Title
Identification of State-Space Models for High-Order Linear Systems and Optical Wavefronts

Permalink
https://escholarship.org/uc/item/40n2d5dh

Author
Faghihi, Azin

Publication Date
2014

Peer reviewed|Thesis/dissertation
Identification of State-Space Models
for High-Order Linear Systems
and Optical Wavefronts

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

by

Azin Faghihi

2014
Abstract of the Dissertation

Identification of State-Space Models for High-Order Linear Systems and Optical Wavefronts

by

Azin Faghihi

Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2014
Professor Steve Gibson, Chair

A state-space disturbance model and associated prediction filter for aero-optical wavefronts are presented. The model is computed by system identification from a sequence of wavefronts measured in an airborne laboratory. Estimates of the statistics and flow velocity of the wavefront data are shown and can be computed from the matrices in the state-space model without returning to the original data. Numerical results compare velocity values and power spectra computed from the identified state-space model with those computed from the aero-optical data.

A lattice-filter based state-space model is developed for multichannel linear systems. This state space model preserves desirable characteristics of the residual lattice filter, which include order recursiveness, numerical efficiency for high orders, and robustness with respect to numerical computations. The new model is compared to several prominent methods for identification of a high-order system from noisy input-output data.
The dissertation of Azin Faghihi is approved.

Tsu-Chin Tsao
Robert M’Closkey
Panagiotis Christofides
Steve Gibson, Committee Chair

University of California, Los Angeles
2014
To my parents, Oranous and Farhang.

To Professor Gibson, from whom I have learned so much.

To my committee members, Professors Tsao, M’Closkey, and Christofides for providing insight to my research.

And to my friends, especially Zohreh Karimi, Zahra Aghajan, Sedigheh Hashemi, Nolan Tsuchiya, Fahimeh Fakour, and Mostafa Majidpour, for providing both empathy and perspective.

There are one-story intellects, two-story intellects, and three-story intellects with skylights. All fact collectors, who have no aim beyond their facts, are one-story men. Two-story men compare, reason, generalize, using the labors of fact collectors as well as their own. Three-story men idealize, imagine, predict.

- Oliver Wendell Holmes
**Table of Contents**

1 Introduction .................................................................................................................. 1
   1.1 Identification and Prediction for Aero-Optical Data .............................................. 1
   1.2 Lattice-Filter Based State-Space Model ............................................................... 2

2 Aero-Optical Wavefront Data and Modal Representation ........................................... 4
   2.1 Karhunen-Loève Modes ......................................................................................... 7
   2.2 Mask Modes ........................................................................................................ 16

3 Two System Identification and Prediction Models ....................................................... 23
   3.1 Subspace System Identification .......................................................................... 23
   3.2 Autoregressive System Identification .................................................................. 26

4 Wavefront Prediction and Prediction Filter ................................................................ 28
   4.1 Quantitative Results for Prediction Errors ......................................................... 28
   4.2 State-Space Model with Wavefront Output ......................................................... 31

5 Flow Velocity ............................................................................................................... 33
   5.1 Velocity Estimates from the Original Data ......................................................... 33
   5.2 Velocity Estimates from the State-Space Model .................................................. 37

6 Frequency Response of the Identified Model ............................................................. 42

7 Lattice Filter-Based State-Space Model ...................................................................... 48
   7.1 Lattice Filter Algorithm ...................................................................................... 48
   7.2 Derivation of the State-Space Model .................................................................. 51
7.3 Computation of the State-Space Coefficients

7.3.1 Direct Calculation from Lattice Gains

7.3.2 Batch Least Squares Construction

7.4 An Additional Channel for the Input Sequence

7.4.1 Lattice ARX

7.4.2 Lattice State-Space

8 Identification of a Test System

8.1 Construction of the Test System

8.2 Comparison Models

8.3 Sensitivity to Noise

8.3.1 Process and Measurement Noise

8.3.2 Comparison of Identification Errors

8.3.3 Frequency Response of the Lattice-Filter State-Space Model

8.4 Identification Error and Length of the Data Sequence

8.5 Effects of Single Precision Computation, Model Order, and Balancing

8.6 Computational Complexity

9 Conclusions

Appendix A Autoregressive Exogenous to State-Space Model

Appendix B Order Reduction

Appendix C N4SID: Weighting Schemes and Prediction Horizons

Appendix D Data Points and Convergence
List of Figures

2.1  A typical aero-optical wavefront ................................................. 5
2.2  The temporal variance computed for all pixels with valid phase values ... 6
2.3  Images of selected Karhunen-Loève modes ...................................... 8
2.4  The frequency response of one dimensional FIR filter $f_1$ ................. 9
2.5  The frequency response of two dimensional FIR filter $f_2$ .................. 10
2.6  A diagram of Karhunen-Loève modes construction ............................ 11
2.7  Top: cumulative sum in (2.8) of the values in the bottom plot. Bottom: Squared values of the norm in (2.7) for modal sequences $y_m$ ............... 14
2.8  Images of selected Mask modes .................................................. 17
2.9  Frequency Response of one dimensional FIR filter $f_1$ .................... 18
2.10 The frequency response of two dimensional FIR filter $f_2$ ............... 19
2.11 A diagram of Mask modes construction ....................................... 20
3.1  Demonstration of AR system identification .................................... 27
4.1  Power spectral densities of selected modal and prediction error sequences ................................................................. 29
4.2  Top (linear axes): Normalized temporal mean-square modal prediction error defined by (4.3) for the 80 modes in the identified state-space model. Bottom (semilog axes, vertical axis in dB): squared values of the norm defined by (2.7) for modal sequences and modal prediction error sequences .......... 30
4.3  Virtual wavefronts .................................................................. 32
5.1  Seventeen locations for velocity estimates. Arrow = mean velocity vector ................................................................. 35
5.2  A sample spatial shift of an array in a single frame ......................... 36
5.3  A sample representation of shifted pixels on two frames .................. 36
5.4 Representation of matrix \( K_\tau \), horizontal and vertical shifts. ............... 37

5.5 Plots of spatial shifts \( \Delta_h \) and \( \Delta_v \) vs. time shift \( \tau \). ................. 40

5.6 Velocity estimates in pixels per time step at seventeen locations. ................. 41

6.1 Pixel locations for power spectral density and frequency response plots. .... 44

6.2 Power spectral densities and frequency response curves \( p_k(\omega) \) for pixels UL, DR, UR and DL. ................................................................. 45

6.3 Power spectral densities and frequency response curves \( p_k(\omega) \) for pixels D, U, L and R. ................................................................. 46

6.4 Power spectral densities and frequency response curves \( p_k(\omega) \) for pixels UL, DR, UR and DL. ................................................................. 47

7.1 Constructing lattice-based state-space model from the lattice algorithm. .... 56

8.1 Bode magnitude plot for the transfer function of the butter filters and their sum. ................................................................. 62

8.2 Hankel singular values of the test system. .................................................. 63

8.3 An input-output diagram for the true system. .......................................... 63

8.4 The comparison models for system identification of the test system. ......... 64

8.5 Relative identification error versus the model order. ............................... 66

8.6 A diagram describing the superposition of the output and a colored noise sequence. ................................................................. 67

8.7 A summary of identification steps in the presence of output noise. ............ 68

8.8 Relative identification error plotted versus the noise ratios, on a logarithmic scale. ................................................................. 69

8.9 Normalized relative identification error, on a linear scale. ....................... 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.10</td>
<td>Frequency response for the true and the lattice state-space identified systems for four noise ratios, $\alpha$.</td>
<td>71</td>
</tr>
<tr>
<td>8.11</td>
<td>Relative identification error versus the number of data points.</td>
<td>72</td>
</tr>
<tr>
<td>8.12</td>
<td>Fixed-Point simulation of the identified models.</td>
<td>73</td>
</tr>
<tr>
<td>8.13</td>
<td>Fixed-Point simulation for the balanced realization of the identified models.</td>
<td>74</td>
</tr>
<tr>
<td>8.14</td>
<td>Relative identification error in precision versus order. Left: identified models in their original coordinates. Right: balanced realizations of the identified models</td>
<td>74</td>
</tr>
<tr>
<td>8.15</td>
<td>A schematic of runtime analysis.</td>
<td>75</td>
</tr>
<tr>
<td>8.16</td>
<td>Identification processing time versus the model order, on a logarithmic scale.</td>
<td>76</td>
</tr>
<tr>
<td>8.17</td>
<td>Identification processing time versus the model order, on a linear scale.</td>
<td>77</td>
</tr>
<tr>
<td>8.18</td>
<td>Identification processing time versus the model order for three models, on a linear scale.</td>
<td>78</td>
</tr>
<tr>
<td>C.1</td>
<td>N4SID identification using different weighting methods and prediction horizons.</td>
<td>86</td>
</tr>
</tbody>
</table>
# List of Tables

2.1 Experimental details for the Notre Dame AAOL dataset. .......................... 5

2.2 Dimensions of the wavefront images, modal images, and modal sequence. . 15

5.1 Statistics of velocity estimates for the selected seventeen locations. .......... 34

8.1 Filter properties. ......................................................................................... 61

B.1 State-space dimensions. ............................................................................... 84

C.1 Relative identification error for selected N4SID options. ......................... 85
ACKNOWLEDGMENTS

This work has been partially supported by the High Energy Laser Joint Technology Office and the U.S. Office of Naval Research under grant N00014 07-1-1063 and the Air Force Office of Scientific Research under grant FA9550-12-1-0375. We are indebted to Professor Eric Jumper and the University of Notre Dame aero-optical research group for the aero-optical data used here.
Vita

2006 B.S. in Mechanical Engineering, minor in Theater & Performance Studies, University of California, Berkeley

2009 M.S. in Mechanical Engineering, University of California, Los Angeles

2008–2009 Graduate Student Instructor, Department of Mechanical and Aerospace Engineering, University of California, Los Angeles

2009–2014 Research Assistant, Department of Mechanical and Aerospace Engineering, University of California, Los Angeles

Publications


CHAPTER 1

Introduction

1.1 Identification and Prediction for Aero-Optical Data

Recent adaptive optics (AO) research has shown the use of linear time-invariant (LTI) state-space prediction filters to predict optical wavefronts [1–11]. The prediction filters have the form of a Kalman predictor, but they are not necessarily constructed by the standard Kalman filter/predictor design methods, which require \textit{a priori} estimates of turbulence statistics. Rather, some prediction filters for adaptive optics have been identified directly from wavefront data by subspace system identification [2,3,5,9–11].

This dissertation presents a state-space model and associated minimum-variance prediction filter for a sequence of aero-optical wavefronts measured in the University of Notre Dame’s Airborne Aero-Optics Laboratory (AAOL) [12]. The state-space model of the wavefront disturbance is generated by subspace system identification. Such state-space prediction filters can be used in optimal control loops for adaptive optics. [2,3,9–11] State-space prediction filters of the type presented here have been used recently in adaptive optics experiments for which wavefront disturbance sequences were constructed from aero-optical data. [9–11]

In those experiments, the aero-optical wavefronts were mapped to the reduced and simplified geometry in a laboratory adaptive optics experiment. This dissertation deals only with the original aero-optical wavefronts on their full geometry. The primary purpose of prediction filters like the one here is optimal wavefront prediction in adaptive optics, although the identified state-space model can be used to analyze wavefront dynamics and to generate sequences of artificial wavefronts for use in adaptive optics simulations or experiments. The
purpose is to validate the identified state-space model and associated prediction filter by demonstrating that the model captures the statistics and the flow of the aero-optical data and that the prediction filter yields minimum-variance prediction error.

Chapter 2 describes the aero-optical data and the Karhunen-Loéve modes used to reduce the dimension of the system identification problem and the state-space model. A set of modes, called Mask modes are also presented in this chapter. Chapter 3 explains the system identification methods applied to the wavefronts. Chapter 4 discusses the identified state-space model of the wavefront sequence and the related prediction filter. Chapter 5 describes the estimation of flow velocity at a number of points in the aperture by a correlation method. First, velocity estimates are computed directly from the aero-optical data. Then, it is shown how the second-order statistics required for computing the velocity estimates can be constructed from only the matrices on the state-space model, and the velocity estimates computed by the two methods are compared. Chapter 6 demonstrates how to use the frequency response of the state-space model to determine the power spectrum of a sequence of wavefront images generated by the state-space model without generating the sequence. Plots presented in Chapter 6 compare the appropriate frequency response curves from the identified state-space model to power spectra for representative pixel time series in the original aero-optical data [13].

1.2 Lattice-Filter Based State-Space Model

The residual-error lattice filter is a recursive-least-squares (RLS) filter derived from the Jiang/Gibson lattice filter [14]. The filter gains are updated directly and in stationary conditions converge to constant values. The algorithm is implementable on digital signal processing (DSP) boards, field programmable gate arrays (FPGA), PCs with real-time operating systems, and application-specific integrated circuits (ASIC). The lattice algorithm is numerically stable in multichannel adaptive filtering. The motivation for this dissertation is the need for a state-space representation of the lattice filter that preserves its desirable
properties such as order-recursiveness, numerical efficiency for higher orders, and robustness with respect to numerical computations. A related approach for generalized one-multiplier lattice two-dimensional filter has been demonstrated in [15–17].

In Chapter 7, a lattice filter-based state-space model derived from parameters in the lattice filter is presented. Chapter 8 presents simulation results comparing the new state-space model to the models identified by several common system identification methods.
CHAPTER 2

Aero-Optical Wavefront Data
and Modal Representation

The aero-optical data used in this dissertation came from Notre Dame’s Airborne Aero-Optics Laboratory (AAOL) [12]. The original aero-optical wavefronts were produced by turbulence over a flat-windowed turret [12, 18–20] during a flight test in which a continuous-wave laser was transmitted between two planes flying in a constant formation at an altitude of 4570 m. The planes were separated by approximately 50 m to ensure aero-optical turbulence was the primary source of wavefront aberrations.

The identification results and analysis presented in this dissertation were generated from a sequence of 8000 wavefronts captured at a 16 KHz frame rate. These wavefronts were reconstructed from Shack-Hartmann wavefront sensor measurements. While the predominant source of wavefront aberrations in the AAOL data is the turbulent shear layer around the turret, some small electronic noise is likely present in the data. A detailed analysis of the statistics of the AAOL data has been published [12]. Other significant parameters for the flight test are given in Table 2.1.

Figure 2.1 shows a typical frame from the wavefront sequence used for this project. The wavefront is represented in both two dimensions and three dimensions in order to illustrate the correspondence between variation in color and variation in optical phase. Each wavefront is given on a 29 × 29 pixel array. There is an annular region of 513 pixels with valid phase values. There is no data at the remaining 328 pixels, shown in black in Fig. 2.1. A central obscuration in the optical system produces the hole in the middle of each wavefront. The temporal variance of the wavefronts were computed for each pixel and displayed in Fig. 2.
Figure 2.1: A typical aero-optical wavefront.

Table 2.1: Experimental details for the Notre Dame AAOL dataset.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turret Azimuthal angle</td>
<td>119°</td>
</tr>
<tr>
<td>Turret Elevation angle</td>
<td>57°</td>
</tr>
<tr>
<td>Freestream Mach</td>
<td>0.36</td>
</tr>
<tr>
<td>Altitude</td>
<td>4570 m</td>
</tr>
<tr>
<td>Target Distance</td>
<td>50 m</td>
</tr>
<tr>
<td>Aperture Size</td>
<td>10.1 cm</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>16 kHz</td>
</tr>
</tbody>
</table>
Figure 2.2: The temporal variance computed for all pixels with valid phase values.
2.1 Karhunen-Loéve Modes

For the state-space model presented in this dissertation, the wavefronts were represented as linear combinations of the Karhunen-Loéve modes constructed from a low-pass filtered version of the original wavefront sequence. These modes constitute an orthonormal set of 513 modal images. Figure 2.3 shows images of representative modes. The vector sequence $y(t)$ used here contains the sequences of modal coefficients obtained by projecting the original (unfiltered) wavefront sequence onto the Karhunen-Loéve modes of the filtered wavefront sequence.

The Karhunen-Loéve modes are the eigenvectors of the covariance matrix of the filtered wavefront sequence. Using the filtered wavefront sequence to construct the modes produced smoother low-order modes than did the modes computed from the unfiltered wavefronts, so that the modes separated the low-frequency and high-frequency components of the wavefronts more effectively.

The low-pass spatial filter $f_2$ used in constructing the modes was generated in the following way:

$$f_2 = f_{\text{trans}2}(f_1) \quad \text{or} \quad f_2 = f_1^T f_1,$$

where $f_1$ is the one-dimensional FIR filter,

$$f_1 = \left[ \frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{4} \right].$$

The MATLAB command $f_{\text{trans}2}$ converts a one-dimensional FIR filter to a two-dimensional FIR filter. The low-pass filter $f_1$ has a gain of one-half at one-half the Nyquist frequency and a gain of zero at the Nyquist frequency. The frequency responses of these two filters are plotted in Figs. 2.4 and 2.5, respectively.

Algorithms 1 illustrates how Karhunen-Loéve modes are constructed. The parameters used in this algorithm are listed in Table 2.2.
Figure 2.3: Images of selected Karhunen-Loève modes.
Frequency Response of $f_1 = [ \frac{1}{4} \ 1/2 \ 1/4 ]$

Figure 2.4: The frequency response of one dimensional FIR filter $f_1$
Frequency Response of $f_2 = f'_1 f_1$, where $f_1 = [1/4 \quad 1/2 \quad 1/4]$
Figure 2.6: A diagram of Karhunen-Loève modes construction.

\[ f_1 = \begin{bmatrix} 1/4 & 1/2 & 1/4 \end{bmatrix} \]

\[ f_2 = \text{ftrans2}(f_1) \]

\[ W = F W \]

\[ W = M \Sigma V' \]

\[ WW' = M \Sigma^2 M' \]
For the identification and analysis discussed here, a reduced number of modes were used, and the dimension of $y(t)$ is the number of modes used. Before the Karhunen-Loéve modes and the modal sequences were computed, the temporal mean was subtracted from the 8000-frame wavefront sequence used. Hence each modal sequence and the vector sequence $y(t)$ have zero temporal mean.

The bottom plot in Fig. 2.7 shows the following norm of the individual modal sequences $y_m(t)$ for the first 80 modes:

$$
\|y_m(:)/W = \|y_m(:)/\|W\|_F, \quad 1 \leq m \leq 513,
$$

(2.7)

where $W$ is the $29^2 \times 8000$ matrix in which each column is a vectorized wavefront image and $\|W\|_F$ is the Frobenius norm of $W$. (The mean over $t$ of $W(:,:)t$ is zero, and the mean over $t$ of each scalar sequence $y_m(t)$ is zero.) The top plot in Fig. 2.7 shows the following cumulative sum:

$$
\text{CumSum}(m) = \sum_{i=1}^{m} \|y_i(:)\|_W^2, \quad 1 \leq m \leq 513.
$$

(2.8)

For the numerical results reported here, only the first 80 Karhunen-Loéve modes were used. As Fig. 2.7 suggests, using more modes did not change the results significantly.
**Algorithm 1: How to construct the Karhunen-Loéve modes.**

- Given images, \( \mathcal{W} \).
- Obtain the number of non-zero (or non-NaN) elements of each frame, \( p \).
- Vectorize the non-zero elements of each image, \( \mathcal{W}_i \), into a column vector \( \mathbf{W}_i \).
- Construct matrix \( \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_2 & \cdots & \mathbf{W}_T \end{bmatrix} \) of dimensions \( p \times T \).
- Construct \( \mathbf{Q} \triangleq \mathbf{W} \mathbf{W}' \) of dimensions \( p \times p \).
- Construct matrix \( \mathbf{F} \) by applying \( f_2 \) to non-NaN elements of the images by redefining \( \mathbf{Q} \) as \( \mathbf{Q} = \mathbf{F} \mathbf{Q} \mathbf{F}' \).
- \( Q_{ij} \) is the product of pixel \( i \) time series with pixel \( j \) time series.
- Singular value decomposition on \( \mathbf{Q} = \mathbf{M} \mathbf{\Sigma} \mathbf{V}' \) yields:

\[
\mathbf{M}' = \mathbf{M}^{-1}, \quad \mathbf{V}' = \mathbf{V}^{-1}, \quad \text{unitary matrices}
\]

\[
\mathbf{Q} = \mathbf{W} \mathbf{W}' = \mathbf{M} \mathbf{\Sigma} \mathbf{V}' = \sum_{l=1}^{p} M_l \sigma_l V'_l, \quad \sigma_1 > ... > \sigma_p
\]

\[
\mathbf{M}' \mathbf{W} \mathbf{W}' \mathbf{V} = \mathbf{\Sigma}
\]

- Matrix \( \mathbf{\Sigma} \) contains the singular values of the covariance matrix of the vectorized images, on its diagonal.
- Construct the modal sequence \( \mathbf{Y} = \mathbf{M}' \mathbf{W} \) of dimensions \( p \times T \).

\[
\mathbf{Y}(l,:) = \mathbf{M}' \mathbf{W}
\]

- When reshaped, columns of \( \mathbf{M} \) correspond to the modal images \( \mathcal{M} \).
- The columns are reshaped into matrices by filling out the non-zero pixels.
- Dimensions of the modal images are \( l \times l \times p \).

\[
\mathbf{W} = \mathbf{M} \mathbf{Y} \rightarrow \mathbf{W}_t = \sum_{l=1}^{p} \mathcal{M}_l Y(l,t)
\]

- Each image at time \( t \) can be constructed from the modal sequence at time \( t \) and the modal images.

\[
\mathcal{W}_t = \sum_{l=1}^{p} \mathcal{M}_l Y(l,t)
\]
Figure 2.7: Top: cumulative sum in (2.8) of the values in the bottom plot. Bottom: Squared values of the norm in (2.7) for modal sequences $y_m$. 
Table 2.2: Dimensions of the wavefront images, modal images, and modal sequence.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>number of frames</td>
<td>$T$</td>
<td>8000</td>
</tr>
<tr>
<td>image dimension</td>
<td>$l \times l$</td>
<td>29×29</td>
</tr>
<tr>
<td>wavefront images</td>
<td>$W$</td>
<td>29×29×8000</td>
</tr>
<tr>
<td>vectorized wavefront images</td>
<td>$W'$</td>
<td>29²×8000</td>
</tr>
<tr>
<td>number of pixels in the annular region</td>
<td>$p$</td>
<td>513</td>
</tr>
<tr>
<td>modal images</td>
<td>$M$</td>
<td>29×29×513</td>
</tr>
<tr>
<td>vectorized modal images</td>
<td>$M'$</td>
<td>29²×513</td>
</tr>
<tr>
<td>modal sequence (all channels)</td>
<td>$Y$</td>
<td>513×8000</td>
</tr>
<tr>
<td>number of channels used for identification</td>
<td>$m$</td>
<td>80</td>
</tr>
<tr>
<td>modal sequence used for identification</td>
<td>$y$</td>
<td>80×8000</td>
</tr>
</tbody>
</table>
2.2 Mask Modes

Another method for representing the state-space model is linear combinations of the Mask modes. These modes constitute an orthonormal set of 513 modal images. Figure 2.8 shows images of representative modes. The vector sequence $y(t)$ in this case contains the sequences of modal coefficients obtained by projecting the original wavefront sequence onto the Mask modes.

The first Mask mode was defined to be the piston mode. The second and third modes correspond to the tilt modes in two orthogonal directions. The orthogonal subspace projection of the first three modes were filtered by a two-dimensional filter $f_2$ by applying the command $\text{ftrans2}$ to the one-dimensional FIR filter $f_1 = \text{fir1}(N, \omega_n)$ as described in Algorithm 3. The frequency responses of these two filters are shown in Figs. 2.9 and 2.10, respectively. Algorithm 3 illustrates how the columns of $U_i$ in (2.12) were filtered.
Figure 2.8: Images of selected Mask modes.
Frequency Response of FIR filter $f_1$, order 8 and cut-off frequency 0.25

Figure 2.9: Frequency Response of one dimensional FIR filter $f_1$
Frequency Response of $f_2 = f'_1 f_1$, where $f_1 = \text{fir1}(8, 0.25)$

Figure 2.10: The frequency response of two dimensional FIR filter $f_2$
Figure 2.11: A diagram of *Mask* modes construction.
Algorithm 2: How to construct the Mask modes.

- Define a mask matrix, \( \mathcal{I} \) consisting of zeros and ones at NaN and non-NaN locations of an image, respectively.
- Construct \( \tilde{U}_0 \) from one piston mode and 2 tilt modes.

\[
\tilde{U}_0 = [ \tilde{U}_{\text{piston}} \, \tilde{U}_{\text{tilt},1} \, \tilde{U}_{\text{tilt},2} ] \in \mathbb{R}^{p \times 3} \tag{2.9}
\]

- Apply the command \( \text{qr}(\tilde{U}_0, 0) \) to produce the "economy size" decomposition. Since \( \tilde{U}_0 \) is a tall matrix, only the first 3 columns of \( U_0 \) and the first 3 rows of \( R \) are computed:

\[
\tilde{U}_0 = U_0 R \quad , \quad U_0 \in \mathbb{R}^{p \times 3} \quad , \quad R \in \mathbb{R}^{3 \times 3} \tag{2.10}
\]

- Compute the orthogonal subspace projection onto the nullspace of \( U_0' \):

\[
P = I - \tilde{U}_0 (\tilde{U}_0' \tilde{U}_0)^{-1} \tilde{U}_0' = I - U_0 U_0' \tag{2.11}
\]

- Compute the singular value decomposition of the orthogonal subspace; since the last 3 singular values are zero, truncated matrices \( U_t \) and \( V_t \) are computed:

\[
P = U \Sigma V' = [ U_t \quad U_2 ] \begin{bmatrix} \Sigma_t & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{t1}' \\ V_{t2}' \end{bmatrix} = U_t \Sigma_t V_t' \tag{2.12}
\]

- Compute the diagonal matrix of eigenvalues \( \Lambda \) and eigenvectors \( V \) of \( U_{t,f}' U_{t,f} \), where \( U_{t,f} \) is matrix \( U_t \) after it has been filtered as explained in Algorithm 3:

\[
U_{t,f}' U_{t,f} V = \Lambda \tag{2.13}
\]

- Multiply the columns of \( U_t \) by the matrix of eigenvectors:

\[
U_{t,f} V = [ X_1 \quad X_2 \quad \cdots \quad X_{p-3} ] \tag{2.14}
\]

- Augment the piston and tilt modes with the rest of the mode vectors.

\[
M = [ U_0 \quad X_{p-3} \quad \cdots \quad X_1 ] \tag{2.15}
\]

- Reshape each column of \( M \) into the image size matrix \( \mathcal{M} \) called a modal image.

\[
M_i \xrightarrow{\text{reshape}} \mathcal{M}_i \quad , \quad i \in 1, \cdots, p \tag{2.16}
\]
Algorithm 3: How to filter the mode shapes.

- Design a low-pass FIR filter $f_1 = \text{fir1}(N, \omega_n)$ with order $N$ and cut-off frequency $\omega_n$ as shown in Fig. 2.9.
- Make a 2-D FIR filter $f_2 = f'_1 f_1$ using the McClellan frequency transformation. Columns of filter $\mathcal{F}$ are constructed using $f_2$ on pixels of $\mathcal{I}$ shown in Fig. 2.10.
- Filter $U_t$:

$$U_{t,f} = \mathcal{F}U_t$$ (2.17)
CHAPTER 3

Two System Identification and Prediction Models

3.1 Subspace System Identification

State-space subspace methods (21–23) can be used to identify state-space models for vector data sequences. The subspace method in [24], which is based on a recursive least-squares lattice filter, is used here to identify disturbance and prediction models for aero-optical wavefront sequences. This method is summarized in Algorithm 4 below.

The identified model has the following form:

\[
\begin{align*}
    x(t + 1) &= Ax(t) + B\varepsilon(t), \\
    y(t) &= Cx(t) + \varepsilon(t). \\
\end{align*}
\tag{3.1}
\]

This is a linear time-invariant (LTI) discrete-time system with input vector \( \varepsilon(t) \), output vector \( y(t) \) and state vector \( x(t) \). The matrices \( A \) and \( A - BC \) are stable; i.e., all the eigenvalues have magnitude less than 1. The system identification procedure, illustrated in Algorithm 4 produces the matrices \( A \), \( B \) and \( C \) along with the covariance matrix of \( \varepsilon \), denoted by \( Q_{\varepsilon\varepsilon}(0) \).

The one-step prediction of \( y(t) \) is

\[
    \tilde{y} = Cx(t),
\tag{3.2}
\]

which is the estimate of \( y(t) \) based on measurements \( y(\tau) \), \( \tau = t - 1, t - 2, \ldots \).

Rewriting the second equation in (3.1) as

\[
    \varepsilon(t) = y(t) - Cx(t) = y(t) - \tilde{y}(t)
\tag{3.3}
\]
shows that $\varepsilon$ is the prediction error or *innovations* sequence. Ideally, $\varepsilon$ is temporally white noise, though usually correlated in space. The state-space model in (3.1) can be re-arranged by using (3.2) and (3.3) to yield the prediction filter

$$
\begin{align*}
    x(t+1) &= [A - BC]x(t) + By(t) \\
    \tilde{y}(t+1) &= C[A - BC]x(t) + CB y(t).
\end{align*}
$$

(3.4)

This filter has the standard form of an LTI state-space model; at sample time $t$, the input is $y(t)$ and the output is $\tilde{y}(t+1)$, which is the one-step prediction of $y(t+1)$. After the matrices $A, B$ and $C$ are identified, the innovations sequence $\varepsilon = y - \tilde{y}$ can be computed with $\tilde{y}$ generated by (3.4). Then $Q_{\varepsilon\varepsilon}(0)$ is computed from $\varepsilon$. 

24
### Algorithm 4: Subspace system identification.

- **QR factorization:**
  \[
  W = \begin{bmatrix} U_f & W_p & Y_f \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix}
  \]  
  \[ (3.5) \]

- **Compute the Hankel matrix; Determine the system order:** compute the singular value decomposition:
  \[
  H = (R_{22}^{-1}R_{23})^T, \quad R_{23}^T = HR_{22}^T = U\Sigma V^T
  \]  
  \[ (3.6) \]

  - Determine the system order \( n \) from \( \Sigma \).
  - \( U_r \) and \( V_r \) are the first \( n \) columns of \( U \) and \( V \) and \( \Sigma_r \) is the first \( n \times n \) block of \( \Sigma \).
  - Compute the extended observability/controllability matrices:
    \[
    \mathcal{O}(M) = U_r\Sigma_r^{1/2}, \quad \mathcal{C}(N) = \Sigma_r^{1/2}V_r^T R^{-T}
    \]  
    \[ (3.7) \]

  - Compute the system matrices \( (A_o, B^u, C) \):
    \[
    A_o : \quad \mathcal{O}_1(M)\dagger \mathcal{O}(M) \\
    B : \quad \text{first left block column of } \mathcal{C}(N) \\
    B^u : \quad \text{from } B = \begin{bmatrix} F & B^u \end{bmatrix} \\
    C : \quad \text{first top block row of } \mathcal{O}(M)
    \]  
    \[ (3.8) \]
3.2 Autoregressive System Identification

The autoregressive (AR)-based state-space model is derived from the following relation:

\[
\tilde{y}(t) = \sum_{k=1}^{N} \tilde{A}_k y(t-k).
\]  

(3.9)

Substituting \( \tilde{y}(t) = y(t) - \varepsilon(t) \) in the previous equation yields

\[
y(t) = \sum_{k=1}^{N} \tilde{A}_k y(t-k) + \varepsilon(t).
\]  

(3.10)

The coefficients \( \tilde{A}_k \) are used to compute a state-space model for the vector sequence \( y(t) \). Figure 3.1 shows the steps involved in the system identification.

This is a an LTI discrete-time system with input vector \( \varepsilon(t) \), output vector \( y(t) \), and state vector \( x(t) \). The system identification procedure produces the matrices \( A, B \) and \( C \) shown in (3.11). Appendix A describes this conversion in detail.

\[
A = \begin{bmatrix}
\tilde{A}_1 & I & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\tilde{A}_{N-1} & 0 & \cdots & I \\
\tilde{A}_N & 0 & \cdots & 0 \\
\end{bmatrix}, \quad B = \begin{bmatrix}
\tilde{A}_1 \\
\vdots \\
\tilde{A}_{N-1} \\
\tilde{A}_N \\
\end{bmatrix}, \quad C = \begin{bmatrix}
I & 0 & \cdots & 0 \\
\end{bmatrix}
\]  

(3.11)

The expressions for the one-step prediction, \( \tilde{y}(t) \), the prediction error, \( \varepsilon \), and the prediction filter are the same as those described in the subspace identification section.
Figure 3.1: Demonstration of AR system identification.
CHAPTER 4

Wavefront Prediction and Prediction Filter

4.1 Quantitative Results for Prediction Errors

The best indication of how accurately the statistics associated with the identified state-space model matches the statistics of the original data is how close the prediction error sequence $\varepsilon$ is to being white. Also, a minimum-variance predictor theoretically should produce a white prediction error sequence. Fig. [4.1] shows the power spectral densities of representative modal sequences $y_m$ and the corresponding prediction error sequences $\varepsilon_m$, where $m$ is the mode number. The prediction errors are nearly, though not perfectly, white. The variation of the prediction error power spectral densities (PSD) in Fig. [4.1] from a straight line is not much greater than is seen for a white sequence of length 8000 produced by a random number generator.

The fraction of the total wavefront power contained in the first 80 modes is

$$\sum_{m=1}^{80} \frac{\|y_m(\cdot)\|^2}{\|W\|^2_F} = 0.9611,$$

(4.1)

and the fraction of the power in the first 80 modes contained in the prediction error is

$$\sum_{m=1}^{80} \frac{\|\varepsilon_m(\cdot)\|^2}{\sum_{m=1}^{80} \|y_m(\cdot)\|^2} = 0.1799.$$

(4.2)

The normalized temporal mean-square modal prediction error, defined by

$$\varepsilon_m^2 = \frac{\|\varepsilon_m(\cdot)\|^2}{\|y_m(\cdot)\|^2} = \frac{\|\varepsilon_m(\cdot)\|^2/W}{\|y_m(\cdot)\|^2/W},$$

(4.3)
Figure 4.1: Power spectral densities of selected modal and prediction error sequences.

provides a more detailed description of the prediction error. The top plot in Fig. 4.2 shows $\bar{\varepsilon}_m^2$ for the 80 modes in the identified state-space model.

The value of $\bar{\varepsilon}_m^2$ generally increases with mode number because the bandwidth of the modal sequences generally increases with mode number, as the representative power spectral densities in Fig. 4.1 show.

The bottom plot in Fig. 4.2 puts in perspective the larger relative prediction errors for the higher-order modes by emphasizing how small the power in the higher-order modes is compared to the power in the lower modes. The values of $\|y_m(\cdot)\|^2_W$ plotted in Fig. 4.2 are the same as those plotted in Fig. 2.7 but these values are especially important for interpreting the significance of the values of $\bar{\varepsilon}_m^2$ for the higher mode numbers.

The magnitude of the prediction error $\varepsilon$ and the magnitudes of its modal components are important characteristics of the identified state-space model. However, these will vary with the statistics of the wavefront sequence. The whiteness of the prediction error sequence
Figure 4.2: Top (linear axes): Normalized temporal mean-square modal prediction error defined by (4.3) for the 80 modes in the identified state-space model. Bottom (semilog axes, vertical axis in dB): squared values of the norm defined by (2.7) for modal sequences and modal prediction error sequences.
remains the main indication of how well the identified state-space model represents the
statistics of the original wavefront data.

4.2 State-Space Model with Wavefront Output

If the matrix $M$ contains the vectorized modal images as its columns, then the sequence

$$w(t) = My(t)$$

(4.4)

is a sequence of vectorized wavefront images corresponding to the sequence $y$ of modal
coefficient vectors. With the matrix $C_w$ defined by

$$C_w = MC,$$  \hspace{1cm} (4.5)

the state-space model in (3.1) can be converted to

$$x(t+1) = Ax(t) + K\varepsilon(t),$$
$$w(t) = C_w x(t) + M\varepsilon(t), \hspace{1cm} t = 0, 1, 2, \ldots,$$  \hspace{1cm} (4.6)

which is a state-space model that generates a vectorized wavefront image as the output at
each $t$; i.e., each component of $w(t)$ is a pixel value.

Although state-space models and the associated prediction filters like those here have
been used primarily for wavefront prediction and control, the state-space model in the form
in (4.6) can be used to generate sequences of artificial wavefronts with statistics very nearly
identical to the statistics of the original wavefront data sequence. For this, a random number
generator is used to produce a new white Gaussian vector sequence $\varepsilon(t)$ with the covariance
$Q_{\varepsilon\varepsilon}(0)$ determined in the system identification procedure. Then the model in (4.6) is driven
with the new sequence $\varepsilon(t)$, and the sequence of output vectors $w(t)$ are reshaped to obtain a
sequence of artificial wavefronts. Sequences of images produced in this way can be arbitrarily
long without repeating, and an arbitrary number of independent sequences can be produced
by generating independent white sequences $\varepsilon(t)$. In side-by-side animations of an artificial
wavefront sequence constructed as described here and of the original aero-optical data, the
Figure 4.3: Virtual wavefronts.

two sequences appear to have the same (although independent) statistics. Notably, the animation of the artificial wavefront sequence in Fig. 4.3 appears to show the same flow velocity observed in the original data.
CHAPTER 5

Flow Velocity

The flow velocity is estimated for a $2 \times 2$ group of pixels at each of the 17 locations shown in Fig. 5.1. One set of velocity estimates is computed from the original wavefront data, and a second set of velocity estimates is computed from the identified state-space model. Sections 5.1 and 5.2 describe the computation of the velocity estimates. Figure 5.6 compares the two sets of velocity estimates, and Table 5 gives statistics of the two sets of velocity estimates and their differences, where $1 \leq k \leq 17$. The terms $v_{\text{data}}(k)$ and $v_{\text{model}}(k)$ represent the velocities constructed directly from wavefront data and from identified state-space model, respectively. The arrow in Fig. 5.1 represents the mean of the 17 local velocity vectors computed from the state-space model. This velocity vector is almost identical to the mean velocity computed from the original data. The values of the mean velocity estimates are given in Fig. 5.6 and Table 5. It is noteworthy that that the mean velocity estimate from the original data and the mean velocity estimate from the state-space model differ by less than 2% for both the horizontal and vertical directions [13].

5.1 Velocity Estimates from the Original Data

The velocity estimates from the original data are computed by a correlation method similar to a method used previously for estimating the flow of turbulence-induced optical wavefronts [25]. Several locations in wavefront are chosen for calculating the velocity. Each location is an array of 4 pixels. Figure 5.2 shows the spatial shift for a combination of horizontal and vertical shifts.
Table 5.1: Statistics of velocity estimates for the selected seventeen locations.

<table>
<thead>
<tr>
<th>Vertical</th>
<th>Horizontal</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean velocity (pixels/time step)</td>
<td>0.4698</td>
</tr>
<tr>
<td>from data</td>
<td>0.4745</td>
</tr>
<tr>
<td>from model</td>
<td>0.0941</td>
</tr>
<tr>
<td>( \frac{\text{std}(v_{\text{model}} - v_{\text{data}})}{</td>
<td>\text{mean}(v_{\text{data}})</td>
</tr>
<tr>
<td>( \max_k \frac{</td>
<td>v_{\text{model}}(k) - v_{\text{data}}(k)</td>
</tr>
</tbody>
</table>

The phase values at each of the \( 2 \times 2 \) pixel groups is shifted by one, two and three time steps.

\[
(\Delta_h, \Delta_v, \tau), \quad \tau \in \{-3, -2, -1, 0, 1, 2, 3\}, \tag{5.1}
\]

where \( \tau \) is the time shift and \( \Delta_h \) and \( \Delta_v \) are the horizontal and vertical spatial shifts, respectively. The row vector \( w_i \) denotes the time sequence for pixel \( i \) with dimensions \( 1 \times T \). The sequence for pixel \( i \) shifted \( \Delta_h \) units horizontally and \( \Delta_v \) units vertically is denoted by \( w_j \). Sample pixels \( i \) and \( j \) are shown in Fig. 5.3.

A matrix of spatial correlation coefficients, \( K_\tau \), is defined for each time shift \( \tau \). Such a matrix is shown in Fig. 5.3. For each time shift, a spatial shift is determined to maximize the correlation between the unshifted pixels and those shifted in time and space. Rows and columns of this matrix correspond to the vertical and horizontal shifts, respectively. The cross-covariance of the unshifted sequence, \( w_i \) with the shifted sequence \( w_j \) are computed by the MATLAB function \texttt{xcov} for the specified time lag \( \tau \in \{-\tau_{\text{max}}, \cdots, -1, 0, 1, \cdots, \tau_{\text{max}}\} \) as shown in (5.2). The average of this value is calculated for all the shifted pixels.
Figure 5.1: Seventeen locations for velocity estimates. Arrow = mean velocity vector.

\[
\rho_{w_iw_j}(\tau) = \left\{ \frac{x\text{cov}(w_i,w_j,\tau)}{\sqrt{x\text{cov}(w_i,0)x\text{cov}(w_j,0)}} \right\}_{2\tau+1} 
\tag{5.2}
\]

Correlation values corresponding to spatial shifts for fractions of pixels are approximated by interpolation from the correlations computed for shifts of integer numbers of pixels.

The horizontal and vertical shifts \((\delta_h(\tau),\delta_v(\tau))\) corresponding to the maximum correlation coefficient are found for each time shift and is shown in (5.3).

\[
K_\tau(\delta_v,\delta_h) = \max_{\delta_h,\delta_v} K_\tau 
\tag{5.3}
\]
Figure 5.2: A sample spatial shift of an array in a single frame.

Figure 5.3: A sample representation of shifted pixels on two frames.
Values of \((\delta_h(\tau)\) and \(\delta_v(\tau)\)) versus \(\tau\) are plotted individually. The estimated velocity components are the slopes of lines determined by least-squares fit to the points plotted in Fig. 5.5.

### 5.2 Velocity Estimates from the State-Space Model

One way to obtain velocity estimates from the identified state-space model in Section 3.1 would be to generate a white vector sequence \(\varepsilon\) with the covariance \(Q_{\varepsilon \varepsilon}(0)\) that was determined from the data as described in Chapter 4 and use this white sequence to drive the state-space model in (4.6) and generate a new wavefront sequence \(w\). This artificial wavefront sequence then could be used exactly as the original data was used to compute velocity estimates. But this is not how velocity estimates are obtained here from the state-space model.

The correlation approach used to compute velocity estimates from the original data is used to compute velocity estimates from the identified state-space model, but with an
important difference: The correlations corresponding to temporal shifts and whole pixel shifts are computed as described below from certain covariance matrices that are constructed directly from the state-space model and the covariance matrix $Q_{\varepsilon\varepsilon}(0)$. No artificial wavefront sequence is generated, and there is no return to the original data.

For vector sequences $\xi$ and $\eta$ of the same dimensions, the following notation is used henceforth:

$$Q_{\xi\eta}(\tau) = E[(z^{\tau}\xi)\eta'], \quad \tau \in \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

(5.4)

where $E[\cdot]$ is expectation and $z^\tau$ is the time shift operator for $\tau > 0$ and backward shift operator for $\tau < 0$. Since each sequence used for identification and analysis in this dissertation have zero temporal means, each $Q_{\xi\eta}(\tau)$ is a covariance matrix. Also, since the sequence $\varepsilon$ generated by the identification procedure is very nearly white, and ideally it should be white, only $Q_{\varepsilon\varepsilon}(0)$ is used here.

It follows from the state-space model in (3.1) and the hypothesis that $\varepsilon$ is white that $Q_{xx}$ satisfies the Lyapunov equation

$$AQ_{xx}(0)A' - Q_{xx}(0) = -BQ_{\varepsilon\varepsilon}(0)B'.$$

(5.5)

The following formulas also follow from (3.1) and the whiteness of $\varepsilon$: For $t > 0$,

$$x(t) = A^tx(0) + \sum_{l=1}^{t} A^{t-l} B\varepsilon(t-l),$$

(5.6)

$$Q_{\varepsilon\varepsilon}(t) = Q'_{\varepsilon\varepsilon}(-t) = E[x(t)\varepsilon'(0)] = A^{t-1}BQ_{\varepsilon\varepsilon}(0),$$

(5.7)

$$Q_{xx}(t) = Q'_{xx}(-t) = E[\varepsilon(t)x'(0)] = 0,$$

(5.8)

$$Q_{xx}(t) = Q'_{xx}(-t) = E[x(t)x'(0)] = A^tQ_{xx}(0).$$

(5.9)

Therefore,

$$Q_{xx}(\tau) = \begin{cases} A^\tau Q_{xx}(0) & , \tau \geq 0 \\ Q_{xx}(0)(A^{-\tau})' & , \tau < 0 \end{cases}$$

(5.10)
\[ Q_{xx}(\tau) = \begin{cases} A^{\tau-1}BQ_{xx}(0), & \tau > 0 \\ 0, & \tau \leq 0 \end{cases} \] (5.11)

\[ Q_{xx}(\tau) = \begin{cases} 0, & \tau \geq 0 \\ Q_{xx}(0)B'(A^{-\tau-1})', & \tau < 0 \end{cases} \] (5.12)

Computing the autocovariance matrix for the output \( y \),

\[ Q_{yy}(\tau) = E[z^{-\tau}yy'] = CQ_{xx}(\tau)C' + Q_{xx}(\tau) + CQ_{xx}(\tau) + Q_{xx}(\tau)C' \] (5.13)

If \( w \) is the sequence of vectorized images related to the sequence \( y \) by (4.4), then the covariance matrix sequence for this sequence of images is

\[ Q_{ww}(\tau) = E[z^{-\tau}M_{yi}yy'M_{j}'] = M_{i}Q_{yy}(\tau)M_{j}' \] (5.14)

where \( w_{i} \) is the vectorized images for location \( i \) and \( M_{i} \) is the row of vectorized modal images \( M \) corresponding to pixel \( i \).

\[ \rho_{ww}(\tau) \triangleq \frac{Q_{ww}(\tau)}{\sqrt{Q_{ww}(0)}\|Q_{ww}(0)}} \] (5.15)

\[ K_{\tau}(\Delta_h, \Delta_v) = \frac{1}{n} \sum_{i} w_{i}w_{j}(\tau) \] (5.16)

Similar procedures as those mentioned in the previous section and \( 5.3 \) are followed to estimate the horizontal and vertical velocities.

Figure 5.5 shows the horizontal and vertical shifts that maximize the correlation coefficient matrix derived from using the original data and the state space model at each time shift.

The estimated slopes are plotted for each specified location in Figure 5.6.
Figure 5.5: Plots of spatial shifts $\Delta_h$ and $\Delta_v$ vs. time shift $\tau$. 
Figure 5.6: Velocity estimates in pixels per time step at seventeen locations.
A matrix $S$ is chosen such that
\[ S Q_{\tilde{\epsilon} \tilde{\epsilon}}(0) S' = I. \] (6.1)

The sequence $\tilde{\epsilon}$ then is defined by
\[ \tilde{\epsilon} = S \epsilon, \] (6.2)
so that $\tilde{\epsilon}$ has the covariance matrix
\[ Q_{\tilde{\epsilon} \tilde{\epsilon}}(0) = I. \] (6.3)

The state-space model in (4.6) now can be written
\[
    x(t+1) = Ax(t) + KS^{-1} \tilde{\epsilon}(t), \quad t = 0, 1, 2, \ldots,
\]
\[
    w(t) = C_w x(t) + MS^{-1} \tilde{\epsilon}(t).
\] (6.4)

It should be recalled that each channel of the sequence $w$ is a time history of values for a single pixel in the sequence of images represented by $w$.

Since the state-space model in (6.4) is a multi-input-multi-output LTI system, it has a transfer matrix $G$. For integers $k$ and $l$, the frequency response of the scalar transfer function from channel $l$ of $\tilde{\epsilon}$ to channel $k$ of $w$ is denoted by $G_{kl}(i\omega)$. As $\tilde{\epsilon}$ is a white sequence with the covariance in (6.3), it follows from standard results in linear system theory that the square root of the power spectral density of the output channel $w_k$ is given by
\[
    p_k(\omega) = \sqrt{\sum_{l} |G_{kl}(i\omega)|^2}. \] (6.5)
The frequency response function \( p_k(\omega) \) is the power spectral density that the identified state-space model predicts for a single pixel time series in the sequence of wavefront images. Since \(|G_{kl}(i\omega)|\) is just the Bode magnitude function for the scalar transfer function \( G_{kl} \), the function \( p_k(\omega) \) can be computed easily.

For eight pixels at the locations shown in Figs. 6.1, 6.2 and 6.3 show the power spectral densities of the original data sequences along with the frequency response curves \( p_k(\omega) \) computed according to (6.5). The pixel locations are also indicated in the images that accompany the plots in Figs. 6.2 and 6.3. It should be recalled from Chapter 5 that the flow direction is approximately along the direction from pixel DR (lower right) to pixel UL (upper left).

The plots in Figs. 6.2 and 6.3 for the different pixel locations show both general similarities but also substantial variation over the aperture of the temporal statistics. In particular, the top two plots in Fig. 6.2 for pixels UL and DR, indicate that the amplitude is significantly larger at all frequencies at the upper left. This results from the fact that the amplitude of the shear layer disturbance increases as the flow moves around the turret and across the aperture with the average velocity found in Chapter 5.

As shown in Figs. 6.2 and 6.3, the PSDs of the original pixel data sequences and the frequency response curves for the identified state-space model show peaks at approximately 3300 Hz, 6200 Hz and 6500 Hz. Possibly, these peaks are due to vortex shedding, but they might be due to electrical noise or structural vibration rather than aero-optical effects. Whatever the source, the amount of power associated with these peaks is very small compared to the power below 3000 Hz.

The plots in Figs. 6.2 and 6.3 have a linear frequency scale to show details at all frequencies. However, this scale can be misleading at frequencies near zero. Figure 6.4 shows the PSDs and frequency response curves for the pixels in Fig. 6.2 but with a logarithmic frequency scale. These plots reveal that the identified state-space model has a nonzero DC gain at each pixel. The DC gains appear to be roughly consistent with the data at low frequencies, but the PSDs and the frequency response curves are somewhat questionable at
low frequencies. Since there is relatively little power below 500 Hz in the original data and because the data sequence represents only a half second, the data is insufficient for computing the PSDs that are highly accurate at low frequencies or identifying a model whose frequency response is highly accurate at low frequencies.
Figure 6.2: Power spectral densities and frequency response curves $p_k(\omega)$ for pixels UL, DR, UR and DL.
Figure 6.3: Power spectral densities and frequency response curves $p_k(\omega)$ for pixels D, U, L and R.
Figure 6.4: Power spectral densities and frequency response curves $p_k(\omega)$ for pixels UL, DR, UR and DL.
CHAPTER 7

Lattice Filter-Based State-Space Model

The lattice algorithm consists of two algorithms, namely the residual error lattice and the model-parameter algorithms. The first algorithm includes the forward- and backward-propagating blocks and is computed in each time step; however, the model-parameter algorithm is executed when the model parameters are required. In this chapter, a state-space model is derived from the parameters in the lattice FIR filter [14] and is used for system-identification in Chapter 8.

7.1 Lattice Filter Algorithm

The lattice filter algorithm consists of forward and backward propagating blocks and the following parameters: $\hat{\alpha}_n^{ki}$, $\hat{\beta}_n^{ki}$, $\bar{\alpha}_n^{ki}$, and $\bar{\beta}_n^{ki}$. The forward blocks relates $\hat{r}_{n,k}^i$ and $\hat{e}_{n,k}^i$ as illustrated in Algorithm 5. Similarly, the backward block relates $\bar{r}_{n,k}^i(t)$ and $\bar{e}_{n,k}^i(t)$ in Algorithm 6. The order initialization and connection procedures are described in Algorithm 7.
**Algorithm 5:** The forward propagating block in the FIR lattice filter.

\[
\text{while } t \geq 0 , 1 \leq n \leq N \text{ do} \\
\quad \text{for } k = 1 \text{ to } q \text{ do} \\
\quad \quad \text{for } i = 1 \text{ to } q + 1 \text{ do} \\
\quad \quad \quad \hat{r}_{n;k}^i(t) = \hat{r}_{n;k-1}^i(t) - \hat{\beta}_{n;i}^k(t)\hat{e}_{n;k-1}^k(t) \\
\quad \quad \end{align}
\]

(7.1)

\[
\quad \quad \text{end} \\
\quad \quad \text{for } i = k + 1 \text{ to } q \text{ do} \\
\quad \quad \end{align}
\]

(7.2)

\[
\quad \quad \text{end} \\
\quad \end{align}
\]

\[
\text{end} \\
\text{end}
\]

**Algorithm 6:** The backward propagating block in the FIR lattice filter.

\[
\text{while } t \geq 0 , 1 \leq n \leq N \text{ do} \\
\quad \text{for } k = 1 \text{ to } q + 1 \text{ do} \\
\quad \quad \text{for } i = 1 \text{ to } q \text{ do} \\
\quad \quad \quad ˇe_{n;k}^i(t) = ˇe_{n;k-1}^i(t) - ˇ\beta_{n;i}^k(t)ˇr_{n;k-1}^k(t) \\
\quad \quad \end{align}
\]

(7.3)

\[
\quad \quad \text{end} \\
\quad \quad \text{for } i = k + 1 \text{ to } q + 1 \text{ do} \\
\quad \quad \end{align}
\]

(7.4)

\[
\quad \quad \text{end} \\
\quad \end{align}
\]

\[
\text{end} \\
\text{end}
\]
**Algorithm 7:** Order initialization and connection in the FIR lattice filter.

```plaintext
while \( t \geq 0 \) do
  for \( i = 1 \) to \( q \) do
    **Order Initialization**
    \[
    \tilde{e}_0:q+1 = \hat{w}^i \\
    \tilde{r}_0:q = \hat{w}^i \\
    \tilde{r}_{0:q+1} = 0
    \]
  end
  **Order Connection**
  \( 0 \leq n \leq N - 1 \)
  for \( i = 1 \) to \( q \) do
    \[
    \tilde{e}_i^{n+1} = \tilde{e}_i^{n+1} = \tilde{e}_i^{n+1} 
    \]
  end
  for \( i = 1 \) to \( q + 1 \) do
    \[
    \tilde{r}_i^{n+1} = \tilde{r}_i^{n+1} = z^{-1}\tilde{r}_i^{n+1}
    \]
  end
end
```
7.2 Derivation of the State-Space Model

The parameters in the forward and backward blocks are vectorized; i.e.,

\[
\begin{bmatrix}
\hat{r}^1_{n:k} \\
\vdots \\
\hat{r}^q_{n:k} \\
\hat{r}^{q+1}_{n:k}
\end{bmatrix}, \quad \hat{e}^1_{n:k} = \begin{bmatrix}
\hat{e}^1_{n:k} \\
\vdots \\
\hat{e}^q_{n:k} \\
\hat{e}^{q+1}_{n:k}
\end{bmatrix}, \quad \tilde{r}^1_{n:k} = \begin{bmatrix}
\tilde{r}^1_{n:k} \\
\vdots \\
\tilde{r}^q_{n:k} \\
\tilde{r}^{q+1}_{n:k}
\end{bmatrix}, \quad \tilde{e}^1_{n:k} = \begin{bmatrix}
\tilde{e}^1_{n:k} \\
\vdots \\
\tilde{e}^q_{n:k} \\
\tilde{e}^{q+1}_{n:k}
\end{bmatrix}.
\] (7.9)

The vector \( \hat{w} \) is

\[
\hat{w} = \begin{bmatrix}
\hat{w}^1 \\
\vdots \\
\hat{w}^q
\end{bmatrix}.
\] (7.10)

The gains are also formed into the following arrays:

\[
\tilde{\alpha}_n = \begin{bmatrix}
0 & \hat{\alpha}^1_n & \cdots & \hat{\alpha}^q_n \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \hat{\alpha}^{q-1}_n \\
0 & 0 & \cdots & 0
\end{bmatrix}, \quad \tilde{\beta}_n = \begin{bmatrix}
\hat{\beta}^1_n & \hat{\beta}^1_{n+1} & \cdots & \hat{\beta}^{q+1}_{n+1} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\beta}^{q+1}_n & \hat{\beta}^{q+1}_{n+1} & \cdots & 0
\end{bmatrix},
\] (7.11)

\[
\tilde{\alpha}_n = \begin{bmatrix}
0 & \tilde{\alpha}^1_n & \cdots & \tilde{\alpha}^q_{n+1} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \tilde{\alpha}^{q-1}_{n+1} \\
0 & 0 & \cdots & 0
\end{bmatrix}, \quad \tilde{\beta}_n = \begin{bmatrix}
\hat{\beta}^1_n & \hat{\beta}^2_n & \cdots & \hat{\beta}^q_n \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\beta}^q_n & \hat{\beta}^{q+1}_n & \cdots & 0
\end{bmatrix},
\] (7.12)

The matrix \( 1^k_p \) is defined as a square \( p \times p \) matrix with one nonzero entry at the \((k,k)\) position; i.e.,

\[
1^k_p \triangleq \begin{bmatrix}
\ddots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
\cdots & \ddots & \ddots & \ddots
\end{bmatrix}.
\] (7.13)
The forward and backward blocks are rewritten in vectorized form as shown in Algorithm 8. In the next step, the propagation equations are simplified in Algorithm 9.

The terms \( \hat{e} \) and \( \hat{r} \) in the forward equations (7.20) are combined in a single equation:

\[
\hat{r}_{n;q}(t) = \frac{\hat{r}_{n;0}(t)}{\hat{r}_{n-1;q}(t-1)} - \beta_n \sum_{i=1}^{q} \frac{i}{I_q} \prod_{j=1}^{l-1} \frac{\hat{e}_{n;0}(t)}{\hat{e}_{n-1;q+1}(t)} \left( I_q - \hat{\alpha}_n^i 1_q^i \right) \hat{r}_{n;0}(t). \tag{7.22}
\]

Similarly, connecting the terms \( \check{e} \) and \( \check{r} \) in the backward equations (7.21) yields the following equation:

\[
\check{e}_{n+1;1}(t) = \frac{\check{e}_{n;0}(t)}{\check{e}_{n+1;2}(t)} - \beta_n \sum_{i=1}^{q+1} \frac{i}{I_q} \prod_{j=1}^{l-1} \frac{\check{r}_{n;0}(t)}{\check{r}_{n-1;q}(t-1)} \left( I_q + 1 - \hat{\alpha}_n^i 1_q^i \right) \check{r}_{n;0}(t). \tag{7.23}
\]

The terms \( \check{e} \) is now written as summations of \( \hat{r} \) and the sequence \( \hat{w} \):

\[
\check{e}_{n+1}(t) = -\tilde{Q}_n \hat{r}_{n-1;1}(t) + \check{e}_{n-1;1}(t) \\
= -\sum_{\eta=1}^{n} \tilde{Q}_\eta \hat{r}_{\eta-1;1}(t) + \check{e}_{0;1}(t). \tag{7.24}
\]

Matrices \( \hat{Q} \) and \( \tilde{Q} \) are defined as follows:

\[
\hat{Q}_n = \beta_n \sum_{i=1}^{q} \frac{i}{I_q} \prod_{j=1}^{l-1} \left( I_q - \hat{\alpha}_n^i 1_q^i \right),
\]

\[
\tilde{Q}_n = \beta_n \sum_{i=1}^{q+1} \frac{i}{I_q} \prod_{j=1}^{l-1} \left( I_q + 1 - \hat{\alpha}_n^i 1_q^i \right). \tag{7.25}
\]

A single equation describes how \( \hat{r} \) is updated; i.e.,

\[
\hat{r}_{n;1}(t) = \hat{r}_{n-1;1}(t) - \hat{Q}_n \check{e}_{n-1;1}(t) \\
= \hat{r}_{n-1;1}(t) + \hat{Q}_n \sum_{\eta=1}^{n-1} \tilde{Q}_\eta \hat{r}_{\eta-1;1}(t) - \tilde{Q}_n \hat{w}(t). \tag{7.26}
\]

The state vector \( x_{n+1} \) at time \( t+1 \) is defined to be \( \hat{r}_{n;1} \) at time \( t \),

\[
x_{n+1}(t+1) \triangleq \hat{r}_{n;1}(t), \quad 0 \leq n \leq N - 1. \tag{7.27}
\]
# Algorithm 8: Lattice equations expressed in arrays of parameters and gains.

while $t \geq 0$ do

Order Initialization

$$\hat{e}_{0,q+1} = \hat{w}, \quad \hat{r}_{0,q} = \begin{bmatrix} \hat{w} \\ 0 \end{bmatrix}$$ (7.14)

Order Connection

$0 \leq n \leq N - 1$

$$\hat{e}_{n+1:0} = \hat{e}_{n+1:0} = \hat{e}_{n;q+1}$$
$$\hat{r}_{n+1:0} = \hat{r}_{n+1:0} = z^{-1}\hat{r}_{n;q}$$ (7.15)

for $k = 1$ to $q$ do

Forward Block

$1 \leq n \leq N$

$$\hat{r}_{n;k} = \hat{r}_{n;k-1} - \hat{\beta}^t_{n} 1_q k \hat{e}_{n;k-1}$$
$$\hat{e}_{n;k} = (I_q - \hat{\alpha}^t_{n} 1_q k) \hat{e}_{n;k-1}$$ (7.16)

end

for $k = 1$ to $q + 1$ do

Backward Block

$1 \leq n \leq N$

$$\hat{e}_{n;k} = \hat{e}_{n;k-1} - \hat{\beta}^t_{n} 1_{q+1} k \hat{r}_{n;k-1}$$
$$\hat{r}_{n;k} = (I_{q+1} - \hat{\alpha}^t_{n} 1_{q+1}) \hat{r}_{n;k-1}$$ (7.17)

end

end
Algorithm 9: Simplified lattice equations.

\[ \text{while } t \geq 0, \ 1 \leq n \leq N \text{ do} \]

Order Initialization

\[ \tilde{e}_{0; q+1}(t) = \tilde{w}(t), \ \hat{r}_{0; q}(t) = \begin{bmatrix} \tilde{w}(t) \\ 0 \end{bmatrix} \quad (7.18) \]

Order Connection

\[ \begin{align*}
\hat{e}_{n; 0}(t) &= \hat{e}_{n-1; q+1}(t) \\
\hat{r}_{n; 0}(t) &= \hat{r}_{n-1; q}(t-1) \\
\tilde{e}_{n; 0}(t) &= \tilde{e}_{n-1; q+1}(t) \\
\tilde{r}_{n; 0}(t) &= \tilde{r}_{n-1; q}(t-1) 
\end{align*} \quad (7.19) \]

for \( k = 1 \) to \( q \) do

Forward Block

\[ \begin{align*}
\hat{r}_{n; k}(t) &= \hat{r}_{n; 0}(t) - \sum_{l=1}^{k} \hat{\beta}_n^{l \dagger} q \hat{e}_{n; l-1}(t) \\
\hat{e}_{n; k}(t) &= \prod_{j=k}^{1} \left( I_q - \hat{\alpha}_n^{j \dagger} q \right) \hat{e}_{n; 0}(t) 
\end{align*} \quad (7.20) \]

end

for \( k = 1 \) to \( q + 1 \) do

Backward Block

\[ \begin{align*}
\tilde{e}_{n; k}(t) &= \tilde{e}_{n; 0}(t) - \sum_{l=1}^{k} \tilde{\beta}_n^{l \dagger} q + 1 \tilde{r}_{n; l-1} \\
\tilde{r}_{n; k}(t) &= \prod_{j=k}^{1} \left( I_{q+1} - \tilde{\alpha}_n^{j \dagger} q + 1 \right) \tilde{r}_{n; 0}(t) 
\end{align*} \quad (7.21) \]

end

end
The sequence $y(t)$ is set equal to the sequence $\hat{w}$,

$$ y(t) \triangleq \hat{w}(t). \quad (7.28) $$

The error vector $\varepsilon(t)$ is defined to be $\tilde{e}_{N,q+1}(t)$; i.e.,

$$ \varepsilon_n(t) \triangleq \tilde{e}_{n,q+1}(t) = - \sum_{\eta=1}^{n} \hat{Q}_\eta x_\eta(t) + y(t), \quad (7.29) $$

$$ \varepsilon(t) \triangleq \varepsilon_N(t), \quad \varepsilon_0(t) = y(t). $$

The one-step prediction of the $y$ sequence is $\tilde{y}$:

$$ \tilde{y}(t) \triangleq y(t) - \varepsilon_N(t). \quad (7.30) $$

The order initializations are used to write the state vector $x_1$ in the following form:

$$ x_1(t+1) = - \hat{Q}_0 y(t) = \begin{bmatrix} y(t) \\ 0 \end{bmatrix}. \quad (7.31) $$

Rewriting the $\hat{r}$ update equation (7.26) in terms of the state vectors and the sequence $y$ yields the following equation:

$$ x_{n+1}(t+1) = x_n(t) + \hat{Q}_n \sum_{\eta=1}^{n-1} \hat{Q}_\eta x_\eta(t) - \hat{Q}_n y(t), \quad (7.32) $$

where $1 \leq n \leq N - 1$ gives us an iterative equation for state vectors. The prediction error $\varepsilon$ and the one-step prediction $\tilde{y}$ are rewritten:

$$ \varepsilon(t) = - \sum_{\eta=1}^{N} \hat{Q}_\eta x_\eta(t) + y(t), \quad (7.33) $$

$$ \tilde{y}(t) = \sum_{\eta=1}^{N} \tilde{Q}_\eta x_\eta(t). $$

The previous equations for state update and the one-step prediction are now combined to
construct a state-space model; i.e.,

\[
\begin{bmatrix}
  x_1(t+1) \\
  x_2(t+1) \\
  x_3(t+1) \\
  \vdots \\
  x_N(t+1)
\end{bmatrix}
= \begin{bmatrix}
  0 & 0 & \ldots & 0 & 0 \\
  I & 0 & \ldots & 0 & 0 \\
  \hat{Q}_2 \hat{Q}_1 & I & \ldots & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  \hat{Q}_{N-1} \hat{Q}_{N-1} \hat{Q}_2 & \ldots & I & 0
\end{bmatrix}
\begin{bmatrix}
  x_1(t) \\
  x_2(t) \\
  x_3(t) \\
  \vdots \\
  x_N(t)
\end{bmatrix}
+ \begin{bmatrix}
  -\hat{Q}_0 \\
  -\hat{Q}_1 \\
  -\hat{Q}_2 \\
  \vdots \\
  -\hat{Q}_{N-1}
\end{bmatrix}
y(t).
\]

(7.34)

\[
\tilde{y}(t) = \begin{bmatrix}
  \hat{Q}_1 & \hat{Q}_2 & \ldots & \hat{Q}_{N-1} & \hat{Q}_N
\end{bmatrix}
\begin{bmatrix}
  x_1(t) \\
  x_2(t) \\
  x_3(t) \\
  \vdots \\
  x_N(t)
\end{bmatrix}.
\]

The basic steps of this derivation are shown in Fig. 7.2.

---

**Figure 7.1: Constructing lattice-based state-space model from the lattice algorithm.**
7.3 Computation of the State-Space Coefficients

The coefficients $\hat{Q}$ and $\breve{Q}$ can be computed directly from the lattice gains or by batch least squares construction.

7.3.1 Direct Calculation from Lattice Gains

The following forms for the coefficients are explicitly computed using the gains $\hat{\alpha}$, $\hat{\beta}$, $\breve{\alpha}$, and $\breve{\beta}$:

$$\hat{Q}_n = \hat{\beta}'_n \sum_{i=1}^{q} \frac{1}{i} \prod_{j=i-1}^{1} (I_q - \hat{\alpha}'_n 1^j_q)$$

(7.35)

$$\breve{Q}_n = \breve{\beta}'_n \sum_{i=1}^{p} \frac{1}{i} \prod_{j=i-1}^{1} (I_p - \breve{\alpha}'_n 1^j_p)$$

(7.36)

7.3.2 Batch Least Squares Construction

Expanding the $\varepsilon_n(t)$ term in (7.29),

$$\varepsilon_n(t) = -\sum_{\eta=1}^{n} \breve{Q}_{n\eta} x_{\eta}(t) + \breve{w}(t)$$

(7.37)

$$= \varepsilon_{n-1}(t) - \breve{Q}_n x_{\eta}(t)$$

and substituting the terms in the state equations, yields

$$x_{n+1}(t+1) = x_n(t) - \breve{Q}_n \left[ - \sum_{\eta=1}^{n-1} \breve{Q}_{\eta\eta} x_{\eta}(t) + \breve{w}(t) \right]$$

(7.38)

$$+ \varepsilon_{n-1}(t)$$

and

$$x_{n+1}(t+1) = x_n(t) - \breve{Q}_n \varepsilon_{n-1}(t)$$

(7.39)

$$\varepsilon_n(t) = -\breve{Q}_n x_n(t) + \varepsilon_{n-1}(t).$$
The state equations are rewritten by moving the terms around and multiplying both sides of the equation by the error sequence; i.e.,

\[
x_n(t) - x_{n+1}(t+1) = \hat{Q}_n \varepsilon_{n-1}(t)
\]

\[
[x_n(t) - x_{n+1}(t+1)] \varepsilon_{n-1}(t)' = \hat{Q}_n \varepsilon_{n-1}(t) \varepsilon_{n-1}(t)'
\]

\[
\sum_{t=1} \left[ x_n(t) - x_{n+1}(t+1) \right] \varepsilon_{n-1}(t)' = \hat{Q}_n \sum_{t=1} \varepsilon_{n-1}(t) \varepsilon_{n-1}(t)'.
\]

Similarly, the error equation is rewritten:

\[
\varepsilon_{n-1}(t) - \varepsilon_n(t) = \check{Q}_n x_n(t)
\]

\[
[\varepsilon_{n-1}(t) - \varepsilon_n(t)] x_n(t)' = \check{Q}_n x_n(t) x_n(t)'
\]

\[
\sum_{t=1} \left[ \varepsilon_{n-1}(t) - \varepsilon_n(t) \right] x_n(t)' = \check{Q}_n \sum_{t=1} x_n(t) x_n(t)'.
\]

Using the batch least-squares construction, the coefficients \( \hat{Q} \) and \( \check{Q} \) can be computed directly from the state and error vectors; i.e.,

\[
\hat{Q}_n = \left\{ \sum_{t=1} \left[ x_n(t) - x_{n+1}(t+1) \right] \varepsilon_{n-1}(t)' \right\} \left\{ \sum_{t=1} \varepsilon_{n-1}(t) \varepsilon_{n-1}(t)' \right\}^{-1}
\]

\[
\check{Q}_n = \left\{ \sum_{t=1} \left[ \varepsilon_{n-1}(t) - \varepsilon_n(t) \right] x_n(t)' \right\} \left\{ \sum_{t=1} x_n(t) x_n(t)' \right\}^{-1}
\]
7.4 An Additional Channel for the Input Sequence

The input sequence \( \mathcal{Y} \) is constructed by augmenting a single input \( u \) with a single output \( y \),

\[
\mathcal{Y} = \begin{bmatrix} y \\ u \end{bmatrix}.
\]  

(7.44)

The state-space prediction model with input \( \mathcal{Y} \) and output \( \hat{y}(t) \) is

\[
X(t+1) = AX(t) + B\mathcal{Y}(t)
\]

\[
\hat{y}(t) = CX(t).
\]  

(7.45)

The single-input, single-output (SISO) model with input \( u \) and output \( y \) is constructed from the lattice state-space and AR models.

7.4.1 Lattice ARX

The lattice AR equation for the coefficients,

\[
\hat{A}_k = \begin{bmatrix} \hat{A}^{yy}_k & \hat{A}^{yu}_k \\ \hat{A}^{uy}_k & \hat{A}^{uu}_k \end{bmatrix},
\]

(7.46)

and the sequence \( \mathcal{Y} \) is written as follows:

\[
\hat{y}(t) = \sum_{k=1}^{N} \hat{A}_k \mathcal{Y}(t-k).
\]  

(7.47)

The ARX model is:

\[
y(t) - \sum_{k=1}^{N} \hat{A}^{yy}_k y(t-k) = \sum_{k=1}^{N} \hat{A}^{yu}_k u(t-k),
\]

(7.48)

A state-space model can be derived from expanding the ARX model above with the known
7.4.2 Lattice State-Space

An equivalent SISO system from the lattice state-space model is constructed here. First the state-space model is made observable; i.e.,

\[
\begin{bmatrix}
X^{no} \\
\vdots \\
X^{o}
\end{bmatrix}
(t+1) =
\begin{bmatrix}
A^{no} & : & A^{12} \\
\vdots & : & \vdots \\
0 & : & A^{o}
\end{bmatrix}
\begin{bmatrix}
X^{no} \\
\vdots \\
X^{o}
\end{bmatrix}
(t) +
\begin{bmatrix}
B^{no} \\
\vdots \\
B^{o}
\end{bmatrix}
\begin{bmatrix}
y \\
\vdots \\
u
\end{bmatrix}
(t)
\]

\[y(t) = \begin{bmatrix} 0 & : & C^{o}_y \end{bmatrix} \begin{bmatrix} X^{no} \\
\vdots \\
X^{o} \end{bmatrix} (t) + \epsilon_y(t)^{0}\]

The state-space model relating the input \(u\) and output \(y\) with observable states is

\[
X^{o}(t+1) = (A^{o} + B^{o}_y C^{o}_y) X^{o}(t) + B^{o}_u u(t),
\]

\[y(t) = C^{o}_y X^{o}(t).\]
CHAPTER 8

Identification of a Test System

8.1 Construction of the Test System

To compare the lattice-filter based identification methods with two other prominent system identification methods, a single-input, single-output (SISO) system with 30 states was constructed as follows. Five digital bandpass digital filters with six states each were superimposed. Each bandpass filter was designed using MATLAB `butter` command:

\[
[b_i, a_i] = \text{butter}(\text{order}_i, [\omega_{1i} \omega_{2i}]).
\] (8.1)

Transfer functions for individual filters were constructed by

\[
H_i = \text{tf}(b_i, a_i, T_s),
\]

\[
H = \sum_{i=1}^{5} H_i.
\] (8.2)

The filter properties are tabulated in Table 8.1 and the frequency response of the individual transfer functions are shown in Fig. 8.1. There are a total of five peaks in the bode magnitude plot corresponding to the bandpass filters.

Table 8.1: Filter properties.

<table>
<thead>
<tr>
<th>filter</th>
<th>order</th>
<th>bandpass</th>
</tr>
</thead>
<tbody>
<tr>
<td>(H_1)</td>
<td>3</td>
<td>[.3 .34 ]</td>
</tr>
<tr>
<td>(H_2)</td>
<td>3</td>
<td>[.4 .44 ]</td>
</tr>
<tr>
<td>(H_3)</td>
<td>3</td>
<td>[.5 .54 ]</td>
</tr>
<tr>
<td>(H_4)</td>
<td>3</td>
<td>[.6 .64 ]</td>
</tr>
<tr>
<td>(H_5)</td>
<td>3</td>
<td>[.7 .74 ]</td>
</tr>
</tbody>
</table>

The Hankel singular values of this system, computed using MATLAB `balreal` command, are plotted in Fig. 8.2. It is interesting to note that there are three step-like range of values. These singular values slowly approach zero as \(N\) approaches the model order, 30.
Figure 8.1: Bode magnitude plot for the transfer function of the butter filters and their sum.

An input sequence $u$ was generated by using a Gaussian random number generator and was passed through the test system to generate the output $y$ sequence; this process is illustrated in Fig. 8.3.
Figure 8.2: Hankel singular values of the test system.

Figure 8.3: An input-output diagram for the true system.
8.2 Comparison Models

The following four methods and models were used to identify the system in Section 8.1:

- Lattice State-Space
- Lattice ARX
- MATLAB N4SID
- MATLAB ARX

In each of the comparisons presented subsequently in this chapter, all four methods used the same input and output data sequences as depicted in Fig. 8.4.

![Figure 8.4: The comparison models for system identification of the test system.](image)

The lattice state-space model and ARX model were constructed for a given model order $N$ as explained in Section 7.4. Two other models used for comparison throughout this chapter were identified by `n4sid` and `arx` commands in MATLAB’s System Identification Toolbox.

The N4SID\(^1\) algorithm uses weighting schemes (N4Weight) for singular value decomposition. Canonical correlation analysis (CVA) \(^{26}\) and multivariable output error state space

\(^1\)N4SID stands for numerical algorithms for subspace state space system identification.
(MOESP) [27] are two of such methods. The forward- and backward-prediction horizons can also be set by the horizon parameter (N4Horizon) that requires a three-element vector \( [r \ s_y \ s_u] \) containing the values for the maximum forward prediction horizon denoted by \( r \), the number of past outputs, \( s_y \), and the number of past inputs, \( s(u) \) [28]. When the auto horizon option is chosen, an Akaike Information Criterion (AIC) is used for the selection of past inputs and outputs [21]².

The \texttt{arx} function takes a vector of polynomial orders, \( [n_a \ n_b \ n_k] \), and returns an ARX structure polynomial model using least squares. The orders of the ARX model corresponding to our test system are \( n_a = N \) and \( n_b = N \), and \( n_k = 1 \) is the delay.

The relative identification error was defined to be the norm of the difference between the true and the identified system by the norm of the true system; i.e.,

\[
\text{Relative Identification Error} \triangleq \frac{||sys_{identified} - sys_{true}||}{||sys_{true}||} \tag{8.3}
\]

This measure of system identification performance is similar to normalized root mean square cost function.

For model orders 15 through 45, the relative identification errors for the four methods are shown in Fig. 8.5. The data length for this identification procedure was 30000 steps. It is interesting to note that the relative identification error for all methods decreases dramatically as the model order approaches the true order, \( N = 30 \). This property of the test system was also observed in Fig. 8.2.

²For the results presented in this chapter, \( [60 \ 120 \ 120] \) horizon and MOESP weighting were chosen.
Figure 8.5: Relative identification error versus the model order.
8.3 Sensitivity to Noise

8.3.1 Process and Measurement Noise

In order to observe the sensitivity of the identified models to the presence of noise, a noise sequence was added to the output as illustrated in the block diagram in Fig. 8.6.

![Block diagram of process and measurement noise](image)

Figure 8.6: A diagram describing the superposition of the output and a colored noise sequence.

Sequences $v_0$ and $v_1$ were generated by a Gaussian random number generator

$$v_0 \sim \mathcal{N}(\mu = 0, \sigma_0^2), \quad v_1 = \text{sys}_{\text{true}} \cdot v_0,$$

$$v_2 \sim \mathcal{N}(\mu = 0, \sigma_2^2), \quad v = v_1 + v_2,$$

where, the standard deviations $\sigma_0$ and $\sigma_2$ were chosen such that $||v|| = ||y||$ holds. The resulting colored noise sequence represents a weighted sum of process and measurement noise. A noise sequence proportional to $v$ was added to the output $y$; i.e.,

$$y_{\text{noisy}} = y + \alpha v,$$

where $\alpha$ is the noise ratio.
### 8.3.2 Comparison of Identification Errors

The relative errors were computed for a range of noise ratios for the aforementioned identification methods with model order $N = 100$. The data length chosen for these results is 30,000. After the identification process, the models were balanced by `balreal` and reduced to 30 states by `modred`. These steps are summarized in Fig. 8.7. The relative identification error values are shown in Fig. 8.8 on a logarithmic scale.

![Figure 8.7: A summary of identification steps in the presence of output noise.](image)

The same data as in Fig. 8.8 normalized by the corresponding noise ratios are shown on linear scale in Fig. 8.9 when zoomed in around higher noise levels to demonstrate more clearly the sensitivity of these models in presence of unmodeled disturbance. The normalized relative identification is defined as

$$\text{Normalized Relative Identification Error} \triangleq \frac{||sys_{\text{identified}} - sys_{\text{true}}||}{\alpha ||sys_{\text{true}}||}, \quad (8.6)$$

where $\alpha$ denotes the noise ratio.
Figure 8.8: Relative identification error plotted versus the noise ratios, on a logarithmic scale.
Figure 8.9: Normalized relative identification error, on a linear scale.
8.3.3 Frequency Response of the Lattice-Filter State-Space Model

The frequency responses of the true model \(8.2\) and the lattice state-space identified model for four levels of noise ratio, \(\alpha\), are shown in Fig. 8.10. The model order, \(N = 100\) was chosen for the identification process and then the state-space model produced by the lattice-filter state-space identification was balanced by MATLAB \texttt{balreal} command and reduced to 30 states.

![Frequency Response Diagrams](image)

Figure 8.10: Frequency response for the true and the lattice state-space identified systems for four noise ratios, \(\alpha\).
8.4 Identification Error and Length of the Data Sequence

The input sequence $u$ was generated in different lengths of time $T$ and was passed through the true system to generate the output $y$. For each value of $T$, the system was identified via four identification methods with model order $N = 100$, and then the balanced realizations were obtained and subsequently reduced to 30 states. The normalized relative identification error is plotted versus the length of the sequence, $T$, in Fig. 8.11 for four noise ratios. A sequence proportional to the inverse of square root of $T$ is plotted alongside the other four curves. The normalized relative identification error decreases with the length of the data sequence as $1/\sqrt{T}$. Related studies on time complexity are presented in [29]. Appendix D explains the parameter convergence as a function of data points for an autoregressive model.

Figure 8.11: Relative identification error versus the number of data points.
8.5 Effects of Single Precision Computation, Model Order, and Balancing

The Fixed-Point simulation in MATLAB was used for observing the behavior of our identification models under single-precision fixed-point arithmetics.

The relative identification errors are defined similar to those in the previous sections; i.e.,

\[
\text{Relative Identification Error} \triangleq \frac{||y_{\text{fixed-point}} - y||}{||y||}.
\]  

(8.7)

The identified models were obtained for a range of model orders, \(N\), and 30,000 data points. For each case, a simulation was conducted for \(t_{\text{simulation}} = 10,000\) as demonstrated in Fig. 8.12. This procedure was conducted once for the models in their original coordinates and then for the balanced realizations [30] as illustrated in Fig. 8.13. The relative identification errors for both cases are shown in Fig. 8.14.

Figure 8.12: Fixed-Point simulation of the identified models.
Figure 8.13: Fixed-Point simulation for the balanced realization of the identified models.

Figure 8.14: Relative identification error in precision versus order. Left: identified models in their original coordinates. Right: balanced realizations of the identified models.
8.6 Computational Complexity

Figures 8.16 and 8.17 present a comparison of processing times for the identification methods. Related research on N4SID algorithm and its complexity is demonstrated in [27,31,36]. The values for processing time were averaged over 50 runs for each identification method. This procedure is demonstrated in Fig. 8.15.

The average run-time for lattice-filter based state space, lattice ARX, and MATLAB ARX are plotted on a linear scale in Fig. 8.18 to demonstrate more clearly the linear relation between the computation time and the model order.

Figure 8.15: A schematic of runtime analysis.

---

3MATLAB version R2013b was used for these computations.
Figure 8.16: Identification processing time versus the model order, on a logarithmic scale.
Figure 8.17: Identification processing time versus the model order, on a linear scale.
Figure 8.18: Identification processing time versus the model order for three models, on a linear scale.
CHAPTER 9

Conclusions

The first part of the dissertation in Chapters 2-4 describes a state-space disturbance model and associated prediction filter for aero-optical wavefronts. The model is computed by subspace system identification from a sequence of wavefronts measured in the University of Notre Dame’s Airborne Aero-Optics Laboratory. For identification, the wavefront sequence is projected onto a set of Karhunen-Loève modes, and a subspace system identification algorithm generates the state-space model from the modal sequences. The identified state-space model is rearranged to obtain a minimum-variance prediction filter. The fact that the prediction error is very nearly white indicates that the statistics associated with the state-space model closely approximate the statistics of the original aero-optical data.

Two other kinds of results in Chapters 5 and 6 indicate that the identified state-space model captures the important characteristics of the aero-optical wavefront data. We have shown how to construct estimates of the local flow velocity directly from the matrices in the state-space model without returning to the original data. These local velocity estimates agree reasonably well with local velocity estimates computed directly from the wavefront data. The mean velocity estimates, which are the averages of 17 local velocity estimates, agree very well; the mean velocities computed directly from the state-space model differ by less than 2% from the mean velocities computed directly from the data.

It was also shown how to use the frequency response of the multi-input-multi-output state-space model to compute a theoretical power spectrum for the time series of phase values at any pixel in the sequence of wavefront images. Plots comparing the theoretical pixel frequency responses to the power spectra of the corresponding pixel time series from the original data show good agreement except at low frequencies, where the accuracy of both
the power spectra of the data and the frequency response of the identified model are limited by the amount of data.

The main contribution of Chapters 7 and 8 has been a methodology for maintaining the desirable properties of the lattice filter using a state-space representation. An important property of this new state-space model is its numerical efficiency for higher system orders. This lattice-filter based state-space model was compared to well known system identification algorithms such as N4SID and least-squares estimation of ARX models. The sensitivity of these models with respect to unmodeled colored noise was investigated. The results illustrated that the new model compared favorably with the other identified models. It was also confirmed that the identification error for the corresponding identified models decreased with increasing the input-output data length; that is, the relative identification error was observed to be inversely proportional to the square root of the data length. The computational complexities of the MATLAB N4SID and the new identification method were compared. The results demonstrated the computational speed of lattice-filter based state-space system identification.
APPENDIX A

Autoregressive Exogenous Model to State-Space Model

In order to convert the following autoregressive exogenous model,

\[ y(t) = -\sum_{k=1}^{N} \alpha_k y(t-k) + \sum_{k=1}^{N} \beta_k u(t-k) + \beta_0 u(t), \]  

(A.1)

to a state-space model,

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

(A.2)

the term \( y(t) \) is rewritten using the matrices \( C \) and \( D \):

\[
Cx(t) = \sum_{k=1}^{N} -\alpha_k Cx(t-k) + (\beta_k - \alpha_k \beta_0) u(t-k), \quad D = \beta_0.
\]

(A.3)

The following terms are defined:

\[
\begin{align*}
    \mathcal{Y} &\triangleq y(t) - \beta_0 u(t), \\
    \gamma_k &\triangleq \beta_k - \alpha_k \beta_0, \\
    \theta_k &\triangleq -\alpha_k.
\end{align*}
\]

(A.4)

The output equations is rewritten by using the newly defined terms; i.e.,

\[
\mathcal{Y}(t) = \sum_{k=1}^{N} \theta_k \mathcal{Y}(t-k) + \sum_{k=1}^{N} \gamma_k u(t-k).
\]

(A.5)

The term \( x_n \) is defined as

\[
x_n(t) \triangleq \sum_{k=n}^{N} \theta_k \mathcal{Y}(t-k-1+n) + \sum_{k=n}^{N} \gamma_k u(t-k-1+n),
\]

(A.6)
and is expanded to

\[
x_n(t) = \theta_n y(t-1) + \gamma_n u(t-1)
\]

\[
+ \sum_{k=n+1}^{N} \theta_k y((t-1) - k - 1 + (n+1)) + \sum_{k=n+1}^{N} \gamma_k u((t-1) - k - 1 + (n+1)) \tag{A.7}
\]

\[
= -\alpha_k x_1(t-1) + x_{n+1}(t-1) + (\beta_k - \alpha_k \beta_0) u(t-1).
\]

Collecting the corresponding coefficients corresponding to \(x_i\), for \(i:1,2,\cdots,N\), yields the following desired state matrices:

\[
A = \begin{bmatrix}
-\alpha_1 & I & \cdots & 0 \\
-\alpha_2 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
-\alpha_N & 0 & \cdots & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
\beta_1 - \alpha_1 \beta_0 \\
\beta_2 - \alpha_2 \beta_0 \\
\vdots \\
\beta_N - \alpha_N \beta_0
\end{bmatrix}, \quad C = \begin{bmatrix}
I & 0 & \cdots & 0
\end{bmatrix}, \quad D = [\beta_0]. \tag{A.8}
\]
APPENDIX B

Order Reduction

For the following state-space system with unobservable states,

\[ x(t+1) = Ax(t) + Bu(t), \]
\[ y(t) = Cx(t) + Du(t), \]  \hspace{1cm} (B.1)

the observability matrix has rank \( r \).

The matrix \( O \) has a similar construction to the observability matrix; i.e.,

\[ O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} \]  \hspace{1cm} (B.2)

The QR factorization for the transpose of \( O \) is written as follows:

\[ \begin{bmatrix} Q \\ R \end{bmatrix} = \text{qr}(\text{flipud}(O)',0), \]  \hspace{1cm} (B.3)

where \( T = Q' \). Using \( T \) as a similarity matrix, the new state matrices are

\[ \bar{A} = TAT', \hspace{0.5cm} \bar{B} = TB, \hspace{0.5cm} \bar{C} = CT'. \]  \hspace{1cm} (B.4)

These new matrices have the following matrix forms:

\[ \bar{A} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ * & 0 & 0 & \cdots & 0 & 0 \\ * & * & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & * & \cdots & 0 & 0 \\ * & * & * & \cdots & * & 0 \end{bmatrix}, \hspace{0.5cm} \bar{B} = \begin{bmatrix} * \\ * \\ * \\ \vdots \\ * \\ * \end{bmatrix}, \hspace{0.5cm} \bar{C} = \begin{bmatrix} * & * & \cdots & * & * \end{bmatrix}. \]  \hspace{1cm} (B.5)

The dimensions of state-space matrices are tabulated in Table B.
Table B.1: State-space dimensions.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$n \times 1$</td>
</tr>
<tr>
<td>$A$</td>
<td>$n \times n$</td>
</tr>
<tr>
<td>$B$</td>
<td>$n \times 2$</td>
</tr>
<tr>
<td>$C$</td>
<td>$1 \times n$</td>
</tr>
<tr>
<td>$D$</td>
<td>$1 \times 2$</td>
</tr>
<tr>
<td>$\mathcal{O}$</td>
<td>$r \times n$</td>
</tr>
<tr>
<td>$\tilde{x}$</td>
<td>$r \times 1$</td>
</tr>
<tr>
<td>$\tilde{A}$</td>
<td>$r \times r$</td>
</tr>
<tr>
<td>$\tilde{B}$</td>
<td>$r \times 2$</td>
</tr>
<tr>
<td>$\tilde{C}$</td>
<td>$1 \times r$</td>
</tr>
<tr>
<td>$\tilde{D}$</td>
<td>$1 \times 2$</td>
</tr>
</tbody>
</table>
APPENDIX C

N4SID: Weighting Schemes and Prediction Horizons

The test system in Chapter 8 is identified by MATLAB n4sid command\(^\text{[37]}\). Several combinations of weighting schemes and prediction horizons are used for obtaining the prediction models. The relative identification errors for various noise ratios are shown in Fig. C.1. These values are computed for several noise ratios, explained in Section 8.3 and are tabulated in Table C.1.

Table C.1: Relative identification error for selected N4SID options.

<table>
<thead>
<tr>
<th>Noise Ratio</th>
<th>Horizon</th>
<th>Weight</th>
<th>relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto</td>
<td>Auto</td>
<td>Auto</td>
<td>0.4452298</td>
</tr>
<tr>
<td>10</td>
<td>[ 60 60 60 ]</td>
<td>CVA</td>
<td>0.4313302</td>
</tr>
<tr>
<td>[ 60 120 120 ]</td>
<td>MOESP</td>
<td>Auto</td>
<td>0.0753642</td>
</tr>
<tr>
<td>1</td>
<td>[ 60 60 60 ]</td>
<td>CVA</td>
<td>0.0720348</td>
</tr>
<tr>
<td>[ 60 120 120 ]</td>
<td>MOESP</td>
<td>Auto</td>
<td>0.0375498</td>
</tr>
<tr>
<td>10(^{-1})</td>
<td>[ 60 60 60 ]</td>
<td>CVA</td>
<td>0.0066607</td>
</tr>
<tr>
<td>[ 60 120 120 ]</td>
<td>MOESP</td>
<td>Auto</td>
<td>0.0064203</td>
</tr>
<tr>
<td>10(^{-2})</td>
<td>[ 60 60 60 ]</td>
<td>CVA</td>
<td>0.0006601</td>
</tr>
<tr>
<td>[ 60 120 120 ]</td>
<td>MOESP</td>
<td>Auto</td>
<td>0.0006556</td>
</tr>
</tbody>
</table>
Figure C.1: N4SID identification using different weighting methods and prediction horizons.
APPENDIX D

Data Points and Convergence

Central Limit Theorem

Let \{X_1, \cdots, X_T\} be a random sample of size \(T\) for a sequence of independent and identically distributed random variables with mean \(\mu = E[X_t]\) and covariance \(Q = E[(X_t - \mu)(X_t - \mu)'] = \sigma^2\). The following sample average,

\[
S_T := \frac{1}{T} \sum_{t=1}^{T} (X_t - \mu),
\]

converges to the Gaussian distribution as \(T\) approaches infinity; i.e., \(\sqrt{T}(S_T - \mu)\) approximates the normal distribution with mean 0 and variance \(\sigma^2\).

Distribution of Estimates

Consider the following model,

\[
y(t) = -\sum_{k=1}^{N} \alpha_k y(t-k) + \sum_{k=1}^{N} \beta_k u(t-k) + e(t)
\]

\[
= \begin{bmatrix} -y(t-1) & \cdots & -y(t-N) & u(t-1) & \cdots & u(t-N) \end{bmatrix} \phi(t)',
\]

\[
\begin{bmatrix}
\alpha_1 \\
\vdots \\
\alpha_N \\
\beta_1 \\
\vdots \\
\beta_N \\
\theta
\end{bmatrix},
\]

(D.2)
where the $\theta$ is a vector of unknown model parameters. This vector is identified based on past inputs $u(1), \cdots, u(T)$ and outputs $y(1), \ldots, y(T)$. The predictor is $\hat{y}(t|\theta) = \phi'(t)\theta$ and the prediction error is $\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta)$. The mean squared error to be minimized is:

$$\mathcal{V}_T(\theta) = \frac{1}{T} \sum_{t=1}^{T} \varepsilon(t)^2$$

$$= \frac{1}{T} \sum_{t=1}^{T} [y(t) - \phi'(t)\theta]^2 \quad \text{(D.3)}$$

$$\hat{\theta} = \text{argmin}_\theta \mathcal{V}_T(\theta)$$

$$\rightarrow \frac{d\mathcal{V}_T(\theta)}{d\theta} = 0$$

$$\frac{1}{T} \sum_{t=1}^{T} [y(t) - \phi'(t)\theta](-\phi(t)) = 0 \quad \text{(D.4)}$$

$$\frac{1}{T} \left[ \sum_{t=1}^{T} \phi(t)\phi'(t) \right] \theta = \frac{1}{T} \sum_{t=1}^{T} y(t)\phi(t).$$

The model parameters estimate is:

$$\hat{\theta}_T = \mathcal{R}_T^{-1} f_T \quad \text{(D.5)}$$

Applying the Central Limit Theorem, the distribution of estimate $\sqrt{T}(\hat{\theta}_T - \theta_0)$ converges to the normal distribution given by

$$\sqrt{T}(\hat{\theta}_T - \theta_{\text{true}}) \sim \mathcal{N}(0, Q) \text{ as } T \rightarrow \infty. \quad \text{(D.6)}$$

Therefore, the estimate converges to the true estimate at a rate proportional to $\frac{1}{\sqrt{T}}$.
References


