mathcal{R}}(z) \geq 0$*,*

$$f_{\mathcal{R}}(z) = 2\delta_r I_2 + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} z + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \bar{z}. \tag{7.8}$$

The parameter δ_r can be made small enough so that the complex parts of the resulting identified eigenvalues are near machine precision.

Though many more types of LMI regions exist, these three are the most useful for identification purposes. The geometric interpretation of each region in the complex plane is shown in Figure 7.1.

7.3 Realization with Eigenvalue Constraints

In this section, we incorporate LMI regions to create a realization-based identification procedure with eigenvalue constraints. We will develop the method

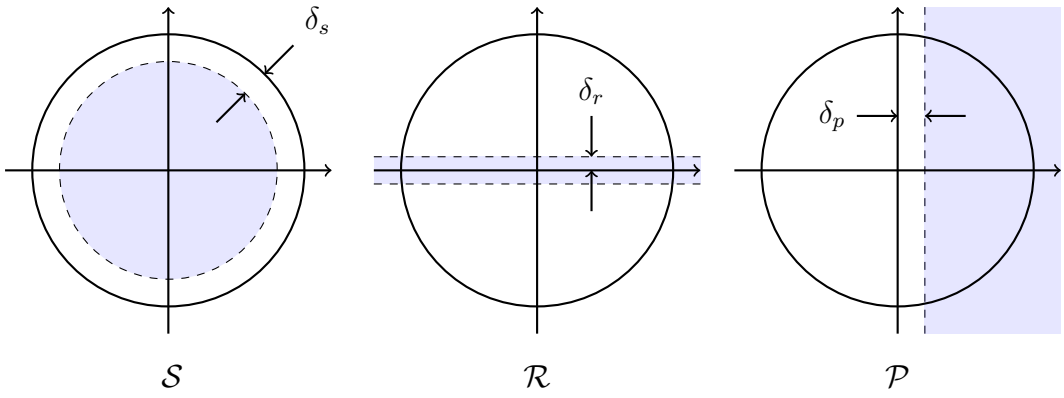

Figure 7.1: Sample LMI Regions in the Complex Plane

for the raw-data case only; it is straightforward to extend the technique to the covariance-based method, and a constrained step-based realization method will be developed in Chapter 8.

The method is based on the observation that the realization procedure minimizes the cost function

$$J_r(\hat{A}) = \left|\left|\hat{\mathcal{O}}_r \hat{A} \hat{X} \Pi_{U^\perp} - Y_{1|r} \Pi_{U^\perp}\right|\right|_F, \tag{7.9}$$

which is affine in the parameter $\hat{A}$. Theorem 7.1 contains the product AP, however, so we augment (7.9) to contain $\hat{A}P$ via a right-hand weighting $W_r = \left(\hat{X}\Pi_{U^\perp}\right)^\dagger P$,

$$\begin{aligned} J_r'(\hat{A},\ P) &= \left|\left|\left(\hat{\mathcal{O}}_r \hat{A} \hat{X} \Pi_{U^\perp} - Y_{1|r}\Pi_{U^\perp}\right) W_r\right|\right|_F \\ &= \left|\left|\left(\hat{\mathcal{O}}_r \hat{A} \hat{X} \Pi_{U^\perp} - Y_{1|r}\Pi_{U^\perp}\right) \left(\hat{X}\Pi_{U^\perp}\right)^\dagger P\right|\right|_F \qquad (7.10) \\ &= \left|\left|\hat{\mathcal{O}}_r \hat{A} P - Y_{1|r}\Pi_{U^\perp}(\hat{X}\Pi_{U^\perp})^\dagger P\right|\right|_F. \qquad (7.11) \end{aligned}$$

Though (7.11) has the same global minimum as (7.9), namely

$$\hat{A} = (\hat{\mathcal{O}}_r)^\dagger Y_{1|r} \Pi_{U^\perp} (\hat{X}\Pi_{U^\perp})^\dagger, \tag{7.12}$$

it does not necessarily have the same minimum over an arbitrary convex set. (This may be seen by calculating the matrix differential in $\hat{A}$ for both costs. See Section B.3.) We must also ensure that P does not become arbitrarily small during the minimization procedure, so we include the constraint

$$\text{trace}(P) = 1$$

to retain numerical stability.

Though (7.11) now contains $\hat{A}P$, it is no longer affine in the parameters $\hat{A}$ and P. We thereby re-parameterize the cost function with an auxiliary term $Q = \hat{A}P$, to create the final convex optimization problem:

Given estimates $\hat{\mathcal{O}}_r$ and $\hat{X}\Pi_{U^\perp}$, a data matrix $Y_{1|r}\Pi_{U^\perp}$, and an LMI region parameterized by α and β,

$$\begin{aligned} \text{minimize} \quad & J_c(Q,\ P) \\ \text{subject to} \quad & \mathcal{M}(Q,\ P) \geq 0, \\ & P = P^T > 0 \\ & \text{trace}(P) = 1 \end{aligned} \tag{7.13}$$

in which

$$J_c(Q,\ P) = \left|\left|\hat{\mathcal{O}}_r Q - Y_{1|r}\Pi_{U^\perp}(\hat{X}\Pi_{U^\perp})^\dagger P\right|\right|_F,$$

and

$$\mathcal{M}(Q,\ P) = \alpha \otimes P + \beta \otimes Q + \beta^T \otimes Q^T.$$

Similar minimization problems may be formulated for all realization-based identification methods. A method for step-based realization procedures was developed in Miller and de Callafon [51]. In Miller and de Callafon [54], a MOESP-type procedure was developed, in which it was also shown that methods for which the weighting reduces to $W_r = P$ will still modify the non-global minima.

At this point we should remark that although the global minimizer (7.12) might be in the set of feasible points, numerical optimization tools may not be able to find it exactly. Optimization routines based on primal-dual gap methods [8] may deviate from (7.12) even when it is feasible and supplied as an initial value. This is because, although the analytic solution to primal and dual problems is the same, the numerical solution might not be. Such numerical difficulties become more common as the row dimension of $\hat{\mathcal{O}}$ becomes very large. In practice, it is best to confirm that the eigenvalues of (7.12) do not satisfy the LMI region's characteristic equation before solving the convex optimization problem.

7.4 Constrained Least-Squares Identification of Input Parameters

In addition to constraining the estimation of A and C in the realization procedure, B and D may be constrained to conform with *a priori* system knowledge as well in the least-squares identification of the input parameters. In the following, we give several useful constraints for identification.

7.4.1 Constrained Steady-State Gain

If A and C are known, then B and D are not only linear in the input-output data, but linear in the steady-state system gain

$$K_{ss} = C(I - A)^{-1}B + D.$$

Thus if the steady gain K_{ss} is known beforehand, then, given estimates $\hat{A}$ and $\hat{C}$, we may express constraints on estimates $\hat{B}$ and $\hat{D}$ as

$$\begin{aligned}
\operatorname{vec}(K_{ss}) &= \operatorname{vec}\left(\hat{C}(I - \hat{A})^{-1}\hat{B} + \hat{D}\right) \\
&= \left[I_{n_u} \otimes \hat{C}(I - \hat{A})^{-1}\right] \operatorname{vec}(\hat{B}) + \operatorname{vec}(\hat{D}) \\
&= \begin{bmatrix} I_{n_u} \otimes \hat{C}(I - \hat{A})^{-1} \\ I_{n_u n_y} \end{bmatrix} \begin{bmatrix} \operatorname{vec}(B) \\ \operatorname{vec}(D) \end{bmatrix}
\end{aligned}$$

These may be included as equality constraints in the least-squares identification procedure when solving (4.34).

7.4.2 Constrained Impulse-Response and Step-Response Behavior

Many of the underlying physical laws that motivate the eigenvalue constraints result in similar constraints on the overall step-response behavior of a system. When a system is constrained to have strictly real eigenvalues, then it is often assumed that the step response should not undershoot its initial value or overshoot its final value. The zeros that result when estimating B and D, however,

may cause just such behavior, even when the eigenvalues of A are constrained to be real.

Because the impulse response is the derivative of the step response, the condition that a step-response have no undershoot and no overshoot is equivalent to the condition that the impulse response coefficients are all positive. Thus the constraint

$$\hat{C}\hat{A}^t\hat{B} > 0 \quad \forall \quad t \tag{7.14}$$

is necessary and sufficient to guarantee such behavior.

Unfortunately, this cannot be numerically enforced for all t, so select values of t must be used to create a numeric constraint. In practice, this condition is usually desired in situations where the eigenvalues of A are assumed to be strictly real, so each eigenvalue of A corresponds to a "time-constant" that may be used to selectively choose t. If a system has n time-constants τ_i, then applying the constraint (7.14) at $n+1$ points t_i such that $t_i < \tau_i$ and $t_{n+1} > \tau_n$ appears sufficient to guarantee such behavior, though it has not yet been proven by the author.

7.5 Acknowledgements

This chapter, in part, includes content from the following publications:

D.N. Miller, J. Hulett, J. McLaughlin, and R.A. de Callafon, "Thermal Dynamical Identification of LEDs by Step-Based Realization and Convex Optimization," submitted to *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 2012.

D.N. Miller and R.A. de Callafon, "Subspace Identification with Eigenvalue Constraints," to appear *Automatica*, 2012.

D.N. Miller and R.A. de Callafon, "Step-Based Realization with Eigenvalue Constraints," to appear, *16th IFAC Symposium on System Identification*, Brussels, Belgium: July 2012.

8 Application II: Transient Thermal Response of an LED

In the second application chapter, the step-based realization procedure is combined with the proposed constraints to identify the thermal response of a high-power LED.

8.1 Introduction

Heat-transfer in powered electronics typically requires very little feedback control. Usually the only requirement is that the temperature stay below a critical threshold at which damage from heat becomes a concern, since changes in temperature have a negligible impact on performance below this point. Feedback is primarily used to engage additional cooling systems to ensure that the system temperature stay within a safe operating range, such as activating additional fans inside a computer case when the processor temperature exceeds a preset level.

Though applications for feedback control may be uncommon, experimental analysis of thermal transients plays an important role in the model validation and failure detection of integrated circuits. Manufacturing errors can result in an increase in thermal impedance between circuit elements that dramatically shortens the operating life of the system. This includes decreases in material conductance and changes in heat capacitance due to faulty components, as well as increases in contact resistance due to faulty connections. Measuring the transient thermal response of a circuit is an effective means of failure detection in the manufacturing process. The response is compared to a baseline response of a system that has

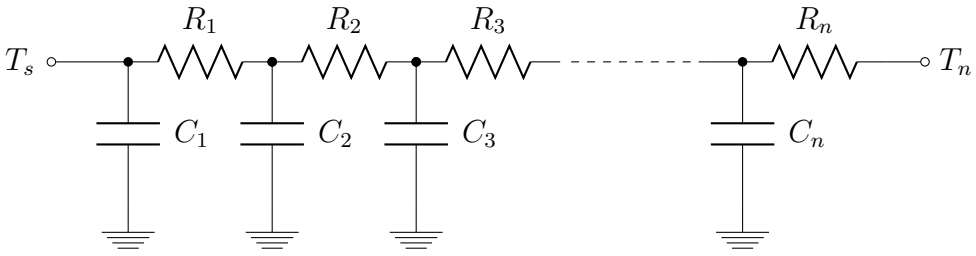

Figure 8.1: RC ladder circuit (Cauer network topology).

been confirmed to be operating within specifications, and the test system fails if the two responses do not match.

Such thermal systems are typically modeled as resistor-capacitor ladder circuits, with the various resistances and capacitances derived from known or measured thermal properties of the materials. These networks are most often created from analytical models or numerical simulations. An example of such an RC-ladder is shown in Figure 8.1. The resistors represent impedance to the flow of heat while the capacitors represent the storage of thermal energy in each component. The temperature T_s is the internal heat source, and T_n is the temperature at the point of measurement. The ground represents the surrounding ambient temperature. The Cauer network topology shown in Figure 8.1 is the most common, though variations, such as Foster topologies, are used occasionally [70].

These RC ladders are often converted into a *time-constant spectrum* in which each RC pair is converted into a pair of a time constant and the relative contribution of the time constant. The spectrum is then plotted with the time constants on the x-axis and the contributions on the y-axis.

Modeling heat transfer in such a circuit topology constrains the dynamic behavior of the model to be consistent with the laws of thermodynamics. Energy-storing elements, such as inductors, would allow for the measured temperature T_n to temporarily exceed the source temperature T_s, which would violate the law of entropy. To ensure that experimentally constructed models share these qualities, identification methods are typically limited to "white-box" approaches that estimate values for the resistances R_i and capacitances C_i separately. These methods

are limited to the identification of models of very low order, often no more than three RC pairs [67].

Although models such as the circuit in Figure 8.1 are commonly used in the design process, few tools exist to produce experimental models of this form. One popular framework for analyzing the thermal response of electronics is that of Székely and Van Bien [70]. In this method, the finite-dimensioned resistor-capacitor network in Figure 8.1 is replaced with an infinitely long network to form a continuum of time-constants as a function of a logarithmic time variable. The step response is formulated in terms of the convolution of an analytic function with a unit exponential rise. The original method consists of a transformation to a logarithmic time variable, followed by a differentiation and a de-convolution to generate a non-parametric estimate of the spectrum function. Relationships between the total impedance of a circuit and its time-constant spectrum were later derived in Székely [68] and Székely and Rencz [69].

While an automatically generated graph of the time-constant spectrum provides an immediate and intuitive interpretation of the various transient artifacts of the step response, an additional curve-fitting or discretization procedure is required in order to define the spectrum with a finite number of parameters. Hence estimating the time-constant spectrum in this way is essentially a non-parametric identification procedure, akin to measuring the frequency response function of a linear system. Also, the calculation of the spectrum requires differentiation of the measured step response followed by de-convolution with a low-pass filter, both of which reduce the signal to noise ratio of the data.

In Székely [66], a weighted-discretization method was proposed which reduced the number of observed time-constants based on equidistant spacing along the logarithmic time axis, but the number of time constants derived was between 60 and 120. In Székely [67], filtering techniques were developed to address loss of resolution in the time-constant spectrum when two known constants are relatively close to each other. In Rencz, Poppe, Kollár, Ress, and Székely [60], the introduction of materials properties was proposed to correct for model discrepancies due to parasitic heat loss.

Other approaches to modeling the thermal response from experimental data include applying a constrained, non-linear, least-squares curve-fitting method to minimize the root-mean-square (RMS) error for a continuous-time parameterization of a step response, developed in Palacín, Salleras, Samitier, and Marco [57]. The method was applied to multivariable data, in which the effects of a step on each input were measured for multiple output signals, forming a step-response matrix of transients. The method requires careful model selection beforehand and an initial guess as to the value of the step-response time constants to avoid becoming trapped in local minima. Genetic algorithms were applied to a similar problem in Palacín, Salleras, Puig, Samitier, and Marco [56].

In Bechtold, Rudnyi, and Korvink [6], automatic model reduction for deterministic electro-thermal models of semiconductor devices was studied. A balanced-truncation method for reducing the order of linear systems, which shares some qualities with the identification method to be presented in this chapter, was shown to be an effective way of reducing model order but was concluded to be computationally unattractive due to the extremely large order of the deterministic models. The application of model reduction to systems identified from simulated or measured data was not studied.

The step-based realization procedure presented in this dissertation is a significant departure from existing methods in the manufacturing and process control literature. This new method has a number of key advantages that separate it from existing methods. (i) The model order is chosen during the identification procedure, and no prior assumptions must be made regarding the number or location of time constants. (ii) The relationship of the algorithm to model-reduction procedures inherently produces low-order models. (iii) The step response can be possibly vector valued. Multivariable measurements do not necessarily increase the system order. (iv) No differentiation of the step-response is required, allowing for application to measurements with low signal-to-noise ratios. (v) The steady-state value of the step response may optionally be constrained during identification. (vi) The method involves no non-convex optimization. The calculation of the model parameters requires only standard operations of linear algebra, a linear program-

ming problem, and a quadratic programming problem, for which robust numerical tools are widely available.

The goal of this application is to produce models of a specific structure to match existing industry standards. Until this point, all models presented in this dissertation have been in discrete time; the industry standard, however, is in continuous time, and so some of the discrete-time LTI equations are re-derived from an initial continuous time model. The resulting algorithm, however, is no different than the step-based realization procedure seen in Chapter 3, but with the addition of convex constraints.

8.2 Problem Formulation

We begin with the assumption that the true noise-free step response has the form

$$y_c(t_c) = \sum_{i=1}^{n} R_i(1 - e^{-t_c/\tau_i}) \tag{8.1}$$

where $t_c \in \mathbb{R}$ is a continuous time variable, $y_c(t_c) \in \mathbb{R}^{n_y}$ is a possibly vector-valued signal, τ_i are the time constants of the response, and $R_i \in \mathbb{R}^{n_y \times 1}$ are the contribution of each time constant to the total response. The R_i in (8.1) are not equivalent to the resistances in Figure 8.1, but (8.1) is the most common notation and is used for the rest of chapter. The pairs $(\tau_i,\ R_i)$ form the *time constant spectrum*. Henceforth, we will focus our efforts on identifying the parameters in (8.1), ignoring the network topology interpretation.

We assume the system is stable so that the step response is bounded, and thus

$$\mathrm{Re}(\tau_i) > 0. \tag{8.2}$$

Additionally, allowing for complex τ_i could result in an oscillatory response. This would imply a possible overshoot in temperature during dissipation, which would violate the laws of thermodynamics. Hence we assume

$$\mathrm{Im}(\tau_i) = 0. \tag{8.3}$$

The alternative non-parametric step-response model proposed by Székely defines the time-constant spectrum as an analytical function $R(\tau)$ derived from the convergence of the $(R_i,\ \tau_i)$ pairs as $\tau_{i+1} - \tau_i \to 0$. The step response is then given by the convolution integral

$$y_c(t_c) = \int_{-\infty}^{\infty} R(\zeta)(1 - \exp(-t_c/e^{\zeta})d\zeta. \tag{8.4}$$

The two definitions, though conceptually similar, are not interchangeable, since the integral (8.4) is always 0 if $R(\tau)$ is bounded and nonzero for a finite number of τ, which is the case for (8.1).

8.2.1 Discretization of the Step-Response Model

Because (8.1) is a finite-dimensional, linear, time-invariant system, it has an alternative state-space representation

$$\begin{aligned} \dot{x}_c(t_c) &= A_c x_c(t_c) + B_c \\ y(t_c) &= C_c x(t_c) \end{aligned} \tag{8.5}$$

in which $x_c(t_c) \in \mathbb{R}^n$ is the state of the system; and $A_c \in \mathbb{R}^{n\times n}$, $B_c \in \mathbb{R}^{n\times 1}$, and $C_c \in \mathbb{R}^{n_y \times n}$ are the state-space parameters. The initial state is $x_c(0) = 0$, and the subscript c denotes that the parameters are from a continuous-time model. We assume that (8.5) is controllable, observable, and minimal, so that the state dimension n cannot be reduced. These assumptions are consistent with (8.1). The step-response $y_c(t_c)$ may be expressed in terms of (8.5) as

$$y_c(t_c) = C_c \int_0^{t_c} e^{A_c(t_c-\tau)} B_c d\tau = C_c A_c^{-1}(e^{A_c t_c} - I)B_c, \tag{8.6}$$

where $e^{(\cdot)}$ is the matrix exponential.

Suppose A_c has the eigenvalue decomposition

$$A_c = V_c \Lambda_c V_c^{-1}.$$

If $\lambda_c^{(i)}$ is an eigenvalue of A_c and $v_c^{(i)}$ its associated eigenvector, then (8.1) may be derived from (8.5) via the identities

$$\tau_i = -1/\lambda_c^{(i)}, \tag{8.7}$$

$$R_i = \tau_i C_c M_c^{(i)} B_c. \tag{8.8}$$

$M_c^{(i)}$ is the rank-1 matrix

$$M_c^{(i)} = v_c^{(i)} (V_c^{-1})_{(k,:)} \in \mathbb{R}^{n \times n}, \tag{8.9}$$

in which "MATLAB-style" indexing notation has been used in the subscript. Note that $M_c^{(i)}$ is strictly real because τ_i is real.

Suppose $y_c(t_c)$ is measured in discrete-time with a sampling rate of T_s. Let $y_d(t) \triangleq y_c(T_s t)$ be the discrete-time step-response measurement, assuming a zero-order hold for each sample. Letting the subscript d denote discrete-time state-space parameters, the discrete-time equivalent of (8.5) is then

$$\begin{aligned} x_d(t+1) &= A_d x_d(t) + B_d \\ y_d(t) &= C_d x_d(t) \end{aligned} \tag{8.10}$$

and the discretization of (8.6) results in the familiar Markov parameters

$$y_d(t) = \begin{cases} 0 & t = 0, \\ \displaystyle\sum_{l=0}^{t} C_d A_d^{l-1} B_d & t > 0, \end{cases} \tag{8.11}$$

where

$$A_d = e^{A_c T_s}, \quad B_d = A_c^{-1}(A_d - I)B_c, \quad \text{and} \quad C_d = C_c.$$

Because our data is measured in discrete time, we first identify the parameters $(A_d,\ B_d,\ C_d)$, convert them to continuous time to find $(A_c,\ B_c,\ C_c)$, and finally use the identities (8.7) and (8.8) to convert the model to the form of (8.1). The state-space models (8.5) and (8.10) allow for a great deal more freedom than (8.1), however, so we incorporate constraints to ensure that the model constructed from the step-based realization can be transformed into the form (8.1).

8.3 Constraining the Estimate

In this section, we derive constraints needed to guarantee that the identified model has an equivalent form (8.1). We then show how the step-based realization procedure described previously may be combined with the constraints in Chapter 7 to produce two sequential convex optimization problems.

8.3.1 Eigenvalue Constraints

If (8.10) is the discretization of (8.1), then the assumptions of τ_i require constraints to be placed on the eigenvalues of A_d. These are stated in the following.

Proposition 8.1. *The system* (8.10) *is a discretization of* (8.1) *if and only if all eigenvalues* $\lambda_d^{(i)}$ *of* A_d *satisfy the following constraints:*

$$|\lambda_d^{(i)}| < 1, \tag{8.12}$$

$$\mathrm{Re}(\lambda_d^{(i)}) > 0, \tag{8.13}$$

$$\mathrm{Im}(\lambda_d^{(i)}) = 0. \tag{8.14}$$

Proof. Noting that $\lambda_d^{(i)} = e^{-T_s/\tau_i}$, (8.12) and (8.14) follow directly from (8.2) and (8.3), respectively. (8.13) is required so that the zero-order hold discretization formulas are invertible. □

Thus in order for (8.10) to be converted to (8.1), the constraints (8.12), (8.14), and (8.13), must be satisfied. The LMI region corresponding to these constraints is the intersection of the regions $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{P}$ in Propositions 7.1, 7.2, and 7.4, respectively, shown in Figure 8.2, with $\delta_s = \delta_r = \delta_p = 0$. Combining Theorem 7.1 with the identity (7.5), we formulate an LMI region to satisfy all three constraints on $\lambda_d^{(i)}$ and state the following Lyapunov-type condition for $\lambda_d^{(i)}$, which may be incorporated into a convex optimization program.

Corollary 8.1. *The discrete-time state-space model* (8.10) *is the discretization of a continuous-time model* (8.1) *if and only if there exists a matrix* $P \in \mathbb{R}^{n\times n}$ *such that*

$$P = P^T \succ 0, \qquad M_d(A_d,\ P) \succeq 0$$

in which

$$\mathcal{M}_d(A_d,\ P) = \alpha_d \otimes P + \beta_d \otimes (A_d P) + \beta^T \otimes (A_d P)^T \tag{8.15}$$

$$\alpha_d = \begin{bmatrix} (1-\delta_s)I_2 & 0 & 0 \\ 0 & 2\delta_r I & 0 \\ 0 & 0 & 2\delta_p I_2 \end{bmatrix}$$

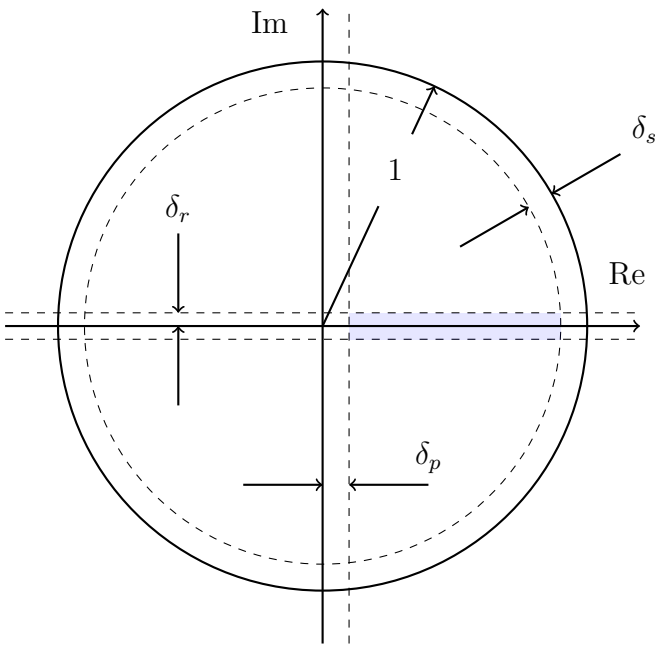

Figure 8.2: LMI region used for identification.

$$\beta_d = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

with $\delta_s = \delta_r = \delta_p = 0$.

The corollary results in sufficient conditions for A_d for any $\delta_s,\ \delta_p \in [0,\ 1)$, and we may still use a nonzero δ_r, provided any remaining imaginary components of $\lambda_d^{(i)}$ are neglected. In practice δ_r can be very small so that the imaginary components are near machine precision.

8.3.2 Additional Constraints on the Discrete-Time State-Space Parameters

Additional assumptions may be made regarding (8.1), such as the assumption that steady-state value of the step-response is equal to some value determined from prior knowledge. For example, if the ambient temperature or steady-state behavior of the system is known with high precision, then the value of the sum of R_i can be assumed beforehand.

Suppose $y_c(\infty)$ is the steady-state value of (8.1). Because the continuous- and discrete-time systems have the same steady-state value, we may form the following constraint:

Proposition 8.2. *If the system* (8.1) *satisfies*

$$\sum_{i=1}^{n} R_i = y_c(\infty), \tag{8.16}$$

then (8.10) *is a discretization of* (8.1) *if and only if*

$$C_d \left(I - A_d\right)^{-1} B_d = y_c(\infty). \tag{8.17}$$

In thermal analysis, it is generally required that the transient thermal response to a step change be monotonically increasing. It is therefor necessary that its discretization also be monotonically increasing, and so the difference between one time step ahead and the current time step must be positive. From the discrete-time step response (8.11), this may be stated as the following inequality constraint.

Proposition 8.3. *If* $y(t)$ *in* (8.1) *is monotonically increasing, then* (8.10) *is a discretization of* (8.1) *only if*

$$C_d A_d^k B_d > 0 \ \forall \ k \tag{8.18}$$

At times it is possible to assume that all R_i in (8.1) are positive. This may also be translated into a constraint on the discrete-time state-space model.

Proposition 8.4. *If the system* (8.1) *satisfies*

$$R_i > 0,$$

then (8.10) *is a discretization of* (8.1) *if and only if*
(i) A_d *satisfies* (8.14) *and* (8.13)
(ii)

$$(\Lambda_c^{-1} \otimes C_d) \begin{bmatrix} M_c^{(1)} \\ M_c^{(2)} \\ \vdots \\ M_c^{(n)} \end{bmatrix} (I - A_d)^{-1} A_c B_d > 0, \tag{8.19}$$

where (using the matrix logarithm)

$$A_c = \frac{1}{T_s} \log(A_d), \tag{8.20}$$

$A_c = V_c \Lambda_c V_c^{-1}$ *is the eigenvalue decomposition,* $M_c^{(i)}$ *is given by* (8.9)*, and* $\otimes$ *is the Kronecker product.*

Proof. From (8.8), we know that R_i are real if and only if $\mathrm{Im}(\lambda_d^{(i)}) = 0$. Also, (8.20) exists if and only if $\mathrm{Re}(\lambda_i^{(i)}) > 0$. Then using (8.8), (8.19) reduces to

$$\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} > 0$$

which is the required constraint. □

8.3.3 Incorporating Constraints into the SBR Method

The step-based realization procedure minimizes the cost function (3.12), which is affine in $\hat{A}$. The LMI constraints, however, contain the product $\hat{A}P$, so we modify cost function with a right-hand weighting $W_r = (\hat{\mathcal{C}}_l U_p)^\dagger P$ to obtain

$$\begin{aligned} J_s'(\hat{A}) &= \left|\left| \left(\hat{\mathcal{O}}_k \hat{A} \hat{\mathcal{C}}_l U_p - Y_{1|k} + M' \right) W_r \right|\right|_F \\ &= \left|\left| \hat{\mathcal{O}}_k \hat{A} P - (Y_{1|k} + M')(\hat{\mathcal{C}}_l U_p)^\dagger P \right|\right|_F \end{aligned} \tag{8.21}$$

Thus, we may formulate a convex optimization procedure similar to the case used for raw data in Chapter 7. Note that the unconstrained minimizer of (8.21) is only

the constrained minimizer if $P = I$. To reduce any errors this may cause, as well as to increase the numerical stability of the problem, we provide as a constraint

$$\text{trace}(P) = n,$$

which still allows for the possibility of $P = I$ while not over-constraining the problem.

We still must re-parameterize (8.21) to be affine in the parameters in order to formulate the constrained optimization as a convex optimization. Letting $Q = \hat{A}P$, we form the following convex optimization problem with convex constraints:

Given estimates $\hat{\mathcal{O}}_k$ and $\hat{\mathcal{C}}_l U_p$,

$$\begin{aligned} \text{minimize} \quad & J(Q,\ P) \\ \text{subject to} \quad & \mathcal{M}(Q,\ P) \succeq 0, \\ & P = P^T \succ 0 \\ & \text{trace}(P) = n \end{aligned} \tag{8.22}$$

in which

$$J(Q,\ P) = \left|\left|\hat{\mathcal{O}}_k Q - (Y_{1|k} + M')(\hat{\mathcal{C}}_l U_p)^\dagger P\right|\right|_F,$$

and

$$\mathcal{M}(Q,\ P) = \alpha_d \otimes P + \beta_d \otimes Q + \beta_d^T \otimes Q^T.$$

Once Q and P are solved for, let $\hat{A}_d = QP^{-1}$.

With $\hat{C}_d$ taken from the first block-row of $\hat{\mathcal{O}}_k$, the constraints (8.17), (8.18), and (8.19) are linear in $\hat{B}_d$ and can be incorporated into the linear-least squares solution for $\hat{B}$ and $\hat{D}$ in the step-based realization method, forming a second convex optimization problem with convex constraints:

Given

$$z = \begin{bmatrix} y_d(1) \\ \vdots \\ y_d(N-1) \end{bmatrix} \in \mathbb{R}^{n_y N} \qquad \text{and} \qquad \phi^T = \begin{bmatrix} \hat{C}_d \\ \hat{C}_d \hat{A}_d \\ \hat{C}_d \hat{A}_d^2 \\ \vdots \\ \hat{C}_d \hat{A}_d^{N-2} \end{bmatrix},$$

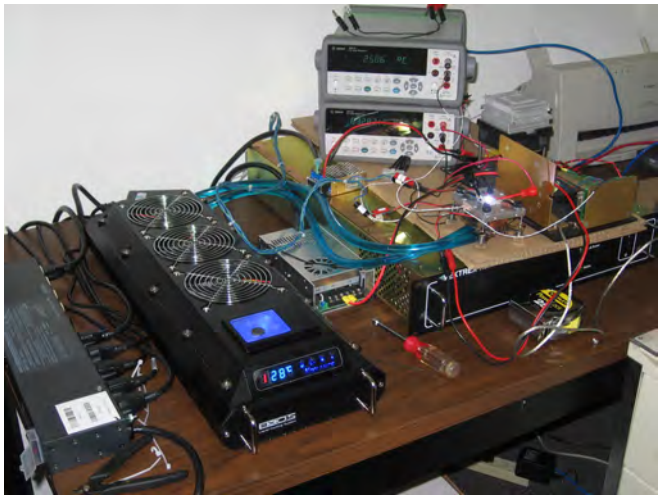

Figure 8.3: Experimental setup for measuring the thermal response of the LED.

$$\begin{aligned} \text{minimize} \quad & \left|\left| z - \phi \hat{B}_d \right|\right|_2 \\ \text{(optionally) subject to} \quad & (8.17), (8.18), \text{ and/or } (8.19). \end{aligned} \tag{8.23}$$

Thus by solving two convex optimization problems, an estimate of a discrete-time state-space system (8.10) which is guaranteed to be a discretization of a model of the form (8.1) is found.

8.4 Analysis of the Thermal Response of an LED

To demonstrate its effectiveness, the proposed algorithm was applied to the measured thermal response of an LED. Experiments were performed at Vektrex in San Diego, CA. A SpikeSafe 200 precision pulsed current source was used to step the current from 0 to the drive current of the device. The voltage was measured with an Agilent34411A 6 1/2 Digit DMM connected to a prototype Vektrex Thermal Platform Controller. The temperature of the device was extrapolated from voltage measurements using the Electronic Industries Association EIA/JEDEC JESD51-1 specification. The LED was mounted to a thermal platform that kept the case temperature constant at 25° C. The full test apparatus may be seen in Figure 8.3.

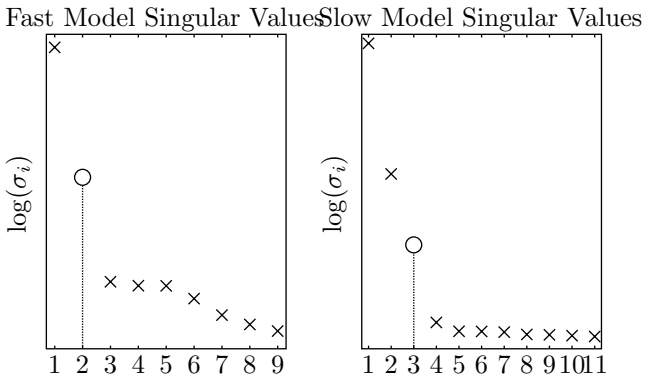

Figure 8.4: Singular values of $Y_{1|k} - M$ for fast and slow datasets.

8.4.1 Analysis of the Response

Because the system itself contains both fast and slow dynamics, the system is "stiff" in the sense that the underlying differential equations are ill-conditioned. Thus to keep the identification procedure computationally feasible, two models were identified. The first was a "fast" model identified from 0 to 0.002 seconds (the first 100 samples of data). The number of block-rows for the data matrices was chosen to be 9. The second model was a "slow" model identified from a dataset down-sampled by a factor of 1,000. Before down-sampling, the data filtered forward and backward through a 4^{th}-order Butterworth filter with a cutoff frequency of 0.1 rad/s. The number of block rows for these data matrices was chosen to be 11.

Singular values of $Y_{1|k} - M$ for each dataset are shown in Figure 8.4. For each, there is a sharp drop off in magnitude within the first few singular values. This implies the order of the underlying dynamics. It also suggests that increasing the model order any higher would not significantly reduce the error between the data and the model. Additionally, increasing the model order would result in $(R_i,\ \tau_i)$ pairs that show signs of linear dependence on one another, so that the number of time constants would not be minimal.

Estimates of $\hat{A}_d$ and $\hat{C}_d$ were found for fast and slow models by solving (8.22) with YALMIP [41] using SDPT3 [71] as the selected solver. $\hat{B}_d$ was solved for each model using the unconstrained linear least-squares solution. Note that these estimates of B_d are only used to simulate the fast and slow models and are

not used to construct the final model. Simulations of both models together with raw data are shown in Figure 8.5.

To combine the two models into one, the sampling time of the slow model was adjusted to match the fast model, and the total model A_d and C_d were taken to be

$$A_d^{(\text{total})} = \begin{bmatrix} A_d^{(\text{fast})} & 0 \\ 0 & A_d^{(\text{slow})} \end{bmatrix}, \qquad C_d^{(\text{total})} = \begin{bmatrix} C_d^{(\text{fast})} \\ C_d^{(\text{slow})} \end{bmatrix}.$$

$B_d^{(\text{total})}$ was found by solving (8.23) with constraints (8.17) and (8.18) using 150 data points space equidistantly along the logarithmic time axis. The steady-state value $y_c(\infty)$ was constrained to be the average of the last 400,000 samples. For (8.18), 150 values of $C_d A_d^k$ were calculated corresponding to the 150 data points. Values of $C_d A_d^k$ which had an entry of magnitude less than 0.01 were discarded to prevent numerical issues, resulting in 49 total inequality constraints of the form (8.18).

A simulation of the total model is shown in Figure 8.5. The time constant spectrum of the model is shown underneath. Of note is that the entire model is described by only 10 parameters and required no nonlinear optimization.

8.5 Acknowledgements

The experimental measurements from LED data were provided by Jeff Hulett and James McLaughlin of Vektrex Corporation in San Diego, CA.

This chapter contains material presented in the following publications:

D.N. Miller, J. Hulett, J. McLaughlin, and R.A. de Callafon, "Parametric Estimation of Step Responses Applied to Power Electronics," submitted to *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 2012.

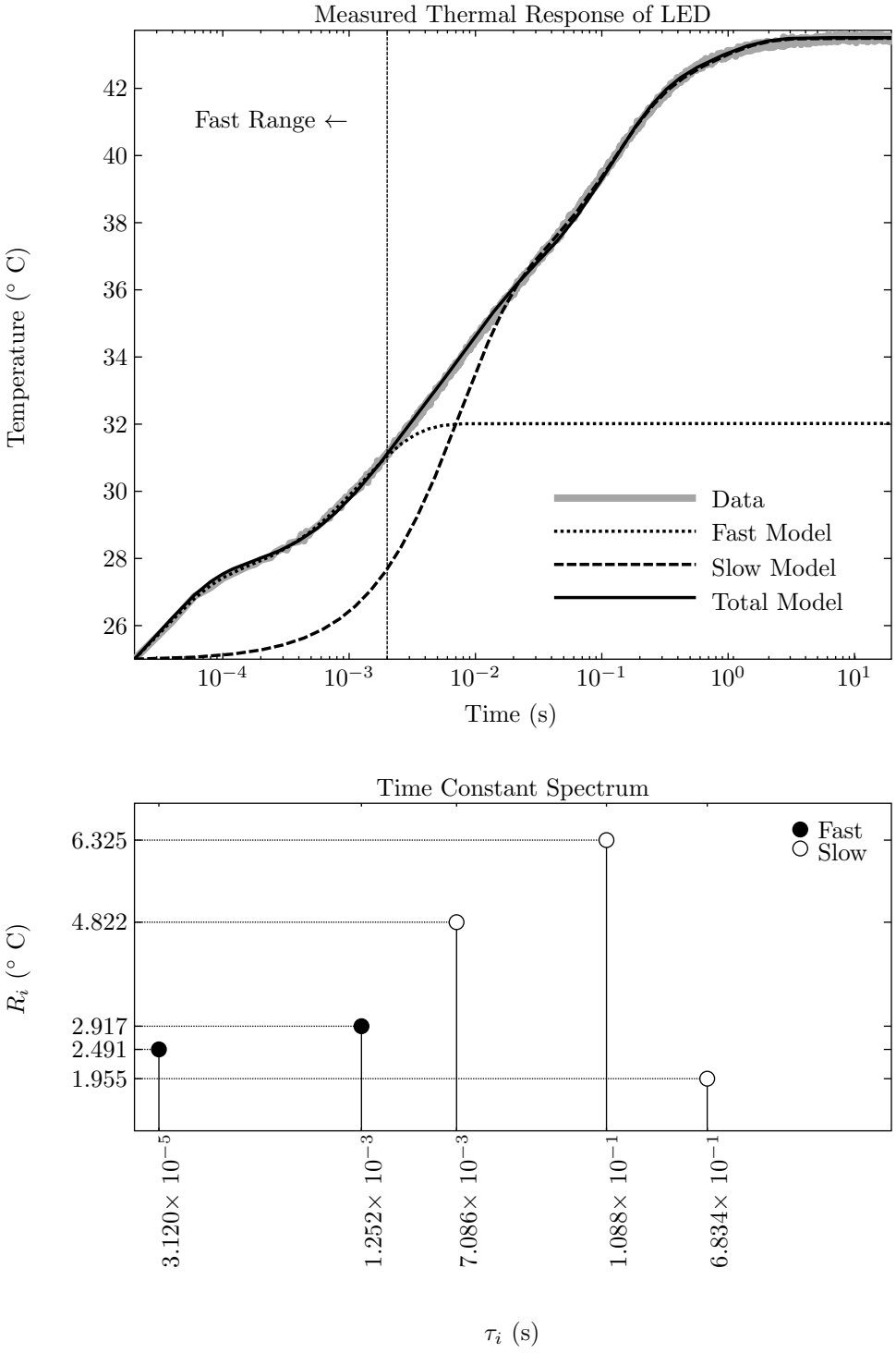

Figure 8.5: Simulations of fast, slow, and total models.

9 Conclusions and Future Work

In this dissertation, a realization-theoretic framework was developed for the identification of discrete-time, linear, time-invariant systems. Beginning with the realization of a state-space system from a measured impulse response, the algorithm was extended to step responses, arbitrary input-output data sequences, and finally covariance function estimates. Convex constraints were developed to transform the realization-based identification algorithms into convex optimization problems to produce models that meet predefined behavioral specifications; the eigenvalues of the models may be constrained to lie within various regions of the complex plane, and the step-response behavior of the models may be constrained to have no undershoot or overshoot.

The covariance-based realization algorithm (COBRA) was shown to be effective in identifying models from experimental data containing many samples with large signal dimensions. Low-order models of aero-servo-elastic dynamics were constructed from noisy data taken from in-flight experiments. The step-based realization procedure was also shown to be effective in analyzing the thermal response of semiconductor devices.

Though convergence was shown to be guaranteed for the input-output data and covariance-based methods, expressions for variance of the parameter estimates were not established. Indeed, usable uncertainty bounds on parameter estimates remains the great white whale for researchers in the field of subspace identification. Although some extension and simplification of existing results were derived, the realization-based methods presented in this dissertation faired no better than existing methods in revealing statistical behavior of the estimates. It may be that identification methods based on the decomposition of structured matrices of data

will always lack useful estimates of parameter variance due to the large number of tunable parameters in the algorithms themselves. There are simply too many indices to analyze the asymptotic properties of them all at once.

The COBRA method works very well for applications with extremely large datasets. In situations such as the AAW experiments, more theoretically robust methods become numerically intractable. As the costs of sensors and data acquisition systems decreases, we may expect the amount of data from an individual experiment to increase. This is particularly true for the analysis of structural dynamics and vibrations, such as the estimation of flutter dynamics.

From experience analyzing the AAW data with the COBRA method, it appears that estimation of the input parameters in the subsequent linear-least-squares step is perhaps the most difficult part of the identification problem. This topic is usually overlooked in the subspace literature, but unjustifiably so. It is intuitively more difficult to identify the zeros of a system than to identify its poles, since the amplitude of the deterministic signal is at its highest at pole frequencies, but at its lowest at zero frequencies. Use of the COBRA pole estimates in a covariance-based orthogonal-basis-function approach is a potential area of investigation, since these methods focus exclusively on the identification of zeros once initial pole estimates are established.

Finally, the constrained step-based identification procedure may be the most useful result of this dissertation. Identification from step-response measurements cannot guarantee convergence regardless of the method used, so the lack of estimates for parameter variance is irrelevant. The method appears to be a very dramatic improvement to existing step-based identification methods, and it is unfortunate that more in-field application data was unavailable to demonstrate the potential of the algorithm.

A Kronecker's Theorem

The following version of Kronecker's theorem appears in [20].

Theorem A.1. *Suppose $G(z) : \mathbb{C} \to \mathbb{C}$ is an infinite series of descending powers of z, starting with z^{-1},*

$$G(z) = g_1 z^{-1} + g_2 z^{-2} + g_3 z^{-3} + \cdots = \sum_{k=1}^{\infty} g_k z^{-k}. \tag{A.1}$$

Assume $G(z)$ is analytic (the series converges) for all $|z| < 1$. Let H be an infinitely large matrix with of the form

$$H = \begin{bmatrix} g_1 & g_2 & g_3 & \cdots \\ g_2 & g_3 & g_4 & \cdots \\ g_3 & g_4 & g_5 & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{A.2}$$

Then H has finite rank n if and only if $g(z)$ is a strictly proper coprime rational function of degree n with poles inside the unit circle. That is, $g(z)$ has an alternative representation

$$g(z) = \frac{b(z)}{a(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0}, \tag{A.3}$$

in which $m < n$, all roots of $a(z)$ satisfy $|z| < 1$, $a(z)$ and $b(z)$ have no common roots, and we have assumed without loss of generality that $a(z)$ is monic.

To prove Theorem A.1, we first prove the following:

Theorem A.2. *The infinitely large matrix H is of finite rank n if and only if there exists a finite sequence $\alpha_1,\ \alpha_2, \cdots, \alpha_n$ such that for $k \geq n$,*

$$g_{k+1} = \sum_{j=1}^{n} \alpha_j g_{k-j+1}, \tag{A.4}$$

and n is the smallest number with this property.

Proof. Let H_k be the row of H beginning with g_k. If H has rank n, then the first $n+1$ rows of H are linearly dependent. This implies that for some $1 \leq p \leq n$, H_{p+1} is a linear combination of $H_1, \ldots, H_p$, and thus there exists some sequence α_k such that

$$H_{p+1} = \sum_{j=1}^{p} \alpha_j H_{p-j+1}. \tag{A.5}$$

The structure and infinite size of H imply that such a relationship must hold for all following rows of H below row p, so that for $q \geq 0$

$$H_{q+p+1} = \sum_{j=1}^{p} \alpha_j H_{q+p-j+1}.$$

Hence for any row H_k, $k > p$, can be expressed as a linear combination of the previous p rows. Since H has at least n linearly independent rows, $p = n$, and since this applies element-wise, $\text{rank}(H) = n$ implies (A.4).

Alternatively, (A.4) implies a relationship of the form (A.5) exists, and hence $\text{rank}(H) = p$. Since n is the smallest possible p, this implies $\text{rank}(H) = n$. □

We now prove Theorem A.1.

Proof. Suppose $g(z)$ is a coprime rational function of the form (A.3) with series expansion (A.1), which we know exists, since $g(z)$ is analytic for $|z| < 1$. Without loss of generality, let $m = n - 1$, since we may always let $b_k = 0$ for any k. Hence

$$\frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1 z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0} = g_1 z^{-1} + g_2 z^{-2} + g_3 z^{-3} + \cdots$$

Multiplying both sides by the denominator,

$$\begin{aligned} b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1 z + b_0 \\ = g_1 z^{n-1} + (g_2 + g_1 a_{n-1})z^{n-2} + (g_3 + g_2 a_{n-1} + g_1 a_{n-2})z^{n-3} + \cdots, \end{aligned}$$

and equating powers of z, we find

$$\begin{aligned} b_{n-1} &= g_1 \\ b_{n-2} &= g_2 + g_1 a_{n-1} \\ b_{n-3} &= g_3 + g_2 a_{n-1} + g_1 a_{n-2} \\ &\vdots \\ b_1 &= g_{n-1} + g_{n-2} a_{n-1} + \cdots + g_1 a_2 \\ b_0 &= g_n + g_{n-1} a_{n-1} + \cdots + g_1 a_1 \\ 0 &= g_{k+1} + g_k a_{n-1} + \cdots + g_{k-n+1} a_0 \qquad k \geq n. \end{aligned} \tag{A.6}$$

From this, we have, for $k \geq n$,

$$g_{k+1} = \sum_{j=1}^{n} -a_j g_{k-j+1},$$

which not only shows that (A.4) holds, but even provides $\alpha_j = -a_j$. Hence by Theorem A.2, H must have finite rank.

Conversely, suppose H has finite rank. Then (A.4) holds, and we may construct $a(z)$ from α_k and $b(z)$ from (A.6) to create a rational function. This function must be coprime since its order n is the smallest possible. □

Matrices with the structure of H are useful enough to have a special name. A *Hankel matrix* is a matrix H constructed from a sequence $\{h_k\}$ so that each element $H_{(j,k)} = h_{j+k}$. For the Hankel matrix in (A.2), $h_k = g_{k-1}$. Of course at the time the rank of a matrix was tested analytically through determinants, and Kronecker provided some formulas for calculating the cofactors of H which are unnecessary with modern computers.

B Linear Algebra and Matrix Identities

This appendix provides some background on orthogonal projectors, some results on the asymptotic behavior of data-matrix products, and some results on the differentials of Frobenius-norm cost functions.

B.1 Projectors and Orthogonal Subspaces

It is well known that for some subspace $\mathcal{S}$, a vector v can be decomposed into two components: one part in $\mathcal{S}$ and one part in the orthogonal complement $\mathcal{S}^\perp$,

$$v = v_\mathcal{S} + v_{\mathcal{S}^\perp}.$$

A *projector matrix* Π projects a vector onto a subspace. All projector matrices satisfy the property

$$\Pi = \Pi^2. \tag{B.1}$$

A number of interesting properties can be derived from (B.1). Π is clearly square; additionally, its eigenvalues must all be either 0 or 1.

An *orthogonal projector* projects a vector onto a subspace and nullifies the component of the vector in the orthogonal complement of the subspace. If $\Pi_\mathcal{S}$ is the orthogonal projector onto $\mathcal{S}$, then

$$\Pi_\mathcal{S} v = \Pi_\mathcal{S} v_\mathcal{S} + \cancelto{0}{\Pi_\mathcal{S} v_{\mathcal{S}^\perp}} = v_\mathcal{S}$$

It straightforward that the projector onto the orthogonal complement of $\mathcal{S}$ is

$$\Pi_{\mathcal{S}^\perp} = I - \Pi_\mathcal{S}.$$

All orthogonal projectors are symmetric, since

$$\begin{aligned}&(\Pi_{\mathcal{S}}v)^T(I-\Pi_{\mathcal{S}})v = v^T(\Pi_{\mathcal{S}}^T - \Pi_{\mathcal{S}}^T\Pi_{\mathcal{S}})v = 0\\ \Rightarrow&\Pi_{\mathcal{S}} = \Pi_{\mathcal{S}}^T\Pi_{\mathcal{S}} = \Pi_{\mathcal{S}}^2\\ \Rightarrow&\Pi_{\mathcal{S}} = \Pi_{\mathcal{S}}^T.\end{aligned}$$

$\Pi_{\mathcal{S}}$ then has singular values equal to its eigenvalues, and since its eigenvalues are 1 or 0,

$$||\Pi_{\mathcal{S}}||_2 = 1 \text{ or } 0.$$

This also proves that orthogonal projection cannot increase the 2-norm of a vector. (This is not true for *oblique projections*, however.)

B.2 Data-Matrix Identities

Asymptotic Behavior of Data-Matrix Products

We first derive some basic results for block-Hankel matrices of arbitrary signal data. Given signals $s(t) \in \mathbb{R}^{n_s}$ and $w(t) \in \mathbb{R}^{n_w}$. This implies the cross-covariance function $R_{sw}(\tau) \in \mathbb{R}^{n_s \times n_w}$ exists.

The following lemma is trivial to prove through cumbersome expansion but useful enough to warrant a reference.

Lemma B.1. *Given data matrices*

$$S_{i_0|i_1} = \begin{bmatrix} s(i_0) & s(i_0+1) & \cdots & s(i_0+l-1) \\ s(i_0+1) & s(i_0+2) & \cdots & s(i_0+l) \\ \vdots & \vdots & & \vdots \\ s(i_1) & s(i_1+1) & \cdots & s(i_1+l-1) \end{bmatrix} \in \mathbb{R}^{(i_1-i_0+1)n_s \times l}$$

and

$$W_{j_0|j_1} = \begin{bmatrix} w(j_0) & w(j_0+1) & \cdots & w(j_0+l-1) \\ w(j_0+1) & w(j_0+2) & \cdots & w(j_0+l) \\ \vdots & \vdots & & \vdots \\ w(j_1) & w(j_1+1) & \cdots & w(j_1+l-1) \end{bmatrix} \in \mathbb{R}^{(j_1-j_0+1)n_w \times l},$$

then

$$S_{i_0|i_1}W_{j_0|j_1}^T =$$
$$\begin{bmatrix} \sum_{k=i_0}^{i_0+l-1} s(k)w^T(k-i_0+j_0) & \cdots & \sum_{k=i_0}^{i_0+l-1} s(k)w^T(k-i_0+j_1) \\ \sum_{k=i_0+1}^{i_0+l} s(k)w^T(k-i_0+j_0-1) & \cdots & \sum_{k=i_0+1}^{i_0+l} s(k)w^T(k-i_0+j_1-1) \\ \vdots & & \vdots \\ \sum_{k=i_1}^{i_1+l-1} s(k)w^T(k-i_1+j_0) & \cdots & \sum_{k=i_1}^{i_1+l-1} s(k)w^T(k-i_1+j_1) \end{bmatrix}$$

Vectorized Data Matrix Products

The following useful identity is derived half-way in Viberg, Wahlberg, and Ottersten [80], but the form presented here appears to be original. This greatly simplifies the derivation of the results in Viberg, Wahlberg, and Ottersten [80] and Gustafsson [23].

Theorem B.1. *Given two sequences of vectors $s_i \in \mathbb{R}^n$ and $w_i \in \mathbb{R}^m$ in which n and m are not necessarily the same, suppose each sequence contains l vectors. Define the block matrices*

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & \cdots & s_{l-1} \end{bmatrix} \in \mathbb{R}^{n\times l}$$

and

$$W = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_{l-1} \end{bmatrix} \in \mathbb{R}^{m\times l}.$$

Then

$$\mathrm{vec}(SW^T)\mathrm{vec}(SW^T)^T = \sum_{j=0}^{l-1}\sum_{k=0}^{l-1} \left(w_j w_k^T\right) \otimes \left(s_j s_k^T\right)$$

Proof. First,

$$SW^T = s_0w_0^T + s_1w_1^T + s_2w_2^T + \cdots + s_{l-1}w_{l-1}^T = \sum_{k=0}^{l-1} s_k w_k^T \in \mathbb{R}^{n\times m}$$

leads to

$$\operatorname{vec}(SW^T) = \begin{bmatrix} \left(\sum_{k=0}^{l-1} s_k w_k^T\right)_{(:,\ 1)} \\ \left(\sum_{k=0}^{l-1} s_k w_k^T\right)_{(:,\ 2)} \\ \vdots \\ \left(\sum_{k=0}^{l-1} s_k w_k^T\right)_{(:,\ n_w r)} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{l-1} \left(s_k w_k^T\right)_{(:,\ 1)} \\ \sum_{k=0}^{l-1} \left(s_k w_k^T\right)_{(:,\ 2)} \\ \vdots \\ \sum_{k=0}^{l-1} \left(s_k w_k^T\right)_{(:,\ n_w r)} \end{bmatrix}$$

$$= \sum_{k=0}^{l-1} \begin{bmatrix} \left(s_k w_k^T\right)_{(:,\ 1)} \\ \left(s_k w_k^T\right)_{(:,\ 2)} \\ \vdots \\ \left(s_k w_k^T\right)_{(:,\ n_w r)} \end{bmatrix} = \sum_{k=0}^{l-1} \begin{bmatrix} s_k \left(w_k\right)_{(1)} \\ s_k \left(w_k\right)_{(2)} \\ \vdots \\ s_k \left(w_k\right)_{(n_w r)} \end{bmatrix},$$

where the subscripts represent MATLAB-style indexing (remembering that w_k is a vector). From this,

$$\operatorname{vec}(SW^T)\operatorname{vec}(SW^T)^T = \left(\sum_{j=0}^{l-1} \begin{bmatrix} s_j \left(w_j\right)_{(1)} \\ s_j \left(w_j\right)_{(2)} \\ \vdots \\ s_j \left(w_j\right)_{(n_w r)} \end{bmatrix}\right) \left(\sum_{k=0}^{l-1} \begin{bmatrix} s_k \left(w_k\right)_{(1)} \\ s_k \left(w_k\right)_{(2)} \\ \vdots \\ s_k \left(w_k\right)_{(n_w r)} \end{bmatrix}\right)^T$$

$$= \sum_{j=0}^{l-1} \sum_{k=0}^{l-1} \begin{bmatrix} s_j \left(w_j\right)_{(1)} \\ s_j \left(w_j\right)_{(2)} \\ \vdots \\ s_j \left(w_j\right)_{(n_w r)} \end{bmatrix} \begin{bmatrix} s_k \left(w_k\right)_{(1)} \\ s_k \left(w_k\right)_{(2)} \\ \vdots \\ s_k \left(w_k\right)_{(n_w r)} \end{bmatrix}^T$$

$$= \sum_{j=0}^{l-1} \sum_{k=0}^{l-1} \begin{bmatrix} s_j \left(w_j\right)_{(1)} \\ s_j \left(w_j\right)_{(2)} \\ \vdots \\ s_j \left(w_j\right)_{(n_w r)} \end{bmatrix} \begin{bmatrix} s_k^T \left(w_k\right)_{(1)} & s_k^T \left(w_k\right)_{(2)} & \cdots & s_k^T \left(w_k\right)_{(n_w r)} \end{bmatrix}$$

$$= \sum_{j=0}^{l-1}\sum_{k=0}^{l-1} \begin{bmatrix} s_j \left(w_j\right)_{(1)} s_k^T \left(w_k\right)_{(1)} & \cdots & s_j \left(w_j\right)_{(1)} s_k^T \left(w_k\right)_{(n_w r)} \\ s_j \left(w_j\right)_{(2)} s_k^T \left(w_k\right)_{(1)} & \cdots & s_j \left(w_j\right)_{(2)} s_k^T \left(w_k\right)_{(n_w r)} \\ \vdots & & \vdots \\ s_j \left(w_j\right)_{(n_w r)} s_k^T \left(w_k\right)_{(1)} & \cdots & s_j \left(w_j\right)_{(n_w r)} s_k^T \left(w_k\right)_{(n_w r)} \end{bmatrix}$$

$$= \sum_{j=0}^{l-1}\sum_{k=0}^{l-1} \begin{bmatrix} \left(w_j\right)_{(1)} \left(w_k\right)_{(1)} & \cdots & \left(w_j\right)_{(1)} \left(w_k\right)_{(n_w r)} \\ \left(w_j\right)_{(2)} \left(w_k\right)_{(1)} & \cdots & \left(w_j\right)_{(2)} \left(w_k\right)_{(n_w r)} \\ \vdots & & \vdots \\ \left(w_j\right)_{(n_w r)} \left(w_k\right)_{(1)} & \cdots & \left(w_j\right)_{(n_w r)} \left(w_k\right)_{(n_w r)} \end{bmatrix} \otimes \left(s_j s_k^T\right)$$

$$= \sum_{j=0}^{l-1}\sum_{k=0}^{l-1} \left(w_j w_k^T\right) \otimes \left(s_j s_k^T\right).$$

□

B.3 Gradients of Frobenius-Norm Cost Functions

In this section, we demonstrate that although the global minimum of a Frobenius-norm cost function does not change for left- and right- weightings, the local minima may change. Suppose we have a cost function for some matrix X,

$$f_0(X) = \frac{1}{2} ||AXB - C||_F^2 . \tag{B.2}$$

This function has the global minimum

$$X = A^\dagger C B^\dagger$$

where $(\cdot)^\dagger$ is the pseudoinverse (left or right depending on context, assuming it exists). The cost function

$$f_1(X) = \frac{1}{2} \left|\left|X - A^\dagger C B^\dagger\right|\right|_F^2 \tag{B.3}$$

clearly has the same global minimum. We would like to know if these functions also have the same minimum over a convex set. It turns out in the general case

that they do not. To see this, we know that

$$\begin{aligned} f_0(X) &= \frac{1}{2}\text{trace}\left[(AXB-C)^T(AXB-C)\right] \\ &= \frac{1}{2}\text{trace}\left(BB^TX^TA^TAX\right) - \text{trace}\left(BC^TAX\right) + \frac{1}{2}\text{trace}\left(C^TC\right), \end{aligned}$$

and

$$\begin{aligned} f_1(X) &= \frac{1}{2}\text{trace}\left[(X-A^\dagger CB^\dagger)^T(X-A^\dagger CB^\dagger)\right] \\ &= \frac{1}{2}\text{trace}\left(X^TX\right) - \text{trace}\left[(B^\dagger)^TC^T(A^\dagger)^TX\right] \\ &\qquad + \frac{1}{2}\text{trace}\left[(B^\dagger)^TC^T(A^\dagger)^TA^\dagger CB^\dagger\right]. \end{aligned}$$

Let $x = \text{vec}(X)$. Using the matrix differential,

$$\begin{aligned} df_0(X) &= \frac{1}{2}\text{trace}\left[BB^T(dX)^TA^TAX\right] + \frac{1}{2}\text{trace}\left[BB^TX^TA^TA(dX)\right] \\ &\qquad - \text{trace}\left[BC^TA(dX)\right] \\ &= \text{trace}\left[(BB^TX^TA^TA - BC^TA)(dX)\right] \\ &= \text{vec}\left(A^TAXBB^T - A^TCB^T\right)^T\text{vec}(dX) \\ &= \text{vec}\left[A^T(AXB-C)B^T\right]^T d\text{vec}(X) \\ &= \text{vec}\left[A^T(AXB-C)B^T\right]^T dx, \end{aligned}$$

hence

$$\frac{df_0(x)}{dx} = \text{vec}\left[A^T(AXB-C)B^T\right]^T,$$

and

$$\begin{aligned} df_1(X) &= \frac{1}{2}\text{trace}\left[(dX)^TX\right] + \frac{1}{2}\text{trace}\left[X^T(dX)\right] \\ &\qquad - \text{trace}\left[(B^\dagger)^TC^T(A^\dagger)^T(dX)\right] \\ &= \text{trace}\left[\left(X^T - (B^\dagger)^TC^T(A^\dagger)^T\right)(dX)\right] \\ &= \text{vec}\left[X - A^\dagger CB^\dagger\right]^T\text{vec}(dX) \\ &= \text{vec}\left[(A^TA)^{-1}A^T(AXB-C)B^T(BB^T)^{-1}\right]^T dx \\ &= \left[\left((A^TA)^{-1}\otimes(BB^T)^{-1}\right)\text{vec}\left(A^T(AXB-C)B^T\right)\right]^T dx \\ &= \text{vec}\left[A^T(AXB-C)B^T\right]^T\left[(A^TA)^{-1}\otimes(BB^T)^{-1}\right]dx, \end{aligned}$$

hence

$$\frac{df_1(x)}{dx} = \frac{df_0(x)}{dx}\left[(A^TA)^{-1} \otimes (BB^T)^{-1}\right].$$

Thus the valid descent directions at any point might not be equivalent, depending on the values of A and B.

C Software

Two MATLAB software packages were written to implement the algorithms contained in this dissertation. The COBRA package estimates system models using the realization-based procedure either directly from input-output data or by first calculating covariance function estimates. The Stepalize tool computes system estimates form step-response data.

C.1 COBRA Toolbox

The COBRA Toolbox is a lightweight framework of functions that can be used to generate system estimates using the realization-theoretic framework presented in this dissertation. The functions are included in MATLAB-package format, so that they are accessed via the syntax

```
cobra.[function name]
```

Help information is included with each function, and may be accessed via the syntax

```
help cobra.[function name]
```

A demo file `cobra_demo.m` is included with the software.

`X = blkhankel(C, R)`

Generate block-Hankel matrix with the first block-column `C` and the last block-row `R`.

`X = datahankel(D, m)`

Construct a block-Hankel matrix of `m` block rows from the $N \times p$ signal `D`, where is N is the number of samples in the signal and p is the dimension of the signal.

`X = datahankel(D, m, r, c)`

Construct a block-Hankel matrix of `m` block rows from the $N \times \texttt{r} \cdot \texttt{c}$ signal `D`, where N is the number of samples in the signal, `r` is the row dimension of the signal, and `c` is the column dimension of the signal. The first `c` columns of the i^{th} row of `D` will be the first row of the i^{th} block of Y.

If `size(D, 3) > 1`, then `D` is treated as an $\texttt{r} \times \texttt{c} \times N$ signal.

`[A, B, C] = hokalman(H, n, ny, nu)`

Given a Hankel matrix `H` consisting of system Markov parameters (impulse-response coefficients), return the system matrices `A`, `B`, and `C`. `n` is the order of the system, `ny` is the dimension of the output, and `nu` is the dimension of the input. `H` must have at least one block row greater than the system dimension. Use "fat" matrices with more columns than rows to get better results. The singular-value decomposition (SVD) is used to determine the state-basis.

`[L, Q] = lq(A)`

Given an $n \times m$ matrix `A`, returns an $n \times n$ lower triangular matrix `L` and an $m \times m$ unitary matrix `Q` so that $\texttt{A} = \texttt{LQ}$.

`M = nullprojection(Y, X)`

Project `Y` onto the null-space of the row-space of `X` and return the result as `M`. This is equivalent to

$$M = Y(I - X^T(XX^T)^{-1}X),$$

though it is computed more efficiently.

`M = nullprojection(Y, X, fY, fX)`

Obliquely project the bottom `fY` rows of `Y` onto the null-space of the bottom `fX` rows of `X` along the row space of the top rows of `X` and `Y`. That is, given

$$Y = \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} \qquad X = \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}$$

where Y_1 has `fY` rows and X_1 has `fX` rows, then `M` is a linear combination of Y_0 and X_0 and in the null-space of X_1.

If `fX` = `fY` then this is the "future horizon" of N4SID.

`n = selectorder(s)`

Plot the entries of `s` with a logarithmic y-axis. Each entry of `s` corresponds to a circle on the x–y plane. When the user clicks on a circle, the circle is highlighted. If the user then clicks "OK," the index of the entry clicked is returned in `n`. If the user exits without selecting an order, then `n` = -1.

`[A, C] = solveAC(YOPi, Y1Pi, n, ny, args)`

Estimate `A` and `C` using shift-invariance of the output data in `YOPi` and `Y1Pi`. `YOPi` is a matrix of output data projected onto some matrix Π, and `Y1Pi` is a matrix of time-shifted output data projected onto the *same* Π. `n` is the desired system order.

The function is general enough to be used with matrices generated from either raw data or covariance function estimates.

`args` is an optional sequence of arguments in ('Property','Value') pairs to be used when eigenvalue constraints are desired. The following properties are valid:

`'Eigenvalues'` — Constraints to place on eigenvalues. This has the following possible values:

- `'Stable'` — Force all eigenvalues to satisfy $|z| < 1 - \delta_s$ where δ_s may be set by the `'DeltaS'` property. By default, $\delta_s = 1 \times 10^{-4}$.

- `'Real'` — Force all eigenvalues to satisfy $|\mathrm{Im}(z)| < \delta_r$, where δ_r may be set by the `'DeltaR'` property. By default, $\delta_r = 1 \times 10^{-8}$.
- `'Positive'` — Force all eigenvalues to satisfy $\mathrm{Re}(z) > \delta_p$, where δ_p may be set by the `'DeltaP'` property. By defaults, $\delta_p = 1 \times 10^{-4}$.

`'DeltaS'` — δ_s to use for stability constraint

`'DeltaR'` — δ_r to use for strictly real constraint

`'DeltaP'` — δ_p to use for positive real constraint

`'Verbose'` — verbosity argument passed to YALMIP

Note that YALMIP and a suitable solver are required for eigenvalue constraints.

`[B, D, x0] = solveBDx0(y, u, A, C, nk)`

Determine a least-squares estimate of the input matrices `B` and `D` and an initial condition `x0` given matrices `A` and `C`, input `u`, and output `y` of an LTI system. `u` and `y` should be $N \times n_u$ and $N \times n_y$, respectively.

`nk` is the number of time delays. This argument is optional. By default it is 1, so that $D = 0$ if the argument is omitted. If `nk` > 1, it assumed that `A` does not include the time delays already. Instead, `u` will be shifted by `nk` $- 1$ places prior to estimating `B` and `D`.

`[B, D, x0] = solveBDx0(y, u, A, C, nk, 'raw')`

Same as above.

`[B, D, Rxz0] = solveBDx0(Ryz, Ruz, A, C, nk, 'cov')`

Determine a least-squares estimate of the input matrices `B` and `D` and the initial value of the covariance function `Rxz(0)` given matrices `A` and `C`, input cross-covariance function `Ruz` (`z` is some external instrument, possibly `u`), and cross-covariance function `Ryz`.

If `size(Ryz, 3)` = 1, then we assume the covariance functions are in the form returned by the xcov function of the Signal Processing Toolbox, and it is assumed that dim(`z`) = dim(`u`). If `size(Ryz, 3)` > 1, then they have the form returned by the `freqresp` function of the Control Systems Toolbox.

```
[B, D, x0] = solveBDx0(y, u, A, C, nk, 'raw', 0)
```

or

```
[B, D, Rxz0] = solveBDx0(Ryz, Ruz, A, C, nk, 'cov', 0)
```

Same as above, but force a zero initial condition. By default, the initial condition is always estimated.

The following two functions essentially replace the `xcov` function of the Signal Processing Toolbox with a version that includes frequency-domain smoothing.

```
R = xcovsmooth(x)
```

Construct a smoothed auto-covariance function estimate of `x`. By default, `x` is split into 8 segments with 50% overlap, and a Hamming window is applied to each segment. (This is the default behavior of the `cpsd` function of the signal processing toolbox.)

```
R = xcovsmooth(x, y)
```

Construct a smoothed cross-covariance function estimate of x and y. The default smoothing behavior is the same as the auto-covariance.

```
R = xcovsmooth(x, y, window)
```

If `window` is a vector, `x` and `y` are divided into overlapping sections of length equal to the length of `window`, and each section is weighted by the vector `window`.

If `window` is an integer, `x` and `y` are divided into that number of segments and a Hamming window of equal length is used.

If `window` is an empty matrix, the default is used.

`window = ones(length(x), 1)` will return an un-smoothed estimate, reproducing the functionality of `xcov`.

For large datasets, the number of windows can have a dramatic effect on the speed at which the estimates are computed. Use `window = N` to apply no window to the estimate.

`R = xcovsmooth(x, y, window, overlap)`

Uses `overlap` samples from section to section. If empty or not specified, 50% overlap is used.

`R = xcovsmooth(x, y, window, overlap, lags)`

Only returns lags from `lags(1)` to `lags(2)`. `lags(1)` should likely be less than 0.

`[R, tau] = xcovsmooth(...)`

A vector tau of time lags is returned in addition to `R`.

For all options, if `x` and `y` are vectors, `R` is also a vector. If either `x` or `y` are matrices, then the cross-covariances of each column of `x` and each column of `y` are computed, and `R` has dimensions $\texttt{size}(\texttt{x}, 2) \times \texttt{size}(\texttt{y}, 2) \times \texttt{size}(\texttt{x}, 1)$. An error occurs if `x` and `y` do not have the same number of rows. Also for all options, the covariance function is scaled in the same manner as the `'unbiased'` option of the `xcov` function.

`xcovsmooth` relies heavily on the function `cpsd` of the Signal Processing Toolbox. See its documentation for more details.

`[Ryz, Ruz, tau] = xcorrsignals(y, u, z, window, overlap, lags)`

Given signals `y`, `u`, and `z`, compute cross-covariance function estimates of `Ryz` and `Ruz`. `y`, `u`, and `z` should have each column contain a different signal (in the case of multivariable signals) and each row contain a time sample.

This function presents a simple way to avoid cluttering the code with calls to `xcovsmooth`.

The remaining arguments are passed to `xcovsmooth`; see the help for that function for their significance.

C.2 Stepalize

The Stepalize software tool is a single MATLAB function `stepalize.m` that generates discrete-time LTI models from measured step responses. If YALMIP and SDPT3 are installed, it can also generate models with constrained eigenvalues and constrained time-domain behavior. The package includes a file `stepalize_test.m` that performs a system check and verifies if the system is capable of solving semidefinite programs.

Usage:

```
[A, B, C, D] = stepalize(y, args)
```

in which 'args' is any of the (optional) following name-value pairs:

`Order` — Order of the model. If not specified, the user will be presented with a singular-value plot on which they may select the system order.

`TimeDelay` — By default, 1 time delay is assumed, and D is always 0. Set this to 0 to estimate a nonzero D. If this is greater than 1, then the data is shifted so that there is 1 time delay, and D is again estimated as 0.

`Eigenvalues` — Constraints placed on the eigenvalues. For each of the following options, all eigenvalues λ will satisfy the corresponding inequality:
`Stable`: $|\lambda| < 1 -$ `DeltaS`
`Real`: $|\mathrm{Im}(\lambda) <$ `DeltaR`

`Positive`: $\mathrm{Re}(\lambda) >$ `DeltaP`

For only one constraint, a single string works, i.e.
`...'Eigenvalues', 'Stable', ...`

For multiple choices, use a cell array, i.e.
`...'Eigenvalues', {'Real', 'Stable'}, ...`

`Delta*` — δ-values for the eigenvalue constraints, where $* =$ S, R, or P. Default values are `DeltaS` $= 1 \times 10^{-4}$, `DeltaR` $= 1 \times 10^{-8}$, and `DeltaP` $= 1 \times 10^{-4}$.

`SteadyState` — Guarantee the model has a fixed stead-state value. The value for this property should be a $1 \times n_y$ vector which will be settling value for the model's step response.

`NoOvershoot` — Guarantee the model does not overshoot the steady-state value of the data. WARNING: This is often infeasible, depending on the eigenvalues. If a step-response with no overshoot is absolutely required, generally the eigenvalues should be constrained to be real.

`NoUndershoot` — Restrict resulting model to have no undershoot in its step response. Undershoot is defined as when the step response has the opposite sign of the steady-state value for a given output channel. This allows for the step response to still have values < 0, but only when the steady-state value at that channel is < 0 as well. This generally eliminates non-minimum-phase behavior from the model.

`UseAC` — Use pre-calculated matrices A and C to estimate B and D only. The value for this must be a 2-element cell array with the first element equal to A and the second equal to C. Constraint options for calculating B and D are still available.

Any options affecting the computation of A and C are ignored.

`WeightBD` — Weight the solution for B and D. This should be an $N \times 1$ vector for which element i corresponds to the weight of the solution for sample i of a step of length N. This is not available if constraints are used with B and D. In this case, the weights will be ignored and a warning will be generated.

WARNING: Combinations of `SteadyState`, `NoOvershoot`, and `NoUndershoot` are prone to over-constraining the model and will often result in an infeasibility error or the solution $B = 0$. Only use them in combination when the estimate is close to satisfying the constraints already. Feasibility issues are also reduced when the system is parameterized with a feed-through (D-matrix) term.

Example:

```
% Generate an n-th order model with a feed-through term and real
% eigenvalues.
[A, B, C, D] = stepalize(y, 'Order', n, ...
                         'TimeDelay', 0, ...
                         'Eigenvalues', 'Real');
```

Bibliography

[1] J.E. Ackermann and R.S. Bucy. "Canonical Minimal Realization of a Matrix of Impulse Response Sequences". In: *Information and Control* 19.3 (1971), pp. 224–231. DOI: 10.1016/S0019-9958(71)90105-7.

[2] Salim Ahmed, Biao Huang, and Sirish L. Shah. "Novel identification method from step response". In: *Control Engineering Practice* 15.5 (May 2007), pp. 545–556. ISSN: 09670661. DOI: 10.1016/j.conengprac.2006.10.005.

[3] A. Alenany, H. Shang, M. Soliman, and I. Ziedan. "Improved subspace identification with prior information using constrained least squares". In: *IET Control Theory & Applications* 5.13 (2011), pp. 1568–1576. ISSN: 17518644. DOI: 10.1049/iet-cta.2010.0585.

[4] Karl J. Aström and Tore Hägglund. *PID Controllers: Theory, Design, and Tuning*. 2nd. ISA, 1995. ISBN: 978-1556175169.

[5] Dario H. Baldelli, Jie Zeng, Richard C. Lind, and Chuck Harris. "Flutter-Prediction Tool for Flight-Test-Based Aeroelastic Parameter-Varying Models". In: *Journal of Guidance, Control, and Dynamics* 32.1 (Jan. 2009), pp. 158–171. ISSN: 0731-5090. DOI: 10.2514/1.36584.

[6] Tamara Bechtold, Evgenii B. Rudnyi, and Jan G. Korvink. "Automatic Generation of Compact Electro-Thermal Models for Semiconductor Devices". In: *IEICE Transactions on Electronics* E86-C.3 (2003), pp. 459–465. DOI: 10.1.1.58.9555.

[7] Qiang Bi, Wen-Jian Cai, Eng-Lock Lee, Qing-Guo Wang, Chang-Chieh Hang, and Yong Zhang. "Robust identification of first-order plus dead-time model from step response". In: *Control Engineering Practice* 7.1 (Jan. 1999), pp. 1–7. ISSN: 09670661. DOI: 10.1016/S0967-0661(98)00166-X.

[8] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN: 978-0521833783.

[9] Martin J. Brenner, Richard C. Lind, and David F. Voracek. *Overview of Recent Flight Flutter Testing Research at NASA Dryden*. Tech. rep. Edwards, California: NASA Dryden Flight Research Center, 1997.

[10] Chi-Tsong Chen. *Linear System Theory and Design*. 1st. New York: Oxford University Press, 1984. ISBN: 0195115953.

[11] Chi-Tsong Chen and Dinesh P. Mital. "A Simplified Irreducible Realization Algorithm". In: *IEEE Transactions on Automatic Control* 17.4 (Aug. 1972), pp. 535–537. ISSN: 0018-9286. DOI: 10.1109/TAC.1972.1100078.

[12] Mahmoud Chilali and Pascal Gahinet. "H-Infinity design with pole placement constraints: an LMI approach". In: *IEEE Transactions on Automatic Control* 41.3 (Mar. 1996), pp. 358–367. ISSN: 00189286. DOI: 10.1109/9.486637.

[13] Alessandro Chiuso. "The role of vector autoregressive modeling in predictor-based subspace identification". In: *Automatica* 43.6 (June 2007), pp. 1034–1048. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2006.12.009.

[14] Robert Clark, David Cox, H.C. Curtiss, John W. Edwards, Kenneth C. Hall, David A. Peters, Robert Scanlan, Emil Simiu, Fernando Sisto, and Thomas W. Strganac. *A Modern Course in Aeroelasticity*. Ed. by Earl H. Dowell. 4th ed. Kluwear Academic Publishers, 2005. ISBN: 1402021062.

[15] Raymond A. de Callafon, B. Moaveni, J. P. Conte, X. He, and E. Udd. "General Realization Algorithm for Modal Identification of Linear Dynamic Systems". In: *Journal of Engineering Mechanics* 134.9 (2008), pp. 712–722. DOI: 10.1061/(ASCE)0733-9399(2008)134:9(712).

[16] Harold M. Edwards. "An Appreciation of Kronecker". In: *The Mathematical Intelligencer* 9.1 (1987), pp. 28–35. DOI: 10.1007/BF03023570.

[17] Harold M. Edwards. *Essays in Constructive Mathematics*. New York, NY: Springer New York, 2005. ISBN: 978-0-387-21978-3. DOI: 10.1007/b138656.

[18] Harold M. Edwards. "Kronecker's Algorithmic Mathematics". In: *The Mathematical Intelligencer* 31.2 (2008), pp. 11–14.

[19] Felix R. Gantmacher. *The Theory of Matrices - Volume One*. New York: Chelsea Publishing Company, 1960.

[20] Felix R. Gantmacher. *The Theory of Matrices - Volume Two*. New York: Chelsea Publishing Company, 1960. ISBN: 978-0821826645.

[21] Elmer G. Gilbert. "Controllability and Observability in Multivariable Control Systems". In: *Journal of the Society for Industrial and Applied Mathematics, Series A: Control* 2.1 (1963), pp. 128–151. DOI: 10.1137/0301009.

[22] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Third. Baltimore, Maryland, USA: The Johns Hopkins University Press, 1996. ISBN: 0-8018-5413-8.

[23] Tony Gustafsson. "Subspace-based system identification: weighting and prefiltering of instruments". In: *Automatica* 38.3 (Mar. 2002), pp. 433–443. ISSN: 00051098. DOI: 10.1016/S0005-1098(01)00235-7.

[24] I. Gustavsson. "Survey of Applications of Identification in Chemical and Physical Processes". In: *Automatica* 11 (1975), pp. 3–24. DOI: 10.1016/0005-1098(75)90005-9.

[25] Jesse B. Hoagg, Seth L. Lacy, R. Scott Erwin, and Dennis S. Bemstein. "First-Order-Hold Sampling of Positive Real Systems And Subspace Identification of Positive Real Models". In: *Proceedings of the 2004 American Control Conference*. Boston, Massachusetts: AACC, 2004, pp. 861–866. ISBN: 0780383354.

[26] Jesse B. Hoagg, Seth L. Lacy, R. Scott Erwin, and Dennis S. Bernstein. "Subspace Identification with Lower Bounded Modal Frequencies". In: *Proceedings of the 2004 American Control Conference*. Boston, Massachusetts: AACC, 2004, pp. 867–872. ISBN: 0780383354.

[27] B. L. Ho and Rudolf E. Kalman. "Effective construction of linear state-variable models from input/output functions". In: *Regelungstechnik* 14 (1966), pp. 545–548.

[28] Shyh-Hong Hwang and Shih-Tsung Lai. "Use of two-stage least-squares algorithms for identification of continuous systems with time delay based on pulse responses". In: *Automatica* 40.9 (Sept. 2004), pp. 1561–1568. ISSN: 00051098. DOI: 10.1016/j.automatica.2004.03.017.

[29] Jer-Nan Juang. *Applied System Identification*. Prentice Hall, 1993. ISBN: 978-0130792112.

[30] Jer-Nan Juang and R. S. Pappa. "An Eigensystem Realization Algorithm (ERA) for Modal Parameter Identification and Model Reduction". In: *JPL Proc. of the Workshop on Identification and Control of Flexible Space Structures* 3 (Apr. 1985), pp. 299–318.

[31] Rudolf E. Kalman. "A New Approach to Linear Filtering and Prediction

Problems". In: *Journal of Basic Engineering* 82 (1960), pp. 35–45.

[32] Rudolf E. Kalman. "Canonical Structure of Linear Dynamical Systems". In: *Proceedings of the National Academy of Sciences* 48 (1962), pp. 596–600.

[33] Rudolf E. Kalman. "Contributions to the Theory of Optimal Control". In: *Boletin de la Sociedad Matematica Mexicana* 5 (1960), pp. 102–119.

[34] Rudolf E. Kalman. "Irreducible Realizations and the Degree of a Rational Matrix". In: *Journal of the Society for Industrial and Applied Mathematics* 13.2 (1965), pp. 520–544. DOI: 10.1137/0113034.

[35] Rudolf E. Kalman. "Mathematical Description of Linear Dynamical Systems". en. In: *Journal of the Society for Industrial and Applied Mathematics, Series A: Control* 1.2 (July 1963), p. 152. ISSN: 08874603. DOI: 10.1137/0301010.

[36] Rudolf E. Kalman. "On the General Theory of Control Systems". In: *Proceedings of the First International Congress of Automatic Control*. Moscow: IRE, 1960.

[37] Tohru Katayama. *Subspace Methods for System Identification*. Communications and Control Engineering. London: Springer-Verlag, 2005. ISBN: 1-85233-981-0.

[38] Sun-Yuan Kung. "A new identification and model reduction algorithm via singular value decomposition". In: *Proceedings of the 12th Asilomar Conference on Circuits, Systems, and Computers*. IEEE, 1978, pp. 705–714.

[39] Seth L. Lacy and Dennis S. Bernstein. "Subspace identification with guaranteed stability using constrained optimization". In: *IEEE Transactions on Automatic Control* 48.7 (July 2003), pp. 1259–1263. ISSN: 0018-9286. DOI: 10.1109/TAC.2003.814273.

[40] Lennart Ljung. *System Identification: Theory for the User*. 2nd. PTR Prentice Hall Information and System Sciences. Upper Saddle River, NJ: Prentice Hall PTR, 1999. ISBN: 0136566952.

[41] Johan Löfberg. "YALMIP : A toolbox for modeling and optimization in MATLAB". in: *Proceedings of the 2004 IEEE International Symposium on Computer Aided Control System Design*. Taipei, Taiwan: IEEE, 2004, pp. 284–289. ISBN: 0780386361.

[42] Jan M. Maciejowski. "Guaranteed stability with subspace methods". In: *Systems & Control Letters* 26.2 (Sept. 1995), pp. 153–156. ISSN: 01676911.

DOI: 10.1016/0167-6911(95)00010-7.

[43] Tucker McElroy. *A to Z of Mathematicians*. VB Techbooks, 2005. ISBN: 0816053383.

[44] Tomas McKelvey, Hüseyin Akçay, and Lennart Ljung. "Subspace-based multivariable system identification from frequency response data". In: *IEEE Transactions on Automatic Control* 41.7 (1996), pp. 960–979. ISSN: 0018-9286. DOI: 10.1109/9.508900.

[45] Tomas McKelvey and S.O. Reza Moheimani. "Estimation of Phase Constrained MIMO Transfer Functions with Applications to Flexible Structures with Mixed Collocated and Non-Collocated Actuators and Sensors". In: *Proceedings of 16th IFAC World Congress*. 4. Prague, Czech Repbulic: Elsevier, 2005.

[46] Jack McLaughlin. "Book Review: The Theory of Matrices and Applications of the Theory of Matrics". In: *Bulletin of the American Mathematical Society* 67.1 (1961), pp. 95–96.

[47] Brockway McMillan. "Introdution to Formal Realizability Theory - I". in: *The Bell System Technical Journal* 31.2 (1952), pp. 217–279.

[48] R.K. Mehra, S. Mahmood, and R. Waissman. "Identification of Aircraft and Rotorcraft Aeroelastic Modes Using State Space System Identification". In: *Proceedings of the 4th IEEE Conference on Control Applications*. IEEE, 1995, pp. 432–437. DOI: 10.1109/CCA.1995.555742.

[49] Laurent Mevel, Albert Benveniste, Michèle Basseville, Maurice Goursat, Bart Peeters, Herman Van der Auweraer, and Antonio Vecchio. "Input/output versus output-only data processing for structural identification - Application to in-flight data analysis". In: *Journal of Sound and Vibration* 295.3-5 (Aug. 2006), pp. 531–552. ISSN: 0022460X. DOI: 10.1016/j.jsv.2006.01.039.

[50] Laurent Mevel, Maurice Goursat, Albert Benveniste, and Michèle Basseville. "Aircraft Flutter Test Design Using Identification and Simulation: a SCILAB Toolbox". In: *Proceedings of 2005 IEEE Conference on Control Applications*. Toronto, Canada: IEEE, 2005, pp. 1115–1120. ISBN: 0-7803-9354-6. DOI: 10.1109/CCA.2005.1507280.

[51] Daniel N. Miller and Raymond A. de Callafon. "Identification of Linear Time-Invariant Systems via Constrained Step-Based Realization". In: *Proceedings of the 16th IFAC Symposium on System Identification*. Brussels, Belgium: IFAC, 2012, to appear.

[52] Daniel N. Miller and Raymond A. de Callafon. "Subspace Identification From Classical Realization Methods". In: *Proceedings of the 15th IFAC Symposium on System Identification*. Saint-Malo, France, July 2009.

[53] Daniel N. Miller and Raymond A. de Callafon. "Subspace identification using dynamic invariance in shifted time-domain data". In: *49th IEEE Conference on Decision and Control (CDC)*. Atlanta, GA, USA: IEEE, Dec. 2010, pp. 2035–2040. ISBN: 978-1-4244-7745-6. DOI: 10.1109/CDC.2010.5717151.

[54] Daniel N. Miller and Raymond A. de Callafon. "Subspace Identification with Eigenvalue Constraints". In: *submitted to Automatica* (2012).

[55] M. Okada and T. Sugie. "Subspace System Identification considering both Noise Attenuation and Use of Prior Knowledge". In: *Proceedings of the 35th IEEE Conference on Decision and Control*. December. Kobe, Japan: IEEE, 1996, pp. 3662–3667. ISBN: 0780335902. DOI: 10.1109/CDC.1996.577200.

[56] Jordi Palacín, Marc Salleras, M Puig, Josep Samitier, and Santiago Marco. "Evolutionary algorithms for compact thermal modelling of microsystems: application to a micro-pyrotechnic actuator". In: *Journal of Micromechanics and Microengineering* 14.7 (July 2004), pp. 1074–1082. ISSN: 0960-1317. DOI: 10.1088/0960-1317/14/7/030.

[57] Jordi Palacín, Marc Salleras, Josep Samitier, and Santiago Marco. "Dynamic Compact Thermal Models With Multiple Power Sources: Application to an Ultrathin Chip Stacking Technology". In: *IEEE Transactions on Advanced Packaging* 28.4 (2005), pp. 694–703. DOI: 10.1109/TADVP.2005.850507.

[58] Rik Pintelon and Johan Schoukens. *System Identification: A Frequency Domain Approach*. New York: Wiley-IEEE Press, Jan. 2001. ISBN: 0780360001.

[59] H. Rake. "Step Response and Frequency Response Methods". In: *Automatica* 16 (1980), pp. 519–526.

[60] Márta Rencz, András Poppe, Ernö Kollár, Sándor Ress, and Vladimir Székely. "Increasing the Accuracy of Structure Function Based Thermal Material Parameter Measurements". In: *IEEE Transactions on Components and Packaging Technologies* 28.1 (2005), pp. 51–57. DOI: 10.1109/TCAPT.2004.843204.

[61] J. Rissanen. "Recursive Identification of Linear Systems". In: *SIAM Journal on Control* 9.3 (1971), pp. 420–430. ISSN: 00361402. DOI: 10.1137/0309031.

[62] Setsuo Sagara and Zhen-Yu Zhao. "Recursive identification of transfer function matrix in continuous systems via linear integral filter". In: *International Journal of Control* 50.2 (Aug. 1989), pp. 457–477. ISSN: 0020-7179. DOI: 10.1080/00207178908953377.

[63] Leonard M. Silverman. "Realization of linear dynamical systems". In: *IEEE Transactions on Automatic Control* 16.6 (Dec. 1971), pp. 554–567. ISSN: 0018-9286. DOI: 10.1109/TAC.1971.1099821.

[64] Leonard M. Silverman. "Synthesis of Impulse Response Matrices by Internally Stable and Passive Realizations". In: *IEEE Transactions on Circuit Theory* 15.3 (1968), pp. 238–244. ISSN: 0018-9324. DOI: 10.1109/TCT.1968.1082818.

[65] Gilbert Strang. *Linear Algebra and Its Applications*. Fourth. Thomson Brooks/Cole, 2006. ISBN: 0030105676.

[66] Vladimir Székely. "A new evaluation method of thermal transient measurement results". In: *Microelectronics Journal* 28 (1997), pp. 277–292.

[67] Vladimir Székely. "Identification of RC Networks by Deconvolution: Chances and Limits". In: *IEEE Transactions on Circuits and Systems* 45.3 (1998), pp. 244–258.

[68] Vladimir Székely. "On the Representation of Infinite-Length Distributed RC One-Ports". In: *IEEE Transactions on Circuits and Systems* 38.7 (July 1991), pp. 711–719. ISSN: 00984094. DOI: 10.1109/31.135743.

[69] Vladimir Székely and Márta Rencz. "Thermal Dynamics and the Time Constant Domain". In: *IEEE Transactions on Components and Packaging Technologies* 23.3 (2000), pp. 587–594.

[70] Vladimir Székely and Tran Van Bien. "Fine Structure of Heat Flow Path in Semiconductor Devices: A Measurement and Identification Method". In: *Solid-State Electronics* 31.9 (1988), pp. 1363–1368.

[71] Kim-Chuan Toh, Michael J. Todd, and Reha H. Tutuncu. "SDPT3 - A Matlab software package for semidefinite programming, Version 1.3". In: *Optimization Methods and Software* 11&12 (1999), pp. 545–581.

[72] Pavel Trnka and Vladimír Havlena. "Subspace like identification incorporating prior information". In: *Automatica* 45.4 (Apr. 2009), pp. 1086–1091. ISSN: 00051098. DOI: 10.1016/j.automatica.2008.12.005.

[73] T. Van Gestel, J.A.K. Suykens, P. Van Dooren, and Bart De Moor. "Identifi-

cation of Stable Models in Subspace Identification by Using Regularization". In: *IEEE Transactions on Automatic Control* 46.9 (2001), pp. 1416–1420. DOI: 10.1109/9.948469.

[74] J.B. van Helmont, A.J.J. van der Weiden, and H. Anneveld. "Design of Optimal Controller for a Coal Fired Benson Boiler Based on a Modified Approximate Realisation Algorithm". In: *Applications of Multivariable System Techniques*. Ed. by R. Whalley. Elsevier Applied Science, 1990, pp. 313–320.

[75] Peter Van Overschee and Bart De Moor. "A Unifying Theorem for Three Subspace System Identification Algorithms". In: *Automatica* 31.12 (Dec. 1995), pp. 1853–1864. DOI: 10.1016/0005-1098(95)00072-0.

[76] Peter Van Overschee and Bart De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. London: Kluwer Academic Publishers, 1996.

[77] P. Verboven, B. Cauberghe, P. Guillaume, S. Vanlanduit, and E. Parloo. "Modal parameter estimation and monitoring for on-line flight flutter analysis". In: *Mechanical Systems and Signal Processing* 18.3 (2004), p. 24. DOI: 10.1016/S0888-3270(03)00074-8.

[78] Michel Verhaegen and Vincent Verdult. *Filtering and System Identification: A Least Squares Approach*. 1st ed. New York: Cambridge University Press, May 2007. ISBN: 0521875129.

[79] Mats Viberg. "Subspace-based Methods for the Identification Linear Time-invariant Systems". In: *Automatica* 31.12 (1995), pp. 1835–1851. DOI: 10.1016/0005-1098(95)00107-5.

[80] Mats Viberg, Bo Wahlberg, and Björn Ottersten. "Analysis of state space system identification methods based on instrumental variables and subspace fitting". In: *Automatica* 33.9 (Sept. 1997), pp. 1603–1616. DOI: 10.1016/S0005-1098(97)00097-6.

[81] A.H. Whitfield and N. Messali. "Integral-equation approach to system identification". In: *International Journal of Control* 45.4 (1987), pp. 1431–1445. DOI: 10.1080/00207178708933819.

[82] J.C. Willems, P. Rapisarda, Ivan Markovsky, and Bart De Moor. "A note on persistency of excitation". In: *Systems & Control Letters* 54.4 (Apr. 2005), pp. 325–329. ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2004.09.003.

[83] Jan R. Wright and Jonathan E. Cooper. *Introduction to Aircraft Aeroelasticity and Loads*. Wiley, 2007. ISBN: 9780470858400.

[84] Dante C. Youla and Plinio Tissi. “n-Port Synthesis via Reactance Extraction - Part I”. in: *IEEE International Convention Record* 14.7 (1966), pp. 183–208.

[85] Paul H. Zeiger and Julia A. McEwen. “Approximate Linear Realizations of Given Dimension via Ho’s Algorithm”. In: *IEEE Transactions on Automatic Control* 19.2 (1973), pp. 153 –153. DOI: 10.1109/TAC.1974.1100525.

[86] Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice Hall, Aug. 1995. ISBN: 0134565673.