

2022-11-28

Constrained LASSO for Sparse Identification of Nonlinear Dynamical Systems (SINDy)

Mackie, Alexander Douglas

Mackie, A. D. (2022). Constrained LASSO for sparse identification of nonlinear dynamical systems (SINDy) (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>.
<http://hdl.handle.net/1880/115562>

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

Constrained LASSO for Sparse Identification of Nonlinear Dynamical
Systems (SINDy)

by

Alexander Douglas Mackie

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN MATHEMATICS AND STATISTICS

CALGARY, ALBERTA

NOVEMBER, 2022

©Alexander Douglas Mackie 2022

Abstract

Alexander Douglas Mackie

*Constrained LASSO for Sparse Identification of Nonlinear
Dynamical Systems (SINDy)*

Our work explores and expands on (Brunton, Proctor, and Kutz, 2016) with regard to bluff body vortex shedding. We have adapted the SINDy method by applying a transformation of the data to reduce the number of dimensions under investigation. We also applied Galerkin constraints associated with our transformation in order to further reduce the variables being considered when model building. Finally, by using LASSO as our method of solving the SINDy problem rather than sequential threshold least squares, we have created a much more efficient approach that attempts to discover the generating equations of the non-linear dynamical system associated with vortex shedding in the wake of a flat plate. Our approach was tested by modeling vortex shedding in the wake of a cylindrical bluff body with a low Reynolds number, and was able to extract expected elements of the governing equations. With this success, we established several models for vortex shedding in the wake of a flat plate bluff body (for both open and closed ends) at a high Reynolds number. Under our transformed data, we obtained 2, 3 and 5 mode models that may shed some light into the dynamics of the system.

Acknowledgements

While there are so many people I would love to thank here, I will limit it to those who played a crucial role with regards to this particular document. Whether through direct guidance, helping me learn how to approach material more thoroughly or simply reminding me to take a break every once in a while, I couldn't have done it without you. I would like to acknowledge and thank:

Dr. Deniz Sezer

Mom and Dad

Dr. Robert Martinuzzi

Allegra Mackie

Odin Chlebek

Sayler Reins

Yang Yang

Glenna Schowalter

Dr. Anatoliy Swishchuk

Dr. Gemai Chen

Dr. Jingjing Wu

Dr. Xuewen Lu

The University of Calgary

GUMS

Whitney Lawson

Chris Smallwood

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Background	1
2 LASSO	5
2.1 Introduction	5
2.2 Derivation	5
2.3 Orthonormal Case	7
2.3.1 Subset Selection	8
2.3.2 Ridge Regression	9
2.3.3 LASSO	9
2.4 Cyclical Coordinate Descent and Path Seeking Algorithms	12
2.4.1 glmnet	13
3 Sparse Identification of Non-Linear Dynamical System (SINDy)	16
3.1 Introduction	16
3.1.1 The Process	16
3.2 Adapting the SINDy Method	18
4 Theory	20
4.1 Theory	20
4.1.1 Notation	20
4.2 Model Comparisons	22
4.2.1 Model 1	22
4.2.2 Model 2	24

4.2.3	Constraints	25
4.3	Finding β from $\tilde{\beta}$	27
5	Design	31
5.1	Overview	31
5.2	Data Introduction	31
5.2.1	Cylinder Data	31
5.2.2	Flat Plate Data	32
5.3	MATLAB code	33
5.3.1	Master Script	34
5.3.2	Uploading Data	34
5.3.3	Data Smoothing	34
5.3.4	Derivatives	35
5.3.5	Θ and $\dot{\theta}$	35
5.3.6	Scaling	35
5.3.7	Transforming the Energy Constraints	35
5.3.8	Calculating Regression Coefficients	36
5.3.9	Obtaining Models	36
6	Results	37
6.1	Cylinder Data	37
6.2	Closed Ends Data	38
6.2.1	2 mode analysis	38
6.2.2	3 mode analysis	40
6.2.3	5 mode analysis	42
6.3	Open Ends Data	44
6.3.1	2 mode analysis	44
6.3.2	3 mode analysis	46
6.3.3	5 mode analysis	48
7	Conclusions	50
7.1	Conclusions	50

A	Matrices and Vectors	53
A.1	Galerkin Constraint Matrix	53
A.2	Constraint Matrix \tilde{E}	55
A.3	Theta Vector	55
A.4	Vectors $x_{20 \times 1}^{Id}$ and $y_{20 \times 1}^{Id}$	56
A.4.1	Construction of $x_{20 \times 1}^{Id}$	56
A.4.2	Construction of $y_{20 \times 1}^{Id}$	56
A.5	Submatrices $\tilde{A}_{20 \times 20}^x$ and $\tilde{A}_{20 \times 20}^y$	57
A.5.1	$\tilde{A}_{20 \times 20}^x$	57
A.5.2	$\tilde{A}_{20 \times 20}^y$	58
B	Calculations	59
B.1	Proposition Calculations	60
B.1.1	Calculations for β^x	60
B.1.2	Calculations for $\tilde{\beta}^y$	61
C	Variation Table	62
D	Models	63
D.1	Closed Ends Data	63
D.1.1	2 Mode Analysis	63
D.1.2	3 Mode Analysis	64
D.1.3	5 Mode Analysis	65
D.2	Open Ends Data	68
D.2.1	2 Mode Analysis	68
D.2.2	3 Mode Analysis	69
D.2.3	5 Mode Analysis	70
E	Spectrograms	74
E.1	Closed Ends Data	74
E.2	Open Ends Data	77
	Bibliography	81

List of Figures

1.1	A depiction of the POD decomposition observed in the cylinder example described in (Brunton, Proctor, and Kutz, 2016). We see the effects of the two harmonic oscillators on the vortex shedding and the mean flow of the model. The shift mode connects the mean flow with the unstable fixed point. Image is from (Brunton, Proctor, and Kutz, 2016)	2
2.1	Geometric Representation of Optimization of $\hat{\beta}$ with l_1 (left) and l_2 (right) penalty parameters. On the left we see that tangency is possible at zero due to the 'sharp corners' of the l_1 penalty parameter. It is highly unlikely to obtain a tangent point at zero in the l_2 case on the right. The image is from (Bühlmann, 2011)	11
5.1	Spectral decomposition of the harmonic oscillating pair. As we can see, there is a spike in the energy (peak in the graph) at 10^{-1} , telling us the modes have similar active frequencies.	32
5.2	A visualization of the system under investigation (PIV := Particle Image Velocimetry FoV:=Field of View). The image is taken from (Braun, Agrey, and Martinuzzi, 2020)	33
6.1	Mean square error vs $\log(\lambda)$ plot for Closed Ends data (2-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error. The selected model has a $\log(\lambda)$ value of -3.9417 and is on the red dotted line.	38
6.2	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{A}/A (left) and \dot{z} (right) for the Closed Ends data.	39

6.3	Mean square error vs $\log(\lambda)$ plot for Closed Ends data (3-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). The selected model has a $\log(\lambda)$ value of -4.4389 and is on the red dotted line.	40
6.4	Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the Closed Ends data.	41
6.5	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the Closed Ends data. .	41
6.6	Mean square error vs $\log(\lambda)$ plot for Closed Ends data (5-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). The selected model has a $\log(\lambda)$ value of -5.5669 and is on the red dotted line.	42
6.7	Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the Closed Ends data.	43
6.8	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the Closed Ends data. .	43
6.9	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{u} (left) and \dot{v} (right) for the Closed Ends data. .	43
6.10	Mean square error vs $\log(\lambda)$ plot for Open Ends data (2-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). We notice that the 'elbow' occurs between the read a blue lines and so we choose the point were the mean square error levels off. The $\log(\lambda)$ value is -5.2058.	44
6.11	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{A}/A (left) and \dot{z} (right) for the Open Ends data.	44

6.12	Mean square error vs $\log(\lambda)$ plot for Open Ends data (3-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). The model select is on the dotted red line and has a $\log(\lambda)$ value of -5.2274.	46
6.13	Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the open ends data.	47
6.14	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the open ends data. . .	47
6.15	Mean square error vs $\log(\lambda)$ plot for Open Ends data (3-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). Our model of choice is on the red line and has an associated $\log(\lambda)$ of -5.8893	48
6.16	Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the Open Ends data.	48
6.17	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the Open Ends data. . .	49
6.18	Observed values (blue line) compared against the fitted values from the model (red line) of \dot{u} (left) and \dot{v} (right) for the Open Ends data. . .	49
C.1	Table of Variation accounted for under different geometries and models. Calculation to obtain values is $1 - (\text{residual variation}) / (\text{observed variation})$	62
E.1	Spectrogram of x , one of the modes that makes up our first harmonic oscillating pair	74
E.2	Spectrogram of y , one of the modes that makes up our first harmonic oscillating pair	75
E.3	Spectrogram of z , also referred to as our shift mode	75
E.4	Spectrogram of w , also referred to as our flapping mode	76
E.5	Spectrogram of u , one of the modes that makes up our second harmonic oscillating pair	76

E.6 Spectrogram of v , one of the modes that makes up our second harmonic oscillating pair	77
E.7 Spectrogram of x , one of the modes that makes up our first harmonic oscillating pair	77
E.8 Spectrogram of y , one of the modes that makes up our first harmonic oscillating pair	78
E.9 Spectrogram of z , also referred to as our shift mode	78
E.10 Spectrogram of w , also referred to as our flapping mode	79
E.11 Spectrogram of u , one of the modes that makes up our second harmonic oscillating pair	79
E.12 Spectrogram of v , one of the modes that makes up our second harmonic oscillating pair	80

List of Abbreviations

KKT	K arush- K uhn T ucker
LASSO	L east A bsolute S hrinkage and S election O perator
LARS	L east A ngle R egression (S implies applications to LASSO)
OLS	O rdinary L east S quares
POD	P roper O rthonormal D ecomposition
SAFE	S Afe F eature E limination
SINDy	S pars I dentification of N on- L inear D ynamical S ystems

List of Symbols

\tilde{A}	A transformation matrix
E	A constraint matrix
P_j	A polynomial of order j
\mathbf{X}^{P_j}	A symbolic vector of all possible permutations of candidate variables with order j
u	POD of one of two modes in the second harmonic oscillating pair
v	POD of one of two modes in the second harmonic oscillating pair
w	POD of the Flapping Mode
x	POD of one of two modes in the first harmonic oscillating pair
\mathbf{X}	A design matrix
$\dot{\mathbf{X}}$	The derivative of the design matrix with respect to time.
y	POD of one of two modes in the first harmonic oscillating pair
z	POD of the Shift Mode
β^m	A vector of regressor coefficients associated with mode m .
β	An array of regressor coefficients
$\hat{\beta}$	An estimate of β
$\beta(:)$	A column vector representing a stacking of each β^m in β
λ	A penalty parameter associated with penalized regression
ψ	An objective function representing a optimization problem
Θ	A matrix of structural parameters
Ξ	An array of coefficients

Chapter 1

Introduction

1.1 Background

The field of fluid dynamics is complex as it requires a deep understanding of the Navier-Stokes equations to establish high-order models of the fluid systems under investigation. While these partial differential equations are able to provide accurate models of a system, it has been discovered that low-order approximations can be created through the use of POD-Galerkin projection. By extracting the principle components of the data through proper orthonormal decomposition, we can recreate an approximation of the system with fewer variables. In (NOACK et al., 2003) they illustrate this process using the example of fluid flow in the wake of a cylindrical bluff body. At a low Reynolds number, they were able to establish a low-order approximation of the system using only three variables: a pair of harmonic oscillators and a shift mode. These equations are

$$\begin{aligned}\dot{u} &= \mu u - v - uw \\ \dot{v} &= \mu v + u - vw \\ \dot{w} &= -w + u^2 + v^2\end{aligned}\tag{1.1}$$

where u and v are our harmonic oscillators, z is the shift mode and μ is an arbitrary constant. The paraboloid formed by these equations represents the steady state of the system, where the harmonic oscillating pair captures the majority of the energy, and the shift mode captures perturbations in the system, keeping the unstable elements close to the limit cycle formed by the two harmonic oscillators. Even though

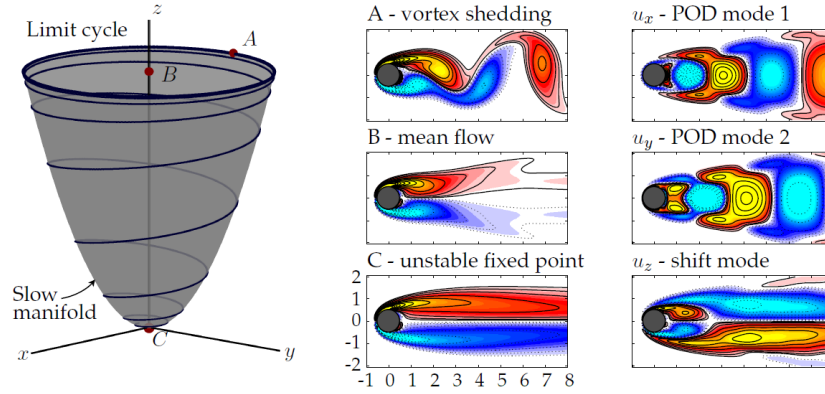


FIGURE 1.1. A depiction of the POD decomposition observed in the cylinder example described in (Brunton, Proctor, and Kutz, 2016).

We see the effects of the two harmonic oscillators on the vortex shedding and the mean flow of the model. The shift mode connects the mean flow with the unstable fixed point. Image is from (Brunton, Proctor, and Kutz, 2016)

this model is fairly simple with regards to its elements, according to (Brunton, Proctor, and Kutz, 2016), it took 30 years to come to this realization.

In 2016, (Brunton, Proctor, and Kutz, 2016) discuss a new data analysis approach called sparse identification of nonlinear dynamical systems (SINDy). This approach utilizes statistical methods driven by sparsity in order to identify governing equations for a dynamical system using real data. Vortex shedding was an area where they hypothesized such an approach may be of use. The example given by (Brunton, Proctor, and Kutz, 2016) depicted in figure 1.1 is similar to the one in (NOACK et al., 2003); they were trying to recreate the governing equations in 1.1. We note here that their approach proved somewhat successful. In the first of two data sets, they were able to identify the variables present in the paraboloid structure. However, they were less successful with the second data set, an issue they justify by the fact that it was twice as energetic; the method identified higher order terms not associated with what one would expect to see in a low-order representation of the system. Nonetheless, their moderate success with this approach sparked interest in the community and continues to draw others to explore the SINDy method.

In (Loiseau and Brunton, 2018) they developed a method called sparse Galerkin regression, which utilizes the SINDy approach, but applies energy conserving constraints associated with Galerkin projection. By doing this, they were able to ensure that the model they found adhered to dynamic constraints as well as physical ones

imposed by the transformation of the data. They tested this method on two well studied examples in fluid dynamics: vortex shedding in the wake of a cylinder and shear driven cavity flow. For each of these cases, they calculated the optimal unconstrained and constrained models and compared them. Their findings were that the unconstrained models were more accurate to the specific data, but didn't adhere to the constraints previously established. This failure overshadows the unconstrained models accuracy and suggests that the constrained models are the superior choice; they may not describe the specific data as well, but they do establish a better understanding of the system.

Whether utilizing SINDy or sparse Galerkin regression, computational efficiency becomes a issue. It may not be of concern in the cases we have just discussed, but what if we have a scenario where we have to examine more variables? Or even solve for more equations?

In (Braun, Agrey, and Martinuzzi, 2020), exploration is done on bluff body vortex shedding at higher Reynolds numbers. They were interested to see what would happen when they changed the cylinder for a flat plate when paired with a higher energy system. They were able to discern some of the qualities of the system from observation, but this would only give them ideas of what variables to examine when trying to find the governing equations. Unfortunately, this lead to the need to solve for a fourth governing equation tied to a mode referred to as the flapping mode and potentially even exploring a second harmonic oscillating pair. The need to look at four or even six modes exponentially increases the number of variables that need to be examined. If we were to attempt to solve this using the initial method discussed in (Brunton, Proctor, and Kutz, 2016), it would prove too cumbersome to solve in a reasonable amount of time.

Fortunately, the world of statistics is constantly evolving and more efficient methods of solving this problem now exist. Our research will attempt to consolidate some of these methods into a single approach. This consolidation will make this computational unfeasible problem much more palatable. First, we will look at transforming the data to reduce the number of equations that we must solve. Second, we will utilize the constraints discussed in (Loiseau and Brunton, 2018), but rather than using them after model construction, we will apply them before we begin model building.

Lastly, the method used in (Brunton, Proctor, and Kutz, 2016) to find a solution was sequential threshold least squares. Due to the discovery of cyclical coordinate descent and path seeking algorithms, LASSO has become an efficient tool in problems such as these and it is for this reason that we will apply it to our research.

Our manuscript will be broken down in the following manner; chapter 2 will define LASSO. In chapter 3 we will look at SINDy in detail and chapter 4 will contain the theory that supports our transformation of the data. Chapter 5 will outline the procedure we followed for model selection. Chapter 6 will discuss results. Finally we will end with chapter 7 which will contain concluding remarks and future directions.

Chapter 2

LASSO

2.1 Introduction

In the mid 90s, Tibshirani introduced the least absolute shrinkage and selection operator (LASSO) as a new tool in penalized regression. In this chapter we will begin by defining the LASSO and we will look at some key results discussed in (Tibshirani, 1996). We will then introduce cyclical coordinate descent and path seeking algorithms so that we may discuss the glmnet package in matlab and explain how it works.

2.2 Derivation

When we wish to model a system using real data, a good starting point is Ordinary Least Squares (OLS) Regression. To do this, we let (\mathbf{X}, \mathbf{y}) be a set such that \mathbf{y} is a n -vector of observations and \mathbf{X} a $n \times p$ design matrix with p predictor variables. With the addition of $\boldsymbol{\beta}$, a p -vector of regressor coefficients such that $\beta_j \in \mathbb{R} \forall j = 1, 2, \dots, p$, we can create a model of the system under observation. The optimal OLS model is the solution to

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} \left(n^{-1} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right) \quad (2.1)$$

It is important to note that the optimal OLS model need not be unique. While OLS is a reasonable approach under ideal conditions, it becomes less desirable in a situation where $p \gg n$: this leads to the issue of over-identification. This issue can be

addressed by adding a penalty parameter, $\lambda > 0$, to the optimization problem. Even with penalization there are still other issues that can arise during the model selection process. One issue is we obtain a model with many parameters, which may not be easy to interpret. Another issue is the model obtained may have high variance, which is not ideal for prediction. We will now look at two penalized regression approaches, specifically subset selection and ridge regression, both of which address these issues respectively.

Subset selection deals with the issue of obtaining a model that is easy to interpret. By penalizing the model for the number of parameters included, this method removes less influential parameters. The optimal subset model with k predictors is the solution to

$$\hat{\beta}(\lambda) = \min_{\beta: \|\beta\|_0 = k} \left(n^{-1}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda \|\beta\|_0 \right) \quad (2.2)$$

where $\|\beta\|_0 = \#\{\beta : \beta_j \neq 0, j = 1, 2, \dots, p\}$. While this reduces the number of parameters rendering a simpler model, it has the draw back of being a discrete process where we add predictors one at a time. The lack of continuity in the process can yield substantially different models with minute variations in the data. This is a result of the penalty being a constant that does not arise from a true norm space ($\|a\beta\|_0 \neq |a|\|\beta\|_0$). By making the model easier to interpret the issue of high variance is exacerbated.

To deal with high variance, we can use ridge regression. This method utilizes an l_2 -regularized penalty parameter yielding the following optimization problem

$$\hat{\beta}(\lambda) = \min_{\beta} \left(n^{-1}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda \|\beta\|_2^2 \right) \quad (2.3)$$

This penalty allows us to shrink the coefficients attached to our parameters through a continuous process, meaning minor perturbations in the data will not have a dramatic effect on the solution. While this shrinkage may address variability in the model, it will unlikely reduce any of our coefficients to zero leaving us with a complicated model and many parameters that may be redundant.

With the two major issues that we are trying to address it appears that by dealing with one we magnify the other. So what happens if we want a parsimonious model AND smaller variance? The answer is we utilize LASSO. By using an l_1 -regularized penalty we have the ability to shrink the variance of the OLS model as well as remove redundant regressors. The optimization problem for LASSO is

$$\hat{\beta}(\lambda) = \min_{\beta} \left(n^{-1}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda \|\beta\|_1 \right) \quad (2.4)$$

Just like the other two methods, there is no free lunch. We now have a scenario where we will suffer from the drawbacks of subset selection and ridge regression, but each to a lesser degree. By balancing both issues, the creation of a model that is easy to interpret and has reduced variance becomes possible. In the following section, we will outline how this works in the orthonormal case.

2.3 Orthonormal Case

(Tibshirani, 1996) discusses subset selection, ridge regression and LASSO in the orthonormal case and presents results regarding each method. Here we will prove the validity of these results under the requirement that our design matrix, $\mathbf{X}_{n \times p}$, be semi-orthogonal. For the proofs that follow we will assume that we are dealing with a situation where a unique solution exists. Under these conditions we will prove that the optimization problems for subset selection, ridge regression and LASSO (2.2, 2.3, and 2.4 respectively) give us models that have a reduced number of regressors, shrink our variance, and balance the previous two properties respectively. For all of the arguments that follow, our objective function for any given scenario will be defined as

$$\psi_q = n^{-1}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + (\lambda \|\beta\|)_q \quad (2.5)$$

where $(\lambda \|\beta\|)_q$ is the penalty parameter and norm product associated with model q , $q \in \{\text{Subset, LASSO, Ridge}\}$. In the case of OLS, $(\lambda \|\beta\|)_{\text{OLS}} = 0$.

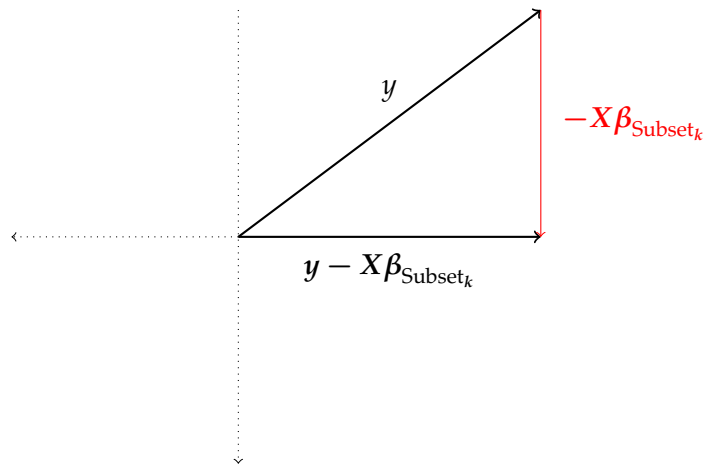
2.3.1 Subset Selection

We first turn our attention to subset selection to show the optimal model of size k is equivalent to selecting the k largest coefficients in magnitude from the full OLS model and setting the remaining coefficients to zero.

Let us consider a subset of k parameters such that $1 \leq k \leq p$. An optimal model in this setting can be obtained by looking at all possible models with k non-zero coefficients and comparing their objective functions. For an arbitrary, but fixed subset model with k non-zero coefficients we will have design matrix $\mathbf{X}_{\text{Subset}_k}$. Due to its semi-orthogonal nature, $\mathbf{X}_{\text{Subset}_k}$ can be obtained by mapping the columns associated with our non-zero parameters in \mathbf{X} to themselves and mapping all other columns to the null space. In order to show that the k parameters that will be included are the k parameters from the full model that have the largest coefficients in magnitude we will solve for our optimal β

$$\begin{aligned} \frac{d\psi_{\text{Subset}_k}}{d\beta} &= n^{-1}(-2\mathbf{X}^T(\mathbf{y} - (\mathbf{X}\beta)_{\text{Subset}_k})) \\ &= \mathbf{y} - (\mathbf{X}\beta)_{\text{Subset}_k} \end{aligned} \tag{2.6}$$

Here is a visual representation of optimization problem



Since the magnitude of vector \mathbf{y} is fixed, we can see that our solution is optimized

when $\mathbf{y} - (\mathbf{X}\boldsymbol{\beta})_{\text{Subset}_k}$ is close to the zero vector (i.e. when $(\mathbf{X}\boldsymbol{\beta})_{\text{Subset}_k}$ is large). Therefore the optimal model with k predictors will be the one that maximizes $\mathbf{X}\boldsymbol{\beta}$, i.e.

$$|(\mathbf{X}\boldsymbol{\beta})_{\text{Subset}_k}| \xrightarrow{\text{Orthogonal}} |\boldsymbol{\beta}_{\text{Subset}_k}| \Rightarrow \sqrt{\beta_1^2 + \beta_2^2 + \dots + \beta_k^2}$$

Here it is plain to see that a model containing the k largest coefficients in magnitude is equivalent to the optimal subset model with k predictors.

2.3.2 Ridge Regression

Next we prove the claim that the ridge regression l_2 -regularized penalty parameter shrinks the variance of the optimal OLS model.

$$\begin{aligned} \frac{d\psi_{\text{Ridge}}}{d\boldsymbol{\beta}} &= n^{-1}(-2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})) + 2\lambda\boldsymbol{\beta} \\ &\implies \mathbf{X}^T\mathbf{y} = \boldsymbol{\beta} + n\lambda\boldsymbol{\beta} \\ &\implies \boldsymbol{\beta} = (1 + n\lambda)^{-1}\mathbf{X}^T\mathbf{y} \end{aligned}$$

Because $\lambda > 0$ it follows that $(1 + n)\lambda > 1$ meaning that we are shrinking the $\boldsymbol{\beta}$ vector. Coincidentally, we also see that if $\lambda = 0$ we retrieve our OLS model.

2.3.3 LASSO

To prove the claims regarding LASSO we need to utilize subdifferential calculus. This will allow us to address LASSO's objective function which is not smooth when a regressor is equal to zero, which is crucial as we expect many of our regressors to be non-active. (Bühlmann, 2011) states this approach is valid since the LASSO objective function is convex and each regressor is bounded above. Therefore we can examine the optimal solutions at each simplex created by any number of regressors being non-active and expand to the original optimization problem. Let us take an arbitrary, yet fixed, model with parameter subset η . The optimization of the objective function for the given parameter subset becomes

$$\frac{d\psi_{L_\eta}}{d\boldsymbol{\beta}_{L_\eta}} = n^{-1}(-2\mathbf{X}_{L_\eta}^T(\mathbf{y} - (\mathbf{X}\boldsymbol{\beta})_{L_\eta})) + \text{sign}(\boldsymbol{\beta}_{L_\eta})\Lambda$$

which gives an optimal regression vector

$$\boldsymbol{\beta}_{L_\eta} = \mathbf{X}_{L_\eta}^T \mathbf{y} - \boldsymbol{\Lambda}_{L_\eta} \quad (2.7)$$

where $\boldsymbol{\Lambda}_{L_\eta} = \text{sign}(\boldsymbol{\beta}_{L_\eta}) \frac{n\lambda}{2}$. Due to the orthogonal nature of our example it follows that any active parameters in our LASSO model will be smaller in magnitude than the equivalent OLS model of the given parameter subset (i.e. it reduces variance in the model). We now focus our attention on showing that LASSO reduces the number of regressors in the model by proving the following statement:

$$\beta_i \neq 0 \rightarrow |\beta_i| > n\lambda \quad (2.8)$$

To prove 2.8, let us consider a second parameter subset, ν , where

$$\|\boldsymbol{\beta}_{L_\nu}\|_0^0 = \|\boldsymbol{\beta}_{L_\eta}\|_0^0 + 1; \quad \boldsymbol{\beta}_{L_\eta} \subset \boldsymbol{\beta}_{L_\nu}$$

Let us compare the objective functions of our two arbitrary, yet fixed, models.

$$\begin{aligned} & \psi_\nu - \psi_\eta \\ &= \boldsymbol{\beta}_{L_\nu}^T \boldsymbol{\beta}_{L_\nu} + \lambda |\boldsymbol{\beta}_{L_\nu}|_1 - 2\mathbf{y}^T (\mathbf{X}\boldsymbol{\beta})_{L_\nu} - \boldsymbol{\beta}_{L_\eta}^T \boldsymbol{\beta}_{L_\eta} - \lambda |\boldsymbol{\beta}_{L_\eta}|_1 + 2\mathbf{y}^T (\mathbf{X}\boldsymbol{\beta})_{L_\eta} \\ &= \beta_{L_\nu,k}^2 + n\lambda |\beta_{L_\nu,k}| - 2\mathbf{y}^T \mathbf{X}_{L_\nu, \{:,k\}} \beta_{L_\nu,k} \\ &\stackrel{n\lambda = |\beta_{L_\nu,k}| + \delta}{=} 2\beta_{L_\nu,k}^2 - 2\mathbf{y}^T \mathbf{X}_{L_\nu, \{:,k\}} \beta_{L_\nu,k} + \delta \\ &= 2\beta_{L_\nu,k} (\beta_{L_\nu,k} - \mathbf{y}^T \mathbf{X}_{L_\nu, \{:,k\}}) + \delta \\ &= \delta \end{aligned}$$

If $\delta > 0$ (i.e. $|\beta_{L_\nu,k}| < n\lambda$) then parameter subset η will be selected over parameter subset ν since $\psi_\eta < \psi_\nu$. Similarly, if $\delta = 0$ then parameter subset η will be selected over parameter subset ν since $\psi_\eta = \psi_\nu$ and $\|\boldsymbol{\beta}_{L_\nu}\|_0^0 > \|\boldsymbol{\beta}_{L_\eta}\|_0^0$ (i.e. subset η represents a more parsimonious model). Lastly, in the event that $\delta < 0$, parameter subset ν will be selected over parameter subset η since $\psi_\eta > \psi_\nu$. From these three cases we

see that 2.8 is true, which implies that LASSO serves as a subset selection mechanism. Once again the orthogonal nature of our example tells us that our optimal model will once again contain an active parameter subset of the original parameter set (note that this need not be a strict subset).

To see why LASSO can reduce coefficients to zero whereas ridge regression is unlikely to do so, we present a visualization of both of the optimization problems.

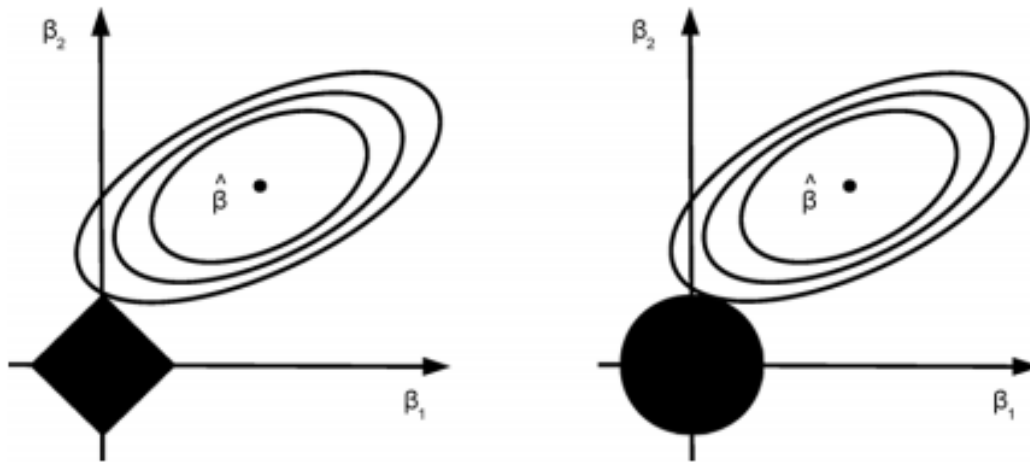


FIGURE 2.1. Geometric Representation of Optimization of $\hat{\beta}$ with l_1 (left) and l_2 (right) penalty parameters. On the left we see that tangency is possible at zero due to the 'sharp corners' of the l_1 penalty parameter. It is highly unlikely to obtain a tangent point at zero in the l_2 case on the right. The image is from (Bühlmann, 2011)

We see the l_1 -regularized penalty creating 'sharp corners' compared to the l_2 -regularized penalty. This special feature allows our ellipse of optimal solutions to be tangent to the l_1 -regularized penalty when β_1 is exactly zero. The same is not true for the l_2 -regularized penalty.

We return to our initial claim that LASSO balances the aspects of subset selection and ridge regression. This becomes apparent when we consider the penalty parameter. The larger our penalty parameter, the more we shrink the magnitude of our regression vector, which is done by only including larger regressors. This leads to a parsimonious model with minor reductions made to its variability. On the other hand, if we select a smaller penalty parameter we are left with a larger number of

active regressors, but with a more significant adjustment made to the variability in the model.

2.4 Cyclical Coordinate Descent and Path Seeking Algorithms

Initial deterrents surrounding the use of LASSO were determining an appropriate choice for λ as well as the computational cost required to solve the corresponding optimization problem. In their paper, (Wu and Lange, 2008) were among the first to address this issue through the use of cyclical coordinate descent. Their vision presented the first piece of the puzzle to bring LASSO into the statistical spotlight. It was later discovered that the addition of path seeking algorithms paired with Wu and Lange's research made the LASSO problem computationally feasible.

Before we address path seeking algorithms, we will first look at how cyclical coordinate descent works. Cyclical coordinate descent is an optimization method that addresses model parameters one at a time. Using an assumed initial model as a starting point, the method begins by finding an optimal solution for a single regressor while keeping the other ones fixed. Once this solution is found the process is repeated for the next regressor. Continuing in this fashion, the regressors are cycled through one at a time until the model converges. This may require multiple cycles to complete and ultimately the amount of time it takes this process to converge depends on how close the initial assumption is to the final model. This is where path seeking algorithms become important.

With path seeking algorithms, we are going to start by making an assumption whose use we will justify later. This assumption is that a model with a larger penalty parameter (i.e. fewer regressors) will prove to be a good initial assumption for a model with a smaller penalty parameter (i.e. more regressors). With this assumption in place, we can also assume that there exists a penalty parameter which we will call λ_0 , that is large enough to be associated with a model that has no regressors at all. If we shrink λ_0 slightly, generating λ_1 , we come to a scenario where our model will have a single regressor. At this point we will run our cyclical coordinate descent

algorithm and find our one-regressor model. Now that we have this model we can shrink λ_1 , obtaining λ_2 , whose model might have two regressors. With our original assumption, we can use the previous model as a 'warm start' for this iteration of the cyclical coordinate descent algorithm until we find our second model. By continuing along this path, we can obtain models for any number of λ s very quickly, eliminating the need for selecting an optimal value of λ .

In theory this method would work if our initial assumption was true. Unfortunately, this is not a practical assumption with real data. Furthermore, the assumption that any model with fewer regressors will be a subset of any larger model does not hold if we can not assume orthogonality. Regardless of these shortcomings, a culmination of research shown in (Ghaoui, Viallon, and Rabbani, 2011), (Tibshirani et al., 2012), and (Efron et al., 2004) provides support for the claim that this approach can generate a path that greatly improves computational efficiency with high probability of obtaining the correct optimal model. These papers will be discussed in more detail in the following section.

2.4.1 glmnet

For our research, the glmnet package in MATLAB, created by (Friedman, Hastie, and Tibshirani, 2010), was utilized. We turn to (Hastie, Qian, and Tay, 2021) to describe the underlying principles of this package. The main highlights that concern our research in this introduction are that we can utilize an elastic net penalty (LASSO being a specific example of elastic net) to build models for prediction through a k -fold cross-validation process. As mentioned above, this process relies on cyclical coordinate descent and path seeking algorithms, where glmnet presets the number of λ s considered along the path to be 100 and the number of folds for cross validation to be 10. These preset values can be changed with user input.

The underlying mechanisms of this package are derived from (Tibshirani et al., 2012), which introduced sequential strong laws for discarding predictors in LASSO optimization problems. This algorithm relies on theoretical results from (Efron et al.,

2004), which discusses the Least Angle Regression (LARS) algorithm, and (Ghaoui, Viallon, and Rabbani, 2011), which addresses concepts known as the Safe Feature Elimination (SAFE) rules. We will look at the key ideas behind these two papers to provide the foundation for which we can look at the strong sequential rules.

Beginning with the LARS algorithm, it uses a modified forward selection process to build a model. The first regressor introduced is the one that is most correlated with the data. Residuals are then obtained from this 'current' model, becoming the new response variable, and the remaining predictors are orthogonally projected from the model. The next regressor to enter the model is chosen under the same criteria as the first (the most correlated to the current response variable), but rather than simply entering the model, as would occur in a forward selection process, a bisecting vector is found between the vector created by the current model and the vector of the new regressor. This bisector becomes the new model and the process continues until desired. In their paper, (Efron et al., 2004) show that LASSO can be expressed as a modified version of the LARS algorithm and that both methods individually yield very similar models, albeit not identical.

We now shift our focus to (Ghaoui, Viallon, and Rabbani, 2011). In their paper, they establish the SAFE rules to be capable of optimizing the process of model specification in LASSO. Using simple inner product calculations, the approach is to solve a problem in a dual space that corresponds to the original optimization problem. This correspondence is limited to identifying features that are irrelevant to the solution of the original problem allowing them to be safely removed from consideration. Even with this limitation, significant improvements are made to computational efficiency. In some cases this improvement is on the order of magnitudes.

While we point out the main features of the LARS algorithm and the SAFE rules, we note here that we utilize the results their respective papers produce and that a deeper understanding of how these results were procured is beyond the scope of our research.

With the ground work outlined, we can now show how (Tibshirani et al., 2012) constructs the sequential strong laws. They begin with the SAFE rules inequality which states that the j^{th} predictor is discarded if

$$|\mathbf{x}_j^T \mathbf{y}| < \lambda - \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}} \quad (2.9)$$

The strong rules adjust this inequality by setting $\frac{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2}{\lambda_{\max}} = 1$. This approach is justified by the fact that LASSO assumes a standardized model. Then when the Cauchy-Schwartz inequality is applied, the result falls out. With the adjustment made, we have

$$|\mathbf{x}_j^T \mathbf{y}| < 2\lambda - \lambda_{\max} \quad (2.10)$$

This will remove more non-active predictors than the initial SAFE rules. The approach is improved even further by adding a sequential aspect that utilizes the residuals of the model found for one iteration and inserts them into the next iteration in the following way

$$|\mathbf{x}_j^T (\mathbf{y} - \hat{\boldsymbol{\beta}}(\lambda_{k-1}))| < 2\lambda_k - \lambda_{k-1} \quad (2.11)$$

This final inequality is used in the strong sequential approach. With high probability this approach significantly reduces computational cost by eliminating many predictors as potential candidates. While no actual measure has been attached to how often the algorithm is erroneous, the conditions under which counterexamples were found was rare. Even in the event that data is collected that leads to a model which is not optimal, the program is able to catch its mistake with a final check using the Karush-Kuhn Tucker conditions (KKT). Since the KKT conditions are grounded in optimization theory, we will forgo describing them here as they are outside of the scope of our research.

In (Tibshirani et al., 2012) real data was used to compare this method with the strong rules by themselves, the SAFE rules, and a recursive form of the SAFE rules. In their tests, the strong sequential approach significantly outperformed all other methods with no mistakes.

Chapter 3

Sparse Identification of Non-Linear Dynamical System (SINDy)

3.1 Introduction

In engineering, an important area of research is identifying the underlying models of dynamical systems from real data. In their paper, (Brunton, Proctor, and Kutz, 2016) discuss the issues that arise from real data and propose the use of sparse identification of nonlinear dynamical systems (SINDy) as a solution. By applying sparse identification methods to systems with the form

$$\frac{d}{dt}x(t) = f(x(t)) \quad (3.1)$$

(Brunton, Proctor, and Kutz, 2016) found some success with SINDy's ability to create low-order models for many important problems. The example in their paper of particular interest attempts to model vortex shedding in the wake of fluid flow past a cylinder. Before we get into the specifics of the experiment, we will first outline the SINDy process.

3.1.1 The Process

We will now break this process down doing our best to keep our notation consistent with the original paper (Brunton, Proctor, and Kutz, 2016).

Our first step is to obtain time series data for our system. This will establish \mathbf{X} to be a $T \times p$ matrix, consisting of T observation rows and p columns of variables

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{t_1}^T \\ \mathbf{x}_{t_2}^T \\ \vdots \\ \mathbf{x}_{t_T}^T \end{bmatrix} = \begin{bmatrix} (x_1)_{t_1} & (x_2)_{t_1} & \dots & (x_p)_{t_1} \\ (x_1)_{t_2} & (x_2)_{t_2} & \dots & (x_p)_{t_2} \\ \vdots & \vdots & \dots & \vdots \\ (x_1)_{t_T} & (x_2)_{t_T} & \dots & (x_p)_{t_T} \end{bmatrix}_{T \times p}$$

Optimally, the time step should be consistent between observations.

Since we are looking to find solutions to equations of the form 3.1, our next step is to obtain a second matrix containing the derivatives with respect to time for each of our p variables. This matrix, $\dot{\mathbf{X}}$, can either be collected at the same time as the data itself through observation, or it can be calculated afterwards using numerical methods.

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}_{t_1^*}^T \\ \dot{\mathbf{x}}_{t_2^*}^T \\ \vdots \\ \dot{\mathbf{x}}_{t_m^*}^T \end{bmatrix} = \begin{bmatrix} (\dot{x}_1)_{t_1^*} & (\dot{x}_2)_{t_1^*} & \dots & (\dot{x}_p)_{t_1^*} \\ (\dot{x}_1)_{t_2^*} & (\dot{x}_2)_{t_2^*} & \dots & (\dot{x}_p)_{t_2^*} \\ \vdots & \vdots & \dots & \vdots \\ (\dot{x}_1)_{t_m^*} & (\dot{x}_2)_{t_m^*} & \dots & (\dot{x}_p)_{t_m^*} \end{bmatrix}$$

We note here that our derivative matrix only has m rows. The value of m is dependent on the method used to obtain the derivatives. As an example, our approach calculates the derivatives using a 4th order differential scheme and therefore $m = T - 4$. It is important to note that our method of calculating derivatives requires prior observations which is why we begin indexing at t_1^* (i.e. $t_1 \neq t_1^*$).

With our data and derivatives obtained, we move to our third step: creating a library of candidate variables. In order to obtain this library, we will create a matrix of functions Θ and apply them to our data. Some examples of functions that could be present in Θ are identity, polynomials, sin, cosine, logarithms, etc.

$$\Theta(\mathbf{X}) = \begin{bmatrix} 1 & \mathbf{X} & \mathbf{X}^{P_2} & \dots & \sin(\mathbf{X}) & \dots \end{bmatrix}$$

In the above example \mathbf{X}^{P_j} represents all possible polynomials of order j that can be produced by our observed variables. As an example, if $j=2$ and our variables are x ,

y and z then

$$\mathbf{X}^{P_2}(t) = \begin{bmatrix} \mathbf{x}^2(t) & \mathbf{xy}(t) & \mathbf{xz}(t) & \mathbf{y}^2(t) & \mathbf{yz}(t) & \mathbf{z}^2(t) \end{bmatrix}$$

Once our library of functions is constructed we can apply it to our observed data and then represent the solution as

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi$$

where Ξ represents a vector of coefficients associate with each term in our collection. If we compare this with the simple regression equation $y = \mathbf{X}\beta$ we see that they are of the same form. Therefore we have established a simple regression problem for which we assume that our solution is sparse by design and many of our regressors in Ξ will be 0.

The desire to utilize statistical methods in this situation is highlighted by the fact that real life measurements are affected by outside noise, making it difficult to distinguish the system's essential elements from confounding ones. LASSO's ability to perform a pseudo subset selection allows us to eliminate variables with less influence on the system with relative confidence. This will allow us to weed through some of the noise that is unavoidable in the system, and allows us to get to the root of its essential elements.

3.2 Adapting the SINDy Method

In order to use the SINDy method for vortex shedding problems, we are required to transform our data so that we can use ordinary differential methods. Using Proper Orthonormal Decomposition (POD), we are able to find the principle components (i.e. modes), therefore allowing us to find a low-order approximation of the system. With this transformation comes a series of constraints that the solution must adhere to in order for the solution to be applicable in the real world. In their paper, (Loiseau and Brunton, 2018) recognize this and introduce the use of these constraints into the SINDy algorithm. While their paper found the application of these constraints to

be useful in finding a valid model, it did not apply the constraints prior to model building. In the follow-up paper, (Agrey et al., 2022) considered a modification of the method of Loiseau and Brunton, by applying the constraints in the LASSO fitting stage before variable selection is performed. This way when a variable is removed then some other variables have to be removed as well due to the constraints imposed, hence the algorithm will choose either keeping all variables affected by a particular constraint or removing them all depending on which action results in better model fitting performance. Hence it can potentially eliminate more variables. This will greatly reduce the number of variables that need to be examined.

On top of applying these constraints prior to model construction, we will also include a transformation within our data. The specifics surrounding our approach deal with the sin/cosine relationship established between our first two POD modes. We can exploit this relationship by transforming these two modes into a new mode in polar coordinates. This transformation will be described in greater detail in the next chapter. By transforming these two modes, as well as the known linear constraints, we will improve computational efficiency by reducing the number of equations sought by one. Once again, we will validate this approach in the next chapter.

In order to test our constrained SINDy approach, we will utilize the data sets in (Brunton, Proctor, and Kutz, 2016) and will attempt to create approximations of the equations that should be found within that data set. Provided we attain reasonable success with this, we will then explore the data from (Braun, Agrey, and Martinuzzi, 2020) and attempt to model the governing equations of those systems. By doing this, we may be able to shed some light on their problem in the hopes that others may be able to use our results as a starting point for future research.

Chapter 4

Theory

4.1 Theory

In this chapter, we will outline the transformation we intend to make with regards to the data and the constraints found in (Loiseau and Brunton, 2018). We will propose a sparse regression model for the transformed variables and show that this model can be represented as a sparse regression model in the original variables with a new set of linear constraints. Hence we will be able to use the software developed by Agrey and Braun which performs sparse regression with linear constraints for model fitting in terms of the original variables. While the tool they developed could have been modified to deal with transformed data, drastic changes would need to be made potentially compromising the integrity of what they had constructed. We therefore deemed the more reasonable approach to be to improve what they had done, rather than change it, therefore necessitating the need for the proposed model in terms of the original variables.

4.1.1 Notation

Before we begin, we would like to define our variables and also explain notation aspects that may be new or unconventional to the reader. First, our variables will be defined in the following way: x and y will represent our most energetic harmonic oscillator pair. The representation of these two variables in polar coordinates will be done using A for magnitude and θ for direction. We will be using Θ to represent our library of candidate variables. A subscript attached to Θ denotes the number

of candidate variables in the library: the number of columns in the matrix. It is important to note that we are using the same Greek letter to represent two unrelated variables, Θ and θ , and attention to whether it is represented in the upper or lower case is required. The final variable definitions we will provide are for the shift mode, z , the flapping mode, w , and our second most energetic harmonic oscillators, u and v . We note here that we will exclude w , u and v in proofs. This will not affect the validity of our results, but is simply done to make presentation of calculations more accessible to the reader.

Our analysis intends to solve a number of ODEs, and therefore we will have a set of coefficients for each variable of interest. When we are referring to the set of coefficients tied to a specific variable, a superscript will be used. As an example, β^z refers to the coefficients associated with the solution to \dot{z} . A subscript attached to a coefficient references the specific regressor. An important note here is that the subscript is dependent on the size of Θ , which will vary depending on the number of modes under consideration. As an example, when we are modeling 3 modes, β_5^x is the coefficient tied to x^2 (5th library element) in our model of \dot{x} . When we are modeling 4 modes, β_5^x is the coefficient tied to w in our model of \dot{x} . The indexing of our coefficients throughout the paper will be consistent with the following criteria.

1. Polynomials of lower orders will have smaller index values than those of higher orders. For example, x will have a lower index than x^3 .
2. Polynomials of the same order will be organized by mode with the order of priority being x, y, z, w, u and v . For example, x will have a lower index than y ; xy will have a lower index than xz , which has a lower index than y^2 .
3. Polynomials of the same order with the same modes will be organized based on the order of priority with a higher order taking precedence. For example, x^2y will have a lower index than xy^2 .

When we refer to our regressors using β , this will imply the matrix layout of our solution. There are times when we wish to express our regressor coefficients as a single vector. We will use a colon to represent the stacking of rows of a matrix.

Specifically,

$$\beta(\cdot) = \begin{bmatrix} \beta^x \\ \beta^y \\ \beta^z \end{bmatrix}$$

In order to express the dimension of a matrix we will use the standard notation of a subscript attached to the matrix variable in the form of $m \times n$. There are times when we will be concerned with specific row and column subsets of a matrix. In these situations we will use a bracketed subscript in the form of $(i : m, j : n)$ representing the i^{th} row to the m^{th} row and the j^{th} column to the n^{th} column of a given matrix.

Lastly, we will note that there may be other situations where new notation or definitions are introduced. We will provide explanations as these cases arise due to the specificity to their scenarios. With that we return to the original objective of this chapter.

4.2 Model Comparisons

Model 1 will represent the model constructed using polar coordinates and Model 2 will be the original model to which we can apply the Galerkin constraints. While we intend to show the validity of our approach with regard to a three mode system, the results will hold for higher dimensions. We refrain from showing these proofs as they are essentially the same with the addition of trivial calculations.

4.2.1 Model 1

Let

$$A(t) = x^2(t) + y^2(t) \tag{4.1}$$

$$x(t) = \sqrt{A(t)} \cos \theta(t) \tag{4.2}$$

$$y(t) = \sqrt{A(t)} \sin \theta(t) \tag{4.3}$$

$$\theta(t) = \arctan(y(t)/x(t))$$

$$z(t) = z(t)$$

Model 1 is:

$$\dot{A} = A(t)\Theta_{10}\tilde{\beta}^A \quad (4.4)$$

$$\dot{z} = \Theta_{20}\beta^z \quad (4.5)$$

Where $\tilde{\beta}^A$ is a 10-dimensional vector and β^z is a 20-dimensional vector (i.e. $\tilde{\beta}^A = [\tilde{\beta}_1^A, \tilde{\beta}_2^A, \dots, \tilde{\beta}_{10}^A]^T_{1 \times 10}$ and $\beta^z = [\beta_1^z, \dots, \beta_{20}^z]^T_{1 \times 20}$). The difference in vector length is a result of our requirement that all solutions be of order 3 in terms of the original modes.

Our next step is to express $\tilde{\beta}^A$ and β^z as an array. Therefore we need a 20×1 vector that is equivalent to $\tilde{\beta}^A$. This new vector, $\tilde{\beta}^{A*}$, is easily found through a one-to-one mapping of the initial vector with the added extension of 10 more dimensions that map to the null space (i.e. $\tilde{\beta}^{A*} = [\tilde{\beta}_1^A, \dots, \tilde{\beta}_{10}^A, 0, \dots, 0]^T_{1 \times 20}$). We can now express our vectors as an array, which we will call $\tilde{\beta}$.

$$\tilde{\beta}^T = \begin{array}{cccccccccccc} 1 & x & y & z & x^2 & xy & xz & y^2 & yz & z^2 & & \\ \tilde{\beta}_1^A & \tilde{\beta}_2^A & \tilde{\beta}_3^A & \tilde{\beta}_4^A & \tilde{\beta}_5^A & \tilde{\beta}_6^A & \tilde{\beta}_7^A & \tilde{\beta}_8^A & \tilde{\beta}_9^A & \tilde{\beta}_{10}^A & \dots & \\ \beta_1^z & \beta_2^z & \beta_3^z & \beta_4^z & \beta_5^z & \beta_6^z & \beta_7^z & \beta_8^z & \beta_9^z & \beta_{10}^z & \dots & \end{array}$$

$$\begin{array}{cccccccccccc} x^3 & x^2y & x^2z & xy^2 & xyz & xz^2 & y^3 & y^2z & yz^2 & z^3 & & \\ \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ \dots & \beta_{11}^z & \beta_{12}^z & \beta_{13}^z & \beta_{14}^z & \beta_{15}^z & \beta_{16}^z & \beta_{17}^z & \beta_{18}^z & \beta_{19}^z & \beta_{20}^z & \end{array} \quad \textcolor{red}{1}$$

¹The top row is an indexing aide and is not a part of the matrix

4.2.2 Model 2

Now we will begin constructing Model 2, β , using equations 4.2, 4.3 and 4.5

$$\begin{aligned}
 \dot{x} &= \frac{1}{2\sqrt{A(t)}} \dot{A} \cos\theta(t) - \sqrt{A(t)} \sin\theta \times \dot{\theta} \\
 &= \frac{\sqrt{A(t)} \cos\theta(t)}{2} \Theta_{10} \tilde{\beta}^A - \sqrt{A(t)} \sin\theta(t) \times \dot{\theta} \\
 &= \frac{x(t)}{2} \Theta_{10} \tilde{\beta}^A - y(t) \times \dot{\theta}
 \end{aligned} \tag{4.6}$$

$$\begin{aligned}
 \dot{y} &= \frac{1}{2\sqrt{A(t)}} \dot{A} \sin\theta(t) + \sqrt{A(t)} \cos\theta \times \dot{\theta} \\
 &= \frac{\sqrt{A(t)} \sin\theta(t)}{2} \Theta_{10} \tilde{\beta}^A + \sqrt{A(t)} \cos\theta \times \dot{\theta} \\
 &= \frac{y(t)}{2} \Theta_{10} \tilde{\beta}^A + x(t) \times \dot{\theta}
 \end{aligned} \tag{4.7}$$

Using equations 4.6, and 4.7 we obtain the following

$$\dot{x} = x \cdot \left(\frac{\tilde{\beta}_1^A}{2} + x \frac{\tilde{\beta}_2^A}{2} + \dots + yz \frac{\tilde{\beta}_9^A}{2} + z^2 \frac{\tilde{\beta}_{10}^A}{2} \right) - y \times \dot{\theta} \tag{4.8}$$

$$\dot{y} = y \cdot \left(\frac{\tilde{\beta}_1^A}{2} + x \frac{\tilde{\beta}_2^A}{2} + \dots + yz \frac{\tilde{\beta}_9^A}{2} + z^2 \frac{\tilde{\beta}_{10}^A}{2} \right) + x \times \dot{\theta} \tag{4.9}$$

When we express β^T as an array using 4.8 and 4.9, we see that there are many zeros: a desired result.

	1	x	y	z	x^2	xy	xz	y^2	yz	z^2	
\dot{x}	0	$\frac{\tilde{\beta}_1^A}{2}$	$-\dot{\theta}$	0	$\frac{\tilde{\beta}_2^A}{2}$	$\frac{\tilde{\beta}_3^A}{2}$	$\frac{\tilde{\beta}_4^A}{2}$	0	0	0	...
\dot{y}	0	$\dot{\theta}$	$\frac{\tilde{\beta}_1^A}{2}$	0	0	$\frac{\tilde{\beta}_2^A}{2}$	0	$\frac{\tilde{\beta}_3^A}{2}$	$\frac{\tilde{\beta}_4^A}{2}$	0	...
\dot{z}	β_1^z	β_2^z	β_3^z	β_4^z	β_5^z	β_6^z	β_7^z	β_8^z	β_9^z	β_{10}^z	...

(4.10)

...	x^3	x^2y	x^2z	xy^2	xyz	xz^2	y^3	y^2z	yz^2	z^3	
...	$\frac{\tilde{\beta}_5^A}{2}$	$\frac{\tilde{\beta}_6^A}{2}$	$\frac{\tilde{\beta}_7^A}{2}$	$\frac{\tilde{\beta}_8^A}{2}$	$\frac{\tilde{\beta}_9^A}{2}$	$\frac{\tilde{\beta}_{10}^A}{2}$	0	0	0	0	
...	0	$\frac{\tilde{\beta}_5^A}{2}$	0	$\frac{\tilde{\beta}_6^A}{2}$	$\frac{\tilde{\beta}_7^A}{2}$	0	$\frac{\tilde{\beta}_8^A}{2}$	$\frac{\tilde{\beta}_9^A}{2}$	$\frac{\tilde{\beta}_{10}^A}{2}$	0	
...	β_{11}^z	β_{12}^z	β_{13}^z	β_{14}^z	β_{15}^z	β_{16}^z	β_{17}^z	β_{18}^z	β_{19}^z	β_{20}^z	

Based on our derivation of 4.10, it follows that Model 1 and Model 2 are equivalent.

4.2.3 Constraints

When examining this system using the first three modes, β can be expressed as

$$\beta = \begin{array}{cccccccccccc} & 1 & x & y & z & x^2 & xy & xz & y^2 & yz & z^2 & \\ \beta = & \beta_1^x & \beta_2^x & \beta_3^x & \beta_4^x & \beta_5^x & \beta_6^x & \beta_7^x & \beta_8^x & \beta_9^x & \beta_{10}^x & \dots \\ & \beta_1^y & \beta_2^y & \beta_3^y & \beta_4^y & \beta_5^y & \beta_6^y & \beta_7^y & \beta_8^y & \beta_9^y & \beta_{10}^y & \dots \\ & \beta_1^z & \beta_2^z & \beta_3^z & \beta_4^z & \beta_5^z & \beta_6^z & \beta_7^z & \beta_8^z & \beta_9^z & \beta_{10}^z & \dots \\ & x^3 & x^2y & x^2z & xy^2 & xyz & xz^2 & y^3 & y^2z & yz^2 & z^3 & \\ \dots & \beta_{11}^x & \beta_{12}^x & \beta_{13}^x & \beta_{14}^x & \beta_{15}^x & \beta_{16}^x & \beta_{17}^x & \beta_{18}^x & \beta_{19}^x & \beta_{20}^x & \\ \dots & \beta_{11}^y & \beta_{12}^y & \beta_{13}^y & \beta_{14}^y & \beta_{15}^y & \beta_{16}^y & \beta_{17}^y & \beta_{18}^y & \beta_{19}^y & \beta_{20}^y & \\ \dots & \beta_{11}^z & \beta_{12}^z & \beta_{13}^z & \beta_{14}^z & \beta_{15}^z & \beta_{16}^z & \beta_{17}^z & \beta_{18}^z & \beta_{19}^z & \beta_{20}^z & \end{array}$$

According to (Loiseau and Brunton, 2018)², β is subject to the following dynamic constraint;

$$\beta_9^x + \beta_7^y + \beta_6^z = 0$$

as well as the following kinematic constraints:

$$\beta_5^x = \beta_8^y = \beta_9^z$$

$$\beta_6^x = -\beta_5^y$$

$$\beta_8^x = -\beta_6^y$$

$$\beta_7^x = -\beta_5^z$$

$$\beta_{10}^x = -\beta_7^z$$

$$\beta_9^y = -\beta_8^z$$

$$\beta_{10}^y = -\beta_9^z$$

²The index between (Loiseau and Brunton, 2018) and our results differ by 1 due to the inclusion of the identity in our design matrix.

Our dynamic constraint is an energy-based constraint whereas the kinematic constraints are also realizability constraints. For more information on these constraints, please refer back to (Loiseau and Brunton, 2018). When applying these constraints to $\beta(\cdot)$ we can express them as a matrix, E (this matrix can be seen in appendix A.1). Expressed in this form, the following equality must hold.

$$E \cdot \beta(\cdot) = \mathbf{0} \quad (4.11)$$

In order to apply equivalent constraints to Model 2, we require a constraint matrix, \tilde{E} , equivalent to E such that

$$\tilde{E} \cdot \tilde{\beta}(\cdot) = \mathbf{0}$$

.

After simple calculations we obtain the following new constraints

$$\tilde{\beta}_2^A = 0 \quad (4.12)$$

$$\tilde{\beta}_3^A = 0 \quad (4.13)$$

$$\tilde{\beta}_4^A = -2 \cdot \beta_5^z = -2 \cdot \beta_8^z \quad (4.14)$$

$$\beta_6^z = 0 \quad (4.15)$$

$$\beta_7^z = 0 \quad (4.16)$$

$$\beta_9^z = 0 \quad (4.17)$$

A visualization of these constraints as a matrix, $\tilde{E}_{(1:10,1:40)}$ can be seen in A.2. We note here that our new constraint matrix is actually $\tilde{E}_{20 \times 40}$ and that the final 10 rows address the extension of $\tilde{\beta}^A$ to $\tilde{\beta}^{\tilde{A}*}$ (mapping extension components to the null space). A full visualization of \tilde{E} would be

$$\tilde{E} = \begin{bmatrix} \tilde{E}_{(1:10,1:10)} & \tilde{E}_{(1:10,11:20)} & \tilde{E}_{(1:10,21:40)} \\ 0_{10 \times 10} & I_{10 \times 10} & 0_{10 \times 20} \end{bmatrix}_{20 \times 40}$$

4.3 Finding β from $\tilde{\beta}$

With our equivalences established, we wish to show that given $\tilde{\beta}$, we can obtain β as expressed in 4.10. For this task we will require two components. The simplest component is $\vec{\theta}_{60 \times 1}$ to address the scalar $\dot{\theta}$ in β . This vector will contain an entry of $-\dot{\theta}$ in the 3^{rd} row and $\dot{\theta}$ in the 22^{nd} row. These rows correspond to the appropriate rows of $\beta(\cdot)$ containing a multiple of $\dot{\theta}^3$. All remaining rows of $\vec{\theta}_{60 \times 1}$ will be 0. A visual representation of $\vec{\theta}_{60 \times 1}$ is presented in A.3. The more complicated component we require is our transformation matrix, which we will define as \tilde{A}

$$\tilde{A} = \begin{bmatrix} \tilde{A}^x_{20 \times 20} & 0_{20 \times 20} \\ \tilde{A}^y_{20 \times 20} & 0_{20 \times 20} \\ 0_{20 \times 20} & I_{20 \times 20} \end{bmatrix}_{60 \times 40}$$

The submatrices \tilde{A}^x and \tilde{A}^y represent the aspects responsible for mapping the coefficients for \dot{A} to \dot{x} and \dot{y} respectively.

In 4.10, we see that $\tilde{\beta}_1^A$ corresponds to the first column in β^x with a variable containing x as well as the first column in β^y with a variable containing y (β_2^x and β_3^y respectively). Similarly, $\tilde{\beta}_i^A$ corresponds to the i^{th} column in β^x with a variable containing x as well as the i^{th} column in β^y with a variable containing y . By developing an algorithm to identify the indices of columns in Θ associated with a given mode, the construction of our submatrices becomes simple.

When dealing with x , the pattern is fairly simple, but y proves to be a little more difficult. Nonetheless, both patterns can be established through a recursive relationship. We will utilize the following diagram to illustrate this relationship. Even though our diagram will only depict the relationship for a 3-mode model, the results that follow apply to models with any number of modes.

³These positions are in reference to a 3-mode model. When more modes are considered, these indices will change.

$$\left\{ \begin{array}{ll} \mathbf{X}^{P_0} & 1 \\ \downarrow & x \cdot [1], y \cdot [1], z \cdot [1] \quad \downarrow \\ \mathbf{X}^{P_1} & x, y, z \\ \downarrow & x \cdot [x y z], y \cdot [x y z], z \cdot [x y z] \quad \downarrow \\ \mathbf{X}^{P_2} & x^2, xy, xz, y^2, yz, z^2 \\ \downarrow & x \cdot [x^2 xy xz y^2 yz z^2], y \cdot [x^2 \dots y^2 yz z^2], z \cdot [x^2 \dots z^2] \quad \downarrow \\ \mathbf{X}^{P_3} & x^3, x^2y, x^2z, \dots \dots \dots, z^3 \end{array} \right. \quad (4.18)$$

In 4.18 we see that the number of elements containing any mode (i.e. x, y, z) in \mathbf{X}^{P_i} is always equal to the total number of elements in $\mathbf{X}^{P_{i-1}}$. Let

$$M_i = \# \text{ of elements in } \mathbf{X}^{P_{i-1}} = \# \text{ of elements in } \mathbf{X}^{P_i} \text{ containing } x$$

$$N_i = \# \text{ of elements in } \mathbf{X}^{P_{i-1}} = \# \text{ of elements in } \mathbf{X}^{P_i} \text{ containing } y$$

Returning to our initial objective, the elements of \mathbf{X}^{P_i} that contain x are simply the first M_i elements of \mathbf{X}^{P_i} . When we look at the pattern for the elements containing y it is more difficult since the elements will not be consecutive for higher orders and the spacing between non consecutive elements is not consistent. Fortunately we can define the position of the the elements based on the previous order. For \mathbf{X}^{P_i} the position of the first N_{i-1} elements containing y will be the same as the positions of the elements containing y in $\mathbf{X}^{P_{i-1}}$. The remaining elements containing y will be in the positions $\left((N_i + 1) : (2N_i - N_{i-1}) \right)$.

With this we have sufficient information to construct our submatrices.

Let us first construct two vectors, $x_{20 \times 1}^{ld}$ and $y_{20 \times 1}^{ld}$ where

$$x_i^{ld} = \begin{cases} 1 & \text{if } i \text{ is a column index for a variable containing } x \\ 0 & \text{O.W.} \end{cases}$$

$$y_i^{Id} = \begin{cases} 1 & \text{if } i \text{ is a column index for a variable containing } y \\ 0 & \text{O.W.} \end{cases}$$

The construction of these vectors can be done by creating smaller vectors using 1s/0s to represent the indices of elements containing/not containing a given mode for \mathbf{X}^{P_0} , \mathbf{X}^{P_1} , \mathbf{X}^{P_2} , and \mathbf{X}^{P_3} . Concatenating these smaller vectors together gives us $x_{20 \times 1}^{Id}$ and $y_{20 \times 1}^{Id}$. For visual representations of these constructions, see [A.4](#). Using vector $x_{20 \times 1}^{Id}$ let us create a collection of coordinates, $(j, k) \in K$ where k corresponds to the k^{th} row of $x_{20 \times 1}^{Id}$ containing a 1 and

$$j = \sum_1^k x_j^{Id}$$

The first three coordinate pairs in K are (2,1), (5,2) and (6,3). Then we let

$$\tilde{A}^x_{(j,k)} = \begin{cases} \frac{1}{2} & (j,k) \in K \\ 0 & (j,k) \notin K \end{cases}$$

We construct \tilde{A}^y in the same fashion using $y_{20 \times 1}^{Id}$. For a visual representation of $\tilde{A}^x_{20 \times 20}$ and $\tilde{A}^y_{20 \times 20}$ please see [A.5](#).

Proposition

$$\tilde{A} \cdot \tilde{\beta}(\cdot) + \vec{\theta}_{60 \times 1} = \beta(\cdot) \quad ^4$$

Proof: Due to the large size of the matrices, the calculations for our proof can be verified in [B.1.1](#) and [B.1.2](#), which will only show calculations for the first 10 columns of \tilde{A}^x and \tilde{A}^y since the remaining columns contain zero entries rendering the calculations trivial.

$$\tilde{A}_{(1:20,1:40)} \cdot \tilde{\beta}(\cdot) + \vec{\theta}_{(1:20,1)} = \tilde{A}^x \cdot \beta^{\tilde{A}^*} + \vec{\theta}_{(1:20,1)} = \beta^x$$

$$\tilde{A}_{(21:40,1:40)} \cdot \tilde{\beta}(\cdot) + \vec{\theta}_{(21:40,1)} = \tilde{A}^y \cdot \beta^{\tilde{A}^*} + \vec{\theta}_{(21:40,1)} = \beta^y$$

⁴This proposition specifically pertains to the 3-mode model, but can easily be expanded to higher dimensions with minor modifications.

$$\tilde{A}_{(41:60,1:40)} \cdot \tilde{\beta}(\cdot) + \vec{\theta}_{(41:60,1)} = I_{20 \times 20} \cdot \tilde{\beta}^z = \beta^z$$

□

We can now express Model 2 as

$$\begin{aligned}\dot{x} &= \frac{x(t)}{2} \Theta_{10} \tilde{\beta}^A - y(t) \times \dot{\theta} \\ \dot{y} &= \frac{y(t)}{2} \Theta_{10} \tilde{\beta}^A + x(t) \times \dot{\theta} \\ \dot{z} &= \Theta_{20} \beta^z\end{aligned}$$

subject to the constraints

$$E \cdot \tilde{A} \cdot \tilde{\beta}(\cdot) + E \cdot \vec{\theta}_{60 \times 1} = \mathbf{0}$$

We can also express Model 1

$$\begin{bmatrix} \dot{A} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A \cdot \Theta_{20} & 0 \\ 0 & \Theta_{20} \end{bmatrix} \cdot \tilde{\beta}(\cdot)$$

or similarly

$$\begin{bmatrix} \frac{\dot{A}}{A} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \Theta_{20} & 0 \\ 0 & \Theta_{20} \end{bmatrix} \cdot \tilde{\beta}(\cdot)$$

subject to the constraints

$$\tilde{E} \cdot \tilde{\beta}(\cdot) = \mathbf{0}$$

Now that we have established an equivalence between models and adapted the constraints, we end this chapter and move to the next where we will discuss our experimental design.

Chapter 5

Design

5.1 Overview

In this chapter, we will present the method by which we applied our theoretical research. To test our proposed transformation, we attempted to model three different data sets. In each case we utilized a portion of the data for model construction while holding a portion back for testing. Selected models were within one standard error of the model with the minimal mean square error. They were also the most parsimonious model; models with improved mean square error resulted in no substantial change to the model. These model were associated with an 'elbow' in the graph of mean squared error vs. $\log(\lambda)$. This chapter will be organized in the following manner. We will begin by outlining the data sets we will be dealing with and then will proceed to outlining adaptations we made to inherited code on the project, paying particular attention to our contributions to the research.

5.2 Data Introduction

5.2.1 Cylinder Data

The first data set comes from the previously discussed cylinder example presented in (Brunton, Proctor, and Kutz, 2016). In this data set, we have two separate trials containing 5000 and 3000 evenly spaced observations. The spacing between observations is 0.2 dimensionless time units. Each trial contains 9 POD modes. The POD modes are found through eigenvalue decomposition as in principal component analysis. The data matrix is formed by arranging columns to represent the spatial points

and the rows to represent time points. The columns are then centered by subtracting the column means. If we denote the resulting matrix by \mathbf{X} , the eigenvalue decomposition is applied to the $(\mathbf{X}^t)\mathbf{X}$ matrix. The principal components are our modes and the loading coefficients give the spatial modes. Spectral graphs depict the energy frequencies that are most prevalent in a particular mode.

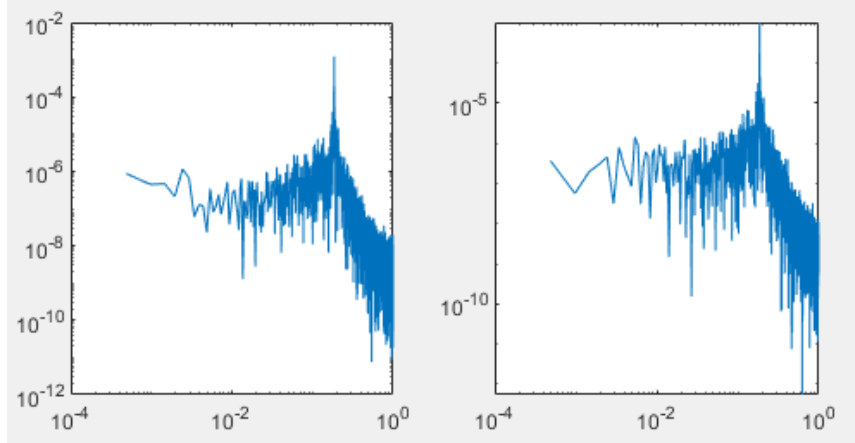


FIGURE 5.1. Spectral decomposition of the harmonic oscillating pair. As we can see, there is a spike in the energy (peak in the graph) at 10^{-1} , telling us the modes have similar active frequencies.

In this particular case, the modes we are interested in are our most energetic harmonic oscillators (1 and 2) as well as our shift mode (9). We will utilize the first trial for model construction and reserve the second for testing. For a deeper understanding of how the data was collected, we direct the reader to refer to the original paper (Brunton, Proctor, and Kutz, 2016).

Based on (NOACK et al., 2003) as well as (Brunton, Proctor, and Kutz, 2016), we have a good theoretical knowledge of the equations as well as the variables that we should find using our method. We will use this case as a test for the adequacy of our approach.

5.2.2 Flat Plate Data

The other two data sets we will be working with are presented in (Braun, Agrey, and Martinuzzi, 2020). In their paper they explore vortex shedding in the wake of a flat plate with closed ends and a flat plate with open ends. The following excerpt is taken from their paper and describes the system that we are trying to model:

The near-wake flows behind high aspect ratio (span-to-chord, $s/c > 40$) thin flat plates normal to a uniform stream of velocity, U_∞ , are considered for two conditions: open ends (OE), without end plates; and closed ends (CE), with end plates isolating the recirculation region from the ambient pressure. Experiments were conducted in an open jet suction-type wind tunnel.

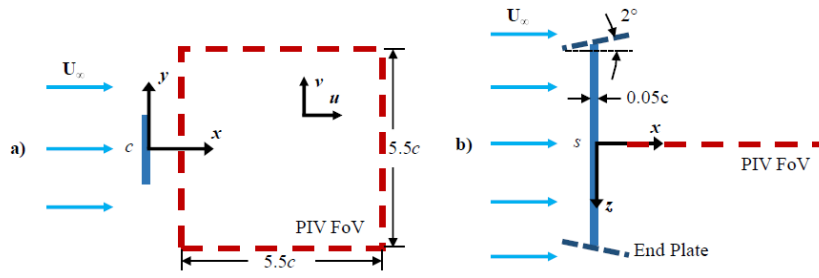


FIGURE 5.2. A visualization of the system under investigation (PIV := Particle Image Velocimetry FoV:=Field of View). The image is taken from (Braun, Agrey, and Martinuzzi, 2020)

For more detail of the experiment itself, please refer to (Braun, Agrey, and Martinuzzi, 2020). In both of these data sets there are three separate trials each containing 2728 evenly spaced observations. The spacing between observations in the closed end data set is 0.001 dimensionless time units. The spacing between observations in the open end data sets is 0.0017 dimensionless time units. Each trial contains 8 POD modes. The modes we are interested in are our most energetic harmonic oscillators (1 and 2), the shift mode (4), the flapping mode (3), and the second most energetic harmonic oscillators (5 and 6). The spectra of these modes, for both the open and closed ends data, can be seen in E. We will concatenate the first two trials for model construction and retain the third trial as our test set.

5.3 MATLAB code

The ground work for our MATLAB code comes from Eric Braun and Kaden Agrey. The code was originally created to analyze the open and closed ends data sets from (Braun, Agrey, and Martinuzzi, 2020). We adapted the code to analyze the data sets

with our approach. This may result in some redundancies left in our adaptation of the code. In this section, we will describe the major features of the code that are pertinent the application of our constrained LASSO approach.

5.3.1 Master Script

We begin by presenting our master script, `SINDy_Analysis.m`. We constructed this script to ease the process of attempting to navigate the plethora of modeling conditions we wish to explore. Rather than have a single script that has the ability to run every possible scenario, we utilize the master script to allow the user to preset the desired data as well as the number of modes. The script will then run an adapted version of the inherited code that meets the preset conditions.

5.3.2 Uploading Data

In the adapted scripts, the first change that was made was to trim the data that was uploaded for analysis. Minor modifications were made to achieve this result. Our reasoning behind this modification was to ensure folds generated by the cross validation process were sequential and also belonged to the same trial. Our modified version of the script is titled `loadAlex.m`.

5.3.3 Data Smoothing

In the cases of the open and closed ends data sets, we interpolated 20 points between each data observation and then applied cubic splines in an attempt to smooth the data. This was done using two scripts, `Dat_spline.m` and `SplineSync.m`. The first script generated the 20 observations and then applied cubic splines to the data. The second script retrieved the new observations that corresponded to our original data so that the size of our data set was not compromised.

5.3.4 Derivatives

The original script was written to calculate the derivatives of the data using a fourth order central differencing scheme.

$$\frac{dx_i}{dt}(t_j) = \frac{-x_i(t_{j+2}) + 8x_i(t_{j+1}) - 8x_i(t_{j-1}) + x_i(t_{j-2})}{12\Delta} \quad (5.1)$$

Our addition to the script was to separate the test data from the training set. We also calculated A in this section as well as its derivatives. We note here that our derivatives for A were calculated using the theoretical approach and not through a fourth order differencing scheme directly performed on A (i.e. $\dot{A} = 2x\dot{x} + 2y\dot{y}$).

5.3.5 Θ and $\dot{\theta}$

Here we had to make some minor alterations to the code so that the construction of Θ would be representative of our transformed modes versus the original data. We also had to calculate $\dot{\theta}$ at this point for our test data.

5.3.6 Scaling

After saving a copy of our data, we then calculate the appropriate scaling to be applied to each mode. Rather than using a standardized approach, we divided all values by their absolute maximum to bring them between -1 and 1. This method was chosen due to the heavy tails that are present in the data.

5.3.7 Transforming the Energy Constraints

The code we inherited was capable of generating the energy constraints for any number of modes. We added a section to generate the constraints necessary to extend A as described in the previous chapter. We then wrote `Atilde_2ndHarmonic.m` to calculate the transformation matrix, \tilde{A} . Our code is capable of handling data sets containing up to six modes.

5.3.8 Calculating Regression Coefficients

At this point the code obtains the regression coefficients for our transformed data. While minor changes were made to labeling our variables and the code was checked for errors, any contributions we made to this section are not significant.

5.3.9 Obtaining Models

To obtain our final models, we stripped necessary components of an inherited script, `explore_regr.m`. Models were obtained both in terms of our transformed modes as well as our original modes. Our original mode model was used to obtain our estimates of \hat{A} and these estimates were used to obtain our residuals.

Chapter 6

Results

In this chapter, we will present the results of our research. The chosen models are all associated with the ‘elbow’ in the mean square error vs. $\log(\lambda)$ plots (as described in 5.1 on page 33). All models examined, regardless of size, can be found in [D](#). Furthermore, we assess the models’ performance by looking at its capture of variance. By dividing the variation of our fitted values and comparing that with the observed variation in the data for each of the modes, we can determine how much of the variability the model accounts. A table containing all these values can be found in [C](#). Before we begin, we note that we do not use an ODE solver to evaluate the modes as this approach has difficulty obtaining solutions that converge. Instead we obtain our estimates of the time derivatives at all observations by calculating the product between our model equations and Θ . One last note is that even though we talk about transforming our harmonic oscillators into a new mode A , our results with regards to its time derivative will be presented in terms of \dot{A}/A .

6.1 Cylinder Data

Testing the performance of our method with a well known example, the cylinder data yields results akin to what we expect to see in cylindrical bluff body vortex shedding. The model generated is sparse in nature and contains the low-order terms we see in (Brunton, Proctor, and Kutz, [2016](#)). We omit the model from this paper, but the results can be obtained with the provided code and data.

6.2 Closed Ends Data

Before we show are models, we would like to note that when $(x^2 + y^2)$ appears in any of our equations, this is simply our newly transformed mode, A . We will be sure to enclose the equality in parenthesis like so, $(x^2 + y^2)$, to make it easier for the reader remember this relationship.

6.2.1 2 mode analysis

Model

$$\frac{\dot{A}}{A} = -5.08 \times 10^4 \cdot z$$

$$\dot{z} = -0.271 - 91.3 \cdot z + 2.55 \times 10^4 \cdot (x^2 + y^2)$$

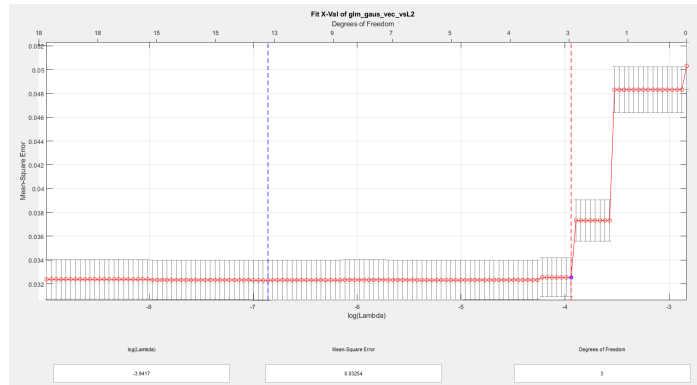


FIGURE 6.1. Mean square error vs $\log(\lambda)$ plot for Closed Ends data (2-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error. The selected model has a $\log(\lambda)$ value of -3.9417 and is on the red dotted line.

In terms of our transformed modes, this 2-mode model contains our mode representing the transformation of our most energetic harmonic oscillators (A or $(x^2 + y^2)$) and our shift mode (z) for the closed ends data. We see in 6.1 that all the models captured within one standard error of the minimal model have similar mean squared errors. We therefore choose the most parsimonious one which has a log penalty parameter of -3.9417.

Our method generates a sparse model. With the exclusion of a constant term in \dot{A}/A , we notice that the system has the same elements as cylindrical bluff body example.

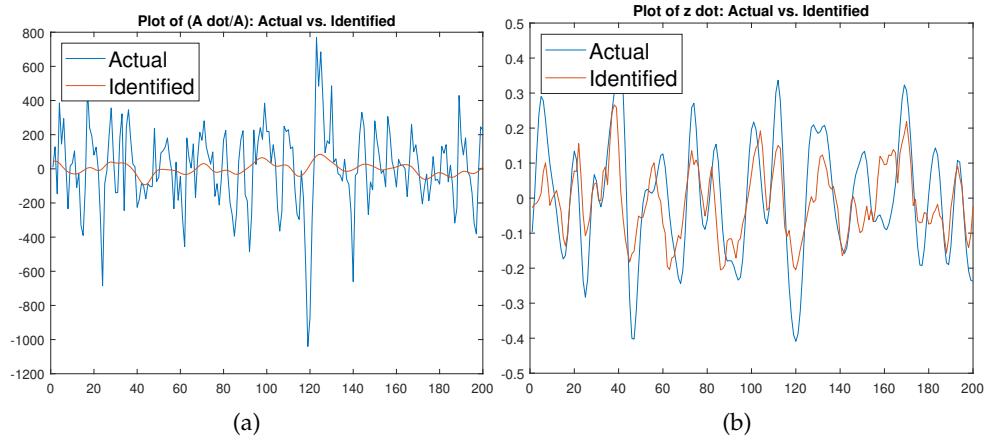


FIGURE 6.2. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{A}/A (left) and \dot{z} (right) for the Closed Ends data.

Looking at the performance of the model with respect to the data, we notice that it only manages to capture 5.69% of the variation in \dot{A}/A . It does markedly better with \dot{z} , capturing 39.34% of the variation. Our model leaves a lot of room for improvement and may benefit from the addition of more modes. We will next examine what happens when we add in the flapping mode.

6.2.2 3 mode analysis

Model

$$\frac{\dot{A}}{A} = -4.79 \times 10^4 \cdot z - 1.22 \times 10^7 \cdot xy - 3.13 \times 10^7 \cdot xw$$

$$\dot{z} = -0.261 - 84.5 \cdot z + 2.39 \times 10^4 \cdot (x^2 + y^2) - 1.20 \times 10^4 \cdot xw - 1.60 \times 10^4 \cdot yw \\ + 5.75 \times 10^3 \cdot w^2$$

$$\dot{w} = 16.8 \cdot x + 1.20 \times 10^4 \cdot xz + 1.60 \times 10^4 \cdot yz - 5.75 \times 10^3 \cdot zw$$

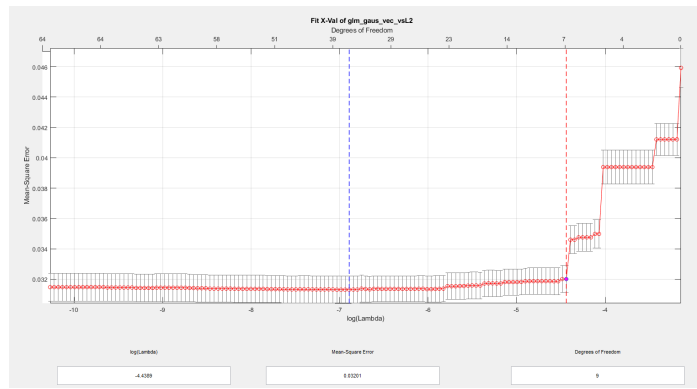


FIGURE 6.3. Mean square error vs $\log(\lambda)$ plot for Closed Ends data (3-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). The selected model has a $\log(\lambda)$ value of -4.4389 and is on the red dotted line.

In terms of our transformed modes, this 3-mode model contains our mode representing the transformation of our most energetic harmonic oscillators (A or $(x^2 + y^2)$), our shift mode (z) and our flapping mode (w) for the closed ends data. Once again, looking at 6.3, we see little difference between all the models within one standard error of our minimal model and therefore we once again choose the most parsimonious. This model has a log penalty parameter of -4.4389.

When we consider our transformed model of 3 modes, the addition of the flapping mode gives us more complex behaviour than we saw before. We see two added terms in \dot{A}/A that will be third order terms when our transformed mode, A , is expanded into its original components of x and y . Even though the flapping modes only adds two terms to \dot{A}/A , it significantly improves its capture of the underlying

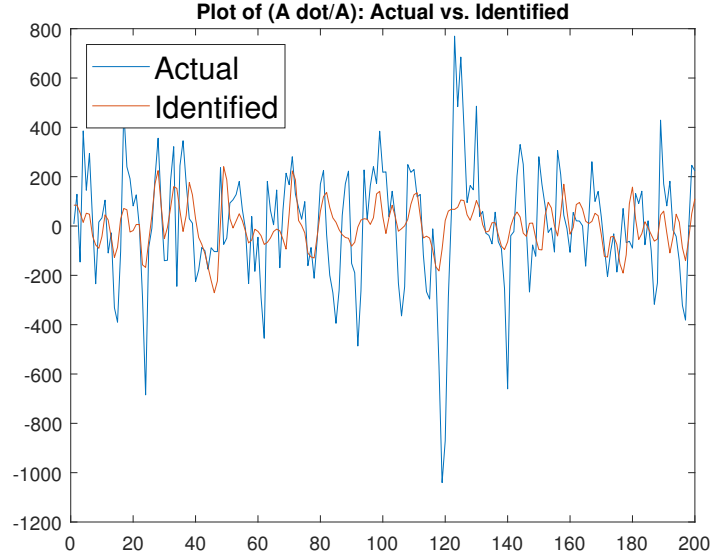


FIGURE 6.4. Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the Closed Ends data.

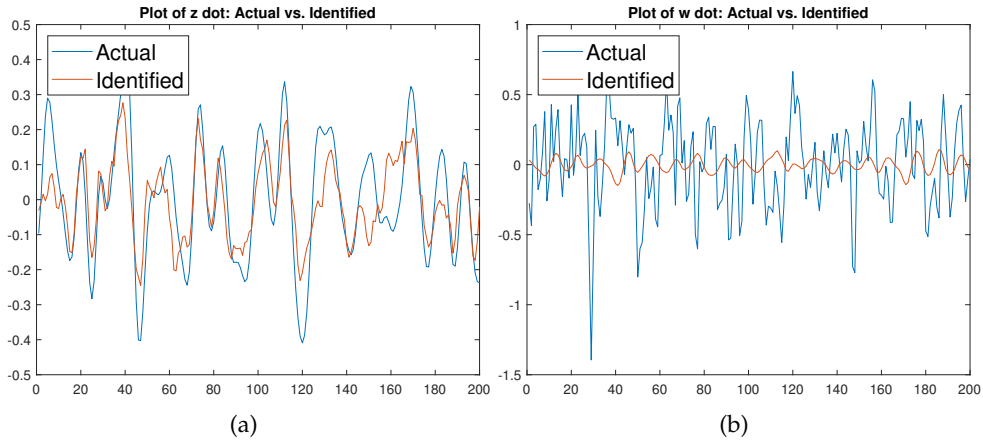


FIGURE 6.5. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the Closed Ends data.

dynamics. With the addition of the flapping mode, we can now explain 13.11% of the variation of \dot{A}/A . Contrarily, it adds three terms to our equation for \dot{z} but only adds slight improvement, capturing 44.8% of the variation. Unfortunately, the model does not perform as well with regard to \dot{w} , only capturing 0.02% of the variation. We may be able to improve on this with the addition of the second most energetic harmonic oscillators.

6.2.3 5 mode analysis

In terms of our transformed modes, this 5-mode model includes our mode representing the transformation of our most energetic harmonic oscillators (A or $(x^2 + y^2)$), our shift mode (z), flapping mode (w) and our second pair of harmonic oscillators (u and v) for the closed ends data. Once again, looking at 6.3, we see little difference between all the models within one standard error of our minimal model and therefore we once again choose the most parsimonious model. This is the model associated with a log penalty parameter of -5.5669. Due to its size, it can be viewed in D.1.3.

Unfortunately, we notice that our model is no longer as sparse as we'd like. It has many third order terms, but fortunately it does not contain any in \dot{A}/A which would end up being fourth order terms when A is broken into its original components of x and y . While we do see marked improvement in the modeling of \dot{z} with 52.07% of the variation captured, we see little improvement in the modeling of \dot{A}/A , capturing 13.6% of the variation. Strangely, we perform worse on our capture of variation for \dot{w} , with -3.16% captured. With regards to \dot{u} the model performs very well with 45.27% of the variation captured. Our capture of \dot{v} is slightly better than \dot{A}/A at 19.32%. With the improvements we see it is hard to say whether the addition of so many terms and equation is merited. As this paper mainly focuses on the statistical approach to modeling the data, we will leave the interpretation of these results to experts in the field of fluid dynamics.

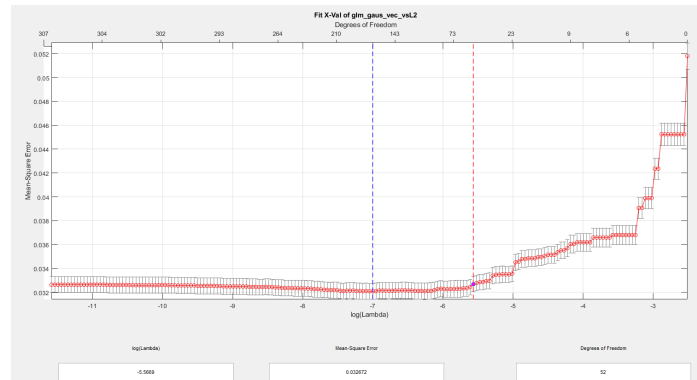


FIGURE 6.6. Mean square error vs $\log(\lambda)$ plot for Closed Ends data (5-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). The selected model has a $\log(\lambda)$ value of -5.5669 and is on the red dotted line.

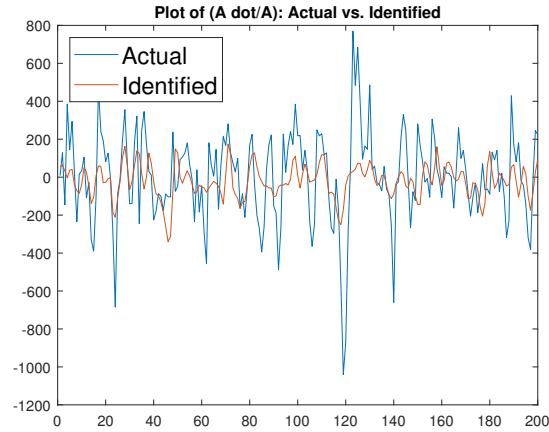


FIGURE 6.7. Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the Closed Ends data.

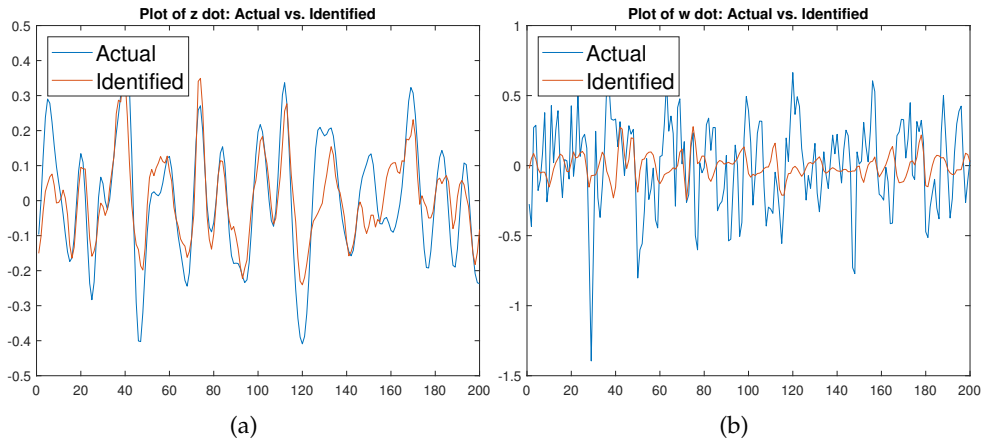


FIGURE 6.8. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the Closed Ends data.

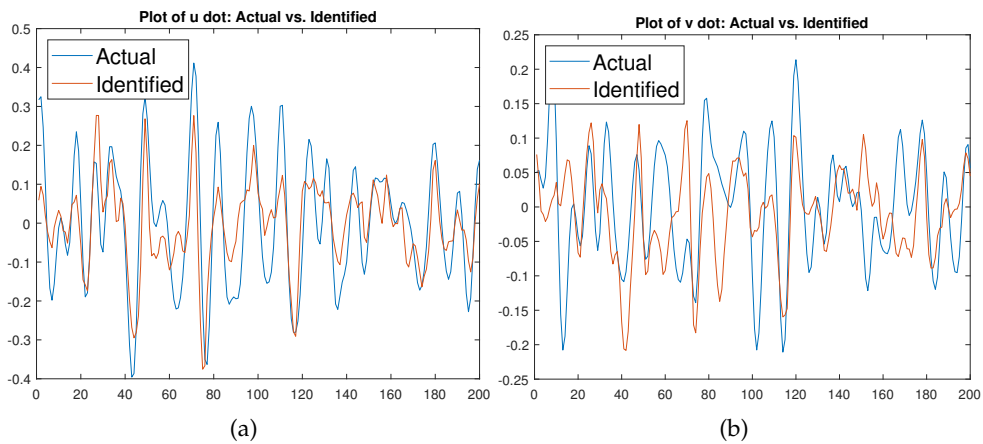


FIGURE 6.9. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{u} (left) and \dot{v} (right) for the Closed Ends data.

6.3 Open Ends Data

6.3.1 2 mode analysis

Transformed Model (1SE)

$$\frac{\dot{A}}{A} = -6.62 \times 10^3 \cdot z - 2.28 \times 10^7 \cdot xy$$

$$\dot{z} = -0.0214 + 3.31 \times 10^3 \cdot (x^2 + y^2)$$

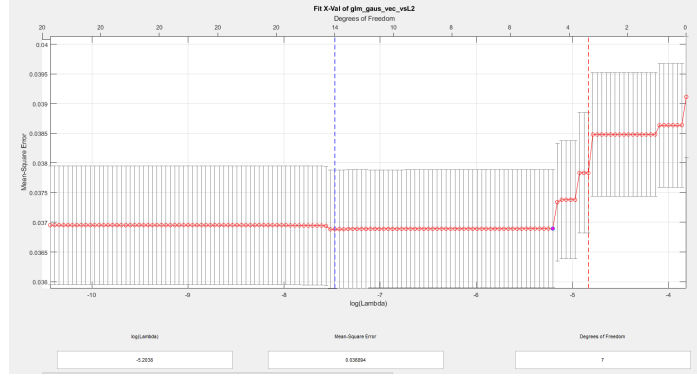


FIGURE 6.10. Mean square error vs $\log(\lambda)$ plot for Open Ends data (2-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). We notice that the 'elbow' occurs between the read a blue lines and so we choose the point were the mean square error levels off. The $\log(\lambda)$ value is -5.2058.

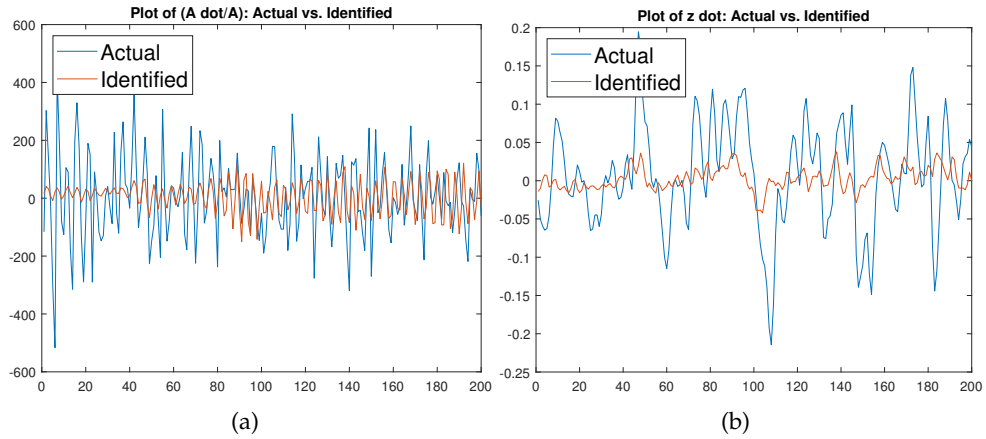


FIGURE 6.11. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{A}/A (left) and \dot{z} (right) for the Open Ends data.

In terms of our transformed modes, our 2-mode model contains the mode representing the transformation of our most energetic harmonic oscillators (A or $(x^2 + y^2)$) and

our shift mode (z) for the open ends data. If we look at 6.10, we see a greater disparity between models within one standard error of our minimal model. In this case, we do not choose the most parsimonious model as it is not associated with the 'elbow'. Instead we pick the point where the mean square error begins to level out. Our chosen model is the one that has a log penalty parameter of -5.2058.

With regard to our 2-mode model, it is reminiscent of cylindrical bluff body vortex shedding, but there is an added xy term associated with \dot{A}/A (which ends up being a third order term when we represent A in terms of x and y). Also, there is no z term associated with \dot{z} . In 6.11, we see that the performance of our variance is surprisingly better than the closed ends model for \dot{A}/A with 9.63% captured, but is dramatically worse for \dot{z} with only 4.41% captured. It is our hope that we find more explanation for the system with the inclusion of more modes. Next we look at the model that includes the flapping mode.

6.3.2 3 mode analysis

Model

$$\frac{\dot{A}}{A} = (-8.80 \times 10^3) \cdot z - (2.42 \times 10^7) \cdot xy + (3.23 \times 10^7) \cdot xw + (3.95 \times 10^7) \cdot yw - (6.17 \times 10^7) \cdot w^2$$

$$\dot{z} = -0.0275 + (4.40 \times 10^3) \cdot (x^2 + y^2) - (1.98 \times 10^4) \cdot xw + (1.47 \times 10^4) \cdot yw + (7.16 \times 10^5) \cdot x^2z + (4.54 \times 10^6) \cdot xzw - (1.89 \times 10^6) \cdot y^2z - (6.87 \times 10^6) \cdot yzw$$

$$\dot{w} = (-4.89 \times 10^1) \cdot x + (1.49 \times 10^2) \cdot y + (1.98 \times 10^4) \cdot xz - (1.47 \times 10^4) \cdot yz + (6.10 \times 10^6) \cdot x^3 - (1.62 \times 10^7) \cdot x^2y + (1.09 \times 10^7) \cdot x^2w - (1.57 \times 10^7) \cdot xyw - (1.61 \times 10^7) \cdot y^3 - (4.49 \times 10^5) \cdot y^2w - (4.48 \times 10^7) \cdot yz^2$$

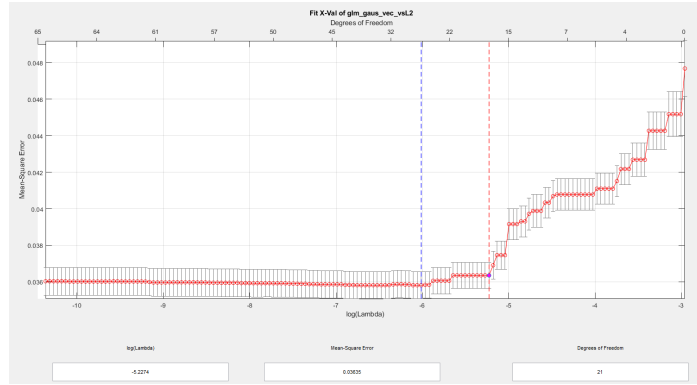


FIGURE 6.12. Mean square error vs $\log(\lambda)$ plot for Open Ends data (3-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). The model select is on the dotted red line and has a $\log(\lambda)$ value of -5.2274.

In terms of our transformed modes, this 3-mode model contains our mode representing the transformation of our most energetic harmonic oscillators (A or $(x^2 + y^2)$), our shift mode (z) and our flapping mode (w) for the open ends data. While there is a smaller range of disparity between models within one standard error of the minimal model (6.12), the 'elbow' is much harder to identify. We do however choose the most parsimonious model within one standard error of the minimal model. It has an associated log penalty of -5.2274.

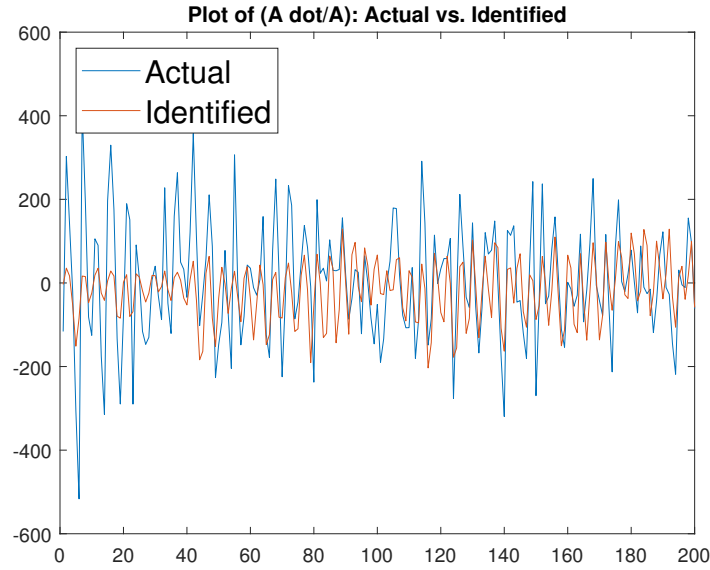


FIGURE 6.13. Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the open ends data.

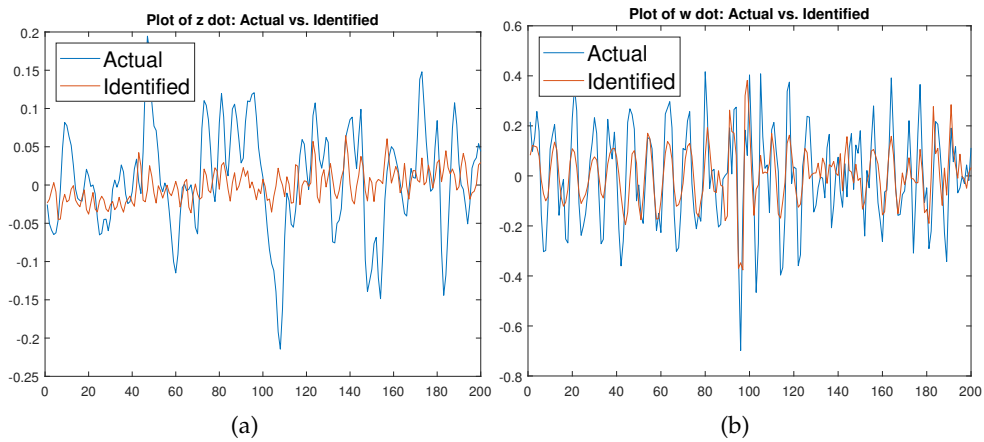


FIGURE 6.14. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the open ends data.

Our 3-mode model offers more insight into the system, but still seems to miss some key dynamics. It seems that the flapping mode is an important variable with regards to \dot{A}/A because its inclusion improves the variation capture to 15.81%. This comes at a cost of 3 added terms to the \dot{A}/A equation. It doesn't fair so well with \dot{z} as we see our variance capture decrease to 2.01% even with six added terms. Astonishingly \dot{w} 's variance capture is significantly better than in the closed ends data at 47.94%. The complexity of the equation may be justifiable because of this performance. We will now look at the model with the second harmonic oscillators added.

6.3.3 5 mode analysis

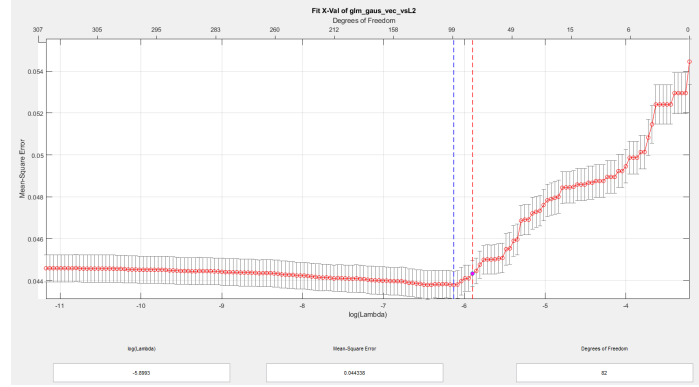


FIGURE 6.15. Mean square error vs $\log(\lambda)$ plot for Open Ends data (3-modes). The red dotted line signifies the model that is most parsimonious within 1 standard error of the model with the minimal standard error (blue line). Our model of choice is on the red line and has an associated $\log(\lambda)$ of -5.8893

In terms of our transformed modes, this 5-mode model includes our mode representing the transformation of our most energetic harmonic oscillators (A or $(x^2 + y^2)$), our shift mode (z), flapping mode (w) and our second pair of harmonic oscillators (u and v) for the open ends data. With 6.15, we see very few models that fall in the range of one standard error of the minimal model and while we chose the most parsimonious model in this case, it may be argued its not the 'elbow'. We also see some interesting behaviour in the mean square error plot that we haven't seen before. After reaching a minimum, the mean square error appears to rise as the models get more complex rather than level off. With all of that in mind, we remain content with our choice of the most parsimonious model which has an associated log penalty parameter of

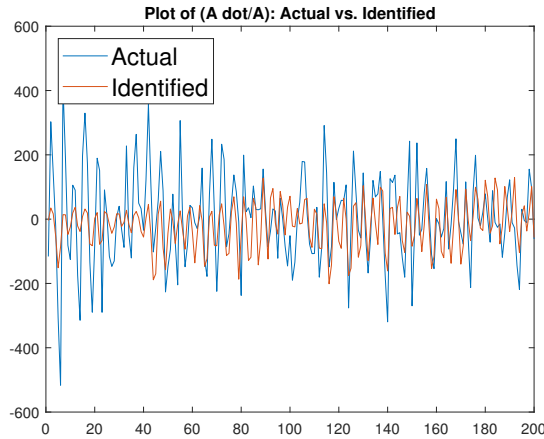


FIGURE 6.16. Observed values of \dot{A}/A (blue line) compared against the fitted values from the model (red line) for the Open Ends data.

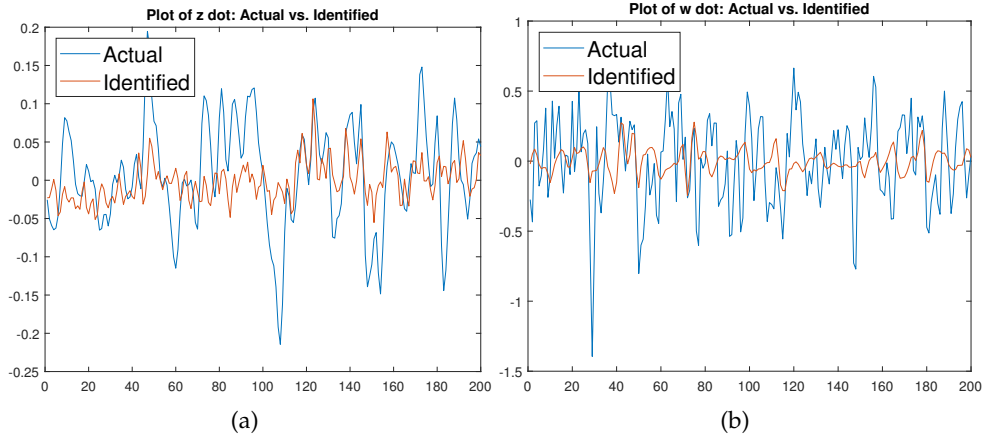


FIGURE 6.17. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{z} (left) and \dot{w} (right) for the Open Ends data.

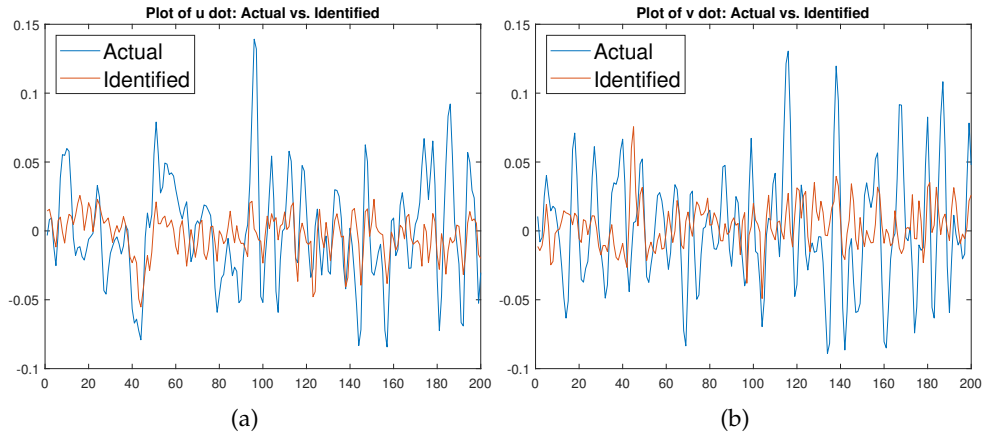


FIGURE 6.18. Observed values (blue line) compared against the fitted values from the model (red line) of \dot{u} (left) and \dot{v} (right) for the Open Ends data.

-5.8893. Due to its size, it can be viewed in [D.2.3](#).

Once again, with our 5-mode model, the model obtained is not as sparse as we would like. The inclusion of the second harmonic oscillators results in a complicated model that doesn't add much insight into the dynamics of the system that wasn't gained from the 3-mode model. While we see limited improvement in the variance capture of \dot{z} at 9.36% and \dot{w} at 49.23% the other derivatives don't do as well. With \dot{u} we only get 6.4%, \dot{v} at 8.05%, and lastly we see a decrease in \dot{A}/A to 15.65%. In this case, it doesn't appear that the 5-mode model performs well compared to the 3-mode model.

Chapter 7

Conclusions

7.1 Conclusions

In summary, our approach looked at making adaptations to the newly developed SINDy method to see if it was effective at identifying a representative low-dimensional system of equations in the field of fluid dynamics. We explored the use of LASSO in order to apply cyclical coordinate descent and path seeking algorithms in our search for the governing equation of vortex shedding in the wake of a bluff body. We then elaborated on sparse Galerkin regression and showed that it was possible to apply the constraints before obtaining the model. With these adjustments, we looked at three different data sets, one being the classic cylindrical example presented in (Brunton, Proctor, and Kutz, 2016), and the other two coming from new research discussed in (Braun, Agrey, and Martinuzzi, 2020). Upon running adapted MATLAB code, we analyzed output generated and established 2, 3 and 5-mode models that contained our mode representing the transformation of our most energetic harmonic oscillators, our shift mode, flapping mode and our second pair of harmonic oscillators.

While we are uncertain how accurate our model is in its portrayal of the dynamics of vortex shedding in the wakes of open and closed ended flat plates, we do feel that our models do have some of the key features present in the system. Our assessment of variation capture tells us that some of the components of the system are better understood than others. For example, in the closed ends data, \dot{z} is explained exceptionally well with our 5-mode model, but all of the models that we obtained failed to give any reasonable explanation of \dot{w} .

Even with our models, there is still some understanding of the dynamics of the system that are not captured. Perhaps our approach would benefit from the use of multivariate time series analysis to model the residual behaviour of the generating equations. While such an approach may shed more light on the systems we are examining, we save it for future research.

What we do glean from our models is that the closed ends data at its very base is similar to vortex shedding in the wake of a cylinder. The model provides some understanding of the system, but not the complete picture. With the addition of the flapping mode we see that it has more dynamic features than cylindrical bluff body vortex shedding; we gain better understanding of the system. Finally, the addition of the second most energetic harmonic oscillators adds more understanding to some of the more chaotic components of the system, but little understanding to \dot{A}/A .

With regards to the open ends data, the system does reasonably well with our approach, but the 5-mode model we obtain does not inspire confidence. The inclusion of the second most energetic harmonic oscillators adds little understanding of the system as a whole and using Occam's razor, we lean towards the 3-mode model as better explaining the system. While the complexity of the equations in the 5-mode model may be merited, the amount of confusion added compared with the understanding gained leaves us with opting for the simpler choice.

Even with the success we have found, our approach has limitations to its application. Firstly, our conclusions are subjective due to the nature of our approach. We are unable to attribute objective measures such as misclassification rates since we are dealing with the identification of an unknown, time-dependent stochastic process. Another drawback of our approach is the possibility that the true model has important variables with small coefficients. If this is the case, the LASSO method will likely fail to capture these attributes, especially if noise in the system generates variable coefficients larger than those we wish to identify. Furthermore, if there are highly correlated variables in the true model, LASSO is likely to exclude all but one of them due to the process by which LASSO arrives at a solution.

Even with our limitations, our constrained LASSO approach in the realm of SINDy research presents opportunity for analysis of real data. The ability to extract key features of a system when the data is noisy is beneficial even as a starting point in

variable selection when attempting to identify the true underlying system. Our approach is not limited to fluid mechanics, but is applicable to any system where a priori knowledge of constraints exists and where a transformation of the data can be justified. In our case, the a priori understanding of constraints that result from our POD transformation allows for low-order models to be generated that can provide useful insight into a system's behaviour without requiring cumbersome software that comes with a high computational cost.

Appendix A

Matrices and Vectors

In this section of the appendix, many of the matrices have only a handful of non-zero elements. In order to make these entries easier to see, we will highlight them in red.

A.1 Galerkin Constraint Matrix

$$E_{10 \times 60} = \begin{array}{c} \begin{array}{cccccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & \dots & 20 & \dots \end{array} \\ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcolor{red}{1} & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \end{array} \right] \end{array}$$

$$\begin{array}{c}
 \dots \quad 21 \quad 22 \quad 23 \quad 24 \quad 25 \quad 26 \quad 27 \quad 28 \quad 29 \quad 30 \quad 31 \quad \dots \quad 40 \quad \dots \\
 \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \left[\begin{array}{cccccccccccccccc}
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & \dots & 0 & \dots \\
 \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & \dots & 0 & \dots
 \end{array} \right]
 \end{array}$$

$$\begin{array}{c}
 41 \quad 42 \quad 43 \quad 44 \quad 45 \quad 46 \quad 47 \quad 48 \quad 49 \quad 50 \quad 51 \quad \dots \quad 60 \\
 \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \left[\begin{array}{cccccccccccc}
 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & 0 & \dots & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \color{red}{1} & 0 & 0 & 0 & \dots & 0
 \end{array} \right]
 \end{array}$$

$$\begin{aligned} & \dots \ 0 \ \dot{\theta} \ 0 \ \dots \\ & \dots \ 0]^T \end{aligned}$$

A.4 Vectors $x_{20 \times 1}^{Id}$ and $y_{20 \times 1}^{Id}$

A.4.1 Construction of $x_{20 \times 1}^{Id}$

Elements containing x in \mathbf{X}^{P_0}

$$[0]$$

Elements containing x in \mathbf{X}^{P_1}

$$[1 \ 0 \ 0]$$

Elements containing x in \mathbf{X}^{P_2}

$$[1 \ 1 \ 1 \ 0 \ 0 \ 0]$$

Elements containing x in \mathbf{X}^{P_3}

$$[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]$$

Concatenation and transposition of the above vectors gives us

$$x^{Id} = [0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$$

A.4.2 Construction of $y_{20 \times 1}^{Id}$

Elements containing y in \mathbf{X}^{P_0}

$$[0]$$

Elements containing y in \mathbf{X}^{P_1}

$$[0 \ 1 \ 0]$$

Elements containing y in \mathbf{X}^{P_2}

$$[0 \ 1 \ 0 \ 1 \ 1 \ 0]$$

A.5.2 $\tilde{A}^y_{20 \times 20}$

[illegible]

Appendix B

Calculations

B.1 Proposition Calculations

B.1.1 Calculations for β^x

$$\tilde{A}^x \cdot \beta^{\tilde{A}^*}(\cdot)_{(1:20)} + \vec{\theta}_{(1:20,1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \cdot \begin{bmatrix} \tilde{\beta}^{A_1} \\ \tilde{\beta}^{A_2} \\ \tilde{\beta}^{A_3} \\ \tilde{\beta}^{A_4} \\ \tilde{\beta}^{A_5} \\ \tilde{\beta}^{A_6} \\ \tilde{\beta}^{A_7} \\ \tilde{\beta}^{A_8} \\ \tilde{\beta}^{A_9} \\ \tilde{\beta}^{A_{10}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\dot{\theta} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\tilde{\beta}^{A_1}}{2} \\ -\dot{\theta} \\ 0 \\ \frac{\tilde{\beta}^{A_2}}{2} \\ \frac{\tilde{\beta}^{A_3}}{2} \\ \frac{\tilde{\beta}^{A_4}}{2} \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\tilde{\beta}^{A_5}}{2} \\ \frac{\tilde{\beta}^{A_6}}{2} \\ \frac{\tilde{\beta}^{A_7}}{2} \\ \frac{\tilde{\beta}^{A_8}}{2} \\ \frac{\tilde{\beta}^{A_9}}{2} \\ \frac{\tilde{\beta}^{A_{10}}}{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

B.1.2 Calculations for $\tilde{\beta}^y$

$$\begin{aligned}
\tilde{A}^y \cdot \beta^{\tilde{A}*}(\cdot)_{(21:40)} + \vec{\theta}_{(21:40,1)} = & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \cdot \begin{bmatrix} \tilde{\beta}^A_1 \\ \tilde{\beta}^A_2 \\ \tilde{\beta}^A_3 \\ \tilde{\beta}^A_4 \\ \tilde{\beta}^A_5 \\ \tilde{\beta}^A_6 \\ \tilde{\beta}^A_7 \\ \tilde{\beta}^A_8 \\ \tilde{\beta}^A_9 \\ \tilde{\beta}^A_{10} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\theta} \\ \frac{\tilde{\beta}^A_1}{2} \\ 0 \\ 0 \\ \frac{\tilde{\beta}^A_2}{2} \\ 0 \\ \frac{\tilde{\beta}^A_3}{2} \\ \frac{\tilde{\beta}^A_4}{2} \\ 0 \\ 0 \\ \frac{\tilde{\beta}^A_5}{2} \\ 0 \\ \frac{\tilde{\beta}^A_6}{2} \\ \frac{\tilde{\beta}^A_7}{2} \\ 0 \\ \frac{\tilde{\beta}^A_8}{2} \\ \frac{\tilde{\beta}^A_9}{2} \\ \frac{\tilde{\beta}^A_{10}}{2} \\ 0 \end{bmatrix}
\end{aligned}$$

Appendix C

Variation Table

Geometry	Modes Present	% of variation explained				
		\dot{A}/A	\dot{z}	\dot{w}	\dot{u}	\dot{v}
Closed	A, z	5.69	39.34	N/A	N/A	N/A
Closed	A, z, w	13.11	44.8	0.02	N/A	N/A
Closed	A, z, w, u, v	13.6	52.07	-3.16	45.27	19.32
Open	A, z	9.63	4.41	N/A	N/A	N/A
Open	A, z, w	15.81	2.07	47.94	N/A	N/A
Open	A, z, w, u, v	15.65	9.36	49.23	6.4	8.05

FIGURE C.1. Table of Variation accounted for under different geometries and models. Calculation to obtain values is $1 - (\text{residual variation})/(\text{observed variation})$

Appendix D

Models

D.1 Closed Ends Data

D.1.1 2 Mode Analysis

Model ($\lambda = 0.0194$)

	AdotOverA	Zdot
1	0	-2.71e-01
X	0	0
Y	0	0
Z	-5.08e04	-9.13e01
X*X	0	2.54e04
X*Y	0	0
X*Z	0	0
Y*Y	0	2.54e04
Y*Z	0	0
Z*Z	0	0
X*X*X	0	0
X*X*Y	0	0
X*X*Z	0	0
X*Y*Y	0	0
X*Y*Z	0	0
X*Z*Z	0	0
Y*Y*Y	0	0
Y*Y*Z	0	0

Y^*Z^*Z	0	0
Z^*Z^*Z	0	0

D.1.2 3 Mode Analysis

Model ($\lambda = 0.0118$)

	AdotOverA	Zdot	Wdot
1	0	-2.61e-01	0
X	0	0	1.68e01
Y	0	0	0
Z	-4.79e04	-8.45e01	0
W	0	0	0
X*X	0	2.39e04	0
X*Y	-1.22e07	0	0
X*Z	0	0	1.20e04
X*W	-3.13e07	-1.20e04	0
Y*Y	0	2.39e04	0
Y*Z	0	0	1.60e04
Y*W	0	-1.60e04	0
Z*Z	0	0	0
Z*W	0	0	-5.75e03
W*W	0	5.75e03	0
X*X*X	0	0	0
X*X*Y	0	0	0
X*X*Z	0	0	0
X*X*W	0	0	0
X*Y*Y	0	0	0
X*Y*Z	0	0	0
X*Y*W	0	0	0
X*Z*Z	0	0	0
X*Z*W	0	0	0
X*W*W	0	0	0

Y*Y*Y	0	0	0
Y*Y*Z	0	0	0
Y*Y*W	0	0	0
Y*Z*Z	0	0	0
Y*Z*W	0	0	0
Y*W*W	0	0	0
Z*Z*Z	0	0	0
Z*Z*W	0	0	0
Z*W*W	0	0	0
W*W*W	0	0	0

D.1.3 5 Mode Analysis

Model ($\lambda = 0.0038$)

	AdotOverA	Zdot	Wdot	Udot	Vdot
1	0	-1.53e-01	0	0	0
X	0	0	9.71e00	0	0
Y	0	0	8.02e00	0	1.33e00
Z	-2.92e04	0	0	-6.85e01	-1.10e01
W	0	0	0	0	0
U	0	1.37e02	0	-1.76e01	-7.65e01
V	0	8.95e01	0	6.11e01	-2.17e01
X*X	0	1.46e04	0	0	0
X*Y	-1.21e07	0	0	0	0
X*Z	0	0	0	0	0
X*W	-3.24e07	0	0	-3.09e04	3.64e03
X*U	0	0	3.09e04	0	0
X*V	0	0	-3.64e03	0	0
Y*Y	0	1.46e04	0	0	0
Y*Z	0	0	8.60e03	0	0
Y*W	0	-8.60e03	0	2.21e04	1.71e04
Y*U	0	0	-2.21e04	0	0

Y*V	0	0	-1.71e04	0	0
Z*Z	0	0	0	-9.30e03	0
Z*W	0	0	3.90e03	0	0
Z*U	0	9.30e03	0	0	0
Z*V	0	0	0	4.72e04	3.76e04
W*W	-1.39e07	-3.90e03	0	0	0
W*U	0	0	0	0	0
W*V	0	0	0	0	0
U*U	-6.11e07	0	0	0	0
U*V	0	-4.72e04	0	0	0
V*V	0	-3.76e04	0	0	0
X*X*X	0	0	0	0	0
X*X*Y	0	0	0	0	0
X*X*Z	0	0	0	0	0
X*X*W	0	0	0	0	0
X*X*U	0	0	0	0	0
X*X*V	0	0	0	0	0
X*Y*Y	0	0	0	0	0
X*Y*Z	0	0	0	0	0
X*Y*W	0	0	0	0	0
X*Y*U	0	0	0	0	0
X*Y*V	0	0	0	0	3.21e06
X*Z*Z	0	0	0	0	0
X*Z*W	0	0	0	8.35e06	0
X*Z*U	0	0	0	0	0
X*Z*V	0	0	4.16e07	0	0
X*W*W	0	0	1.05e07	0	0
X*W*U	0	-1.22e07	0	0	0
X*W*V	0	0	0	0	0
X*U*U	0	0	0	0	0
X*U*V	0	0	0	0	0

X*V*V	0	0	0	-1.15e07	0
Y*Y*Y	0	0	0	0	0
Y*Y*Z	0	-6.05e06	0	0	0
Y*Y*W	0	0	7.15e06	0	0
Y*Y*U	0	0	0	0	-5.84e05
Y*Y*V	0	0	0	0	0
Y*Z*Z	0	0	0	0	0
Y*Z*W	0	0	0	-1.12e07	0
Y*Z*U	0	0	-2.47e07	0	0
Y*Z*V	0	0	6.43e07	0	6.91e06
Y*W*W	0	0	0	0	0
Y*W*U	0	-1.22e07	0	0	0
Y*W*V	0	1.40e07	0	-6.33e06	0
Y*U*U	0	0	0	0	0
Y*U*V	0	1.66e07	0	0	0
Y*V*V	0	0	0	0	0
Z*Z*Z	0	0	0	0	0
Z*Z*W	0	0	0	0	0
Z*Z*U	0	0	0	0	0
Z*Z*V	0	0	0	0	0
Z*W*W	0	0	0	0	-8.24e06
Z*W*U	0	0	0	0	0
Z*W*V	0	0	0	0	0
Z*U*U	0	0	0	0	-1.27e07
Z*U*V	0	0	0	0	-5.70e07
Z*V*V	0	-6.77e07	0	0	0
W*W*W	0	0	0	0	0
W*W*U	0	0	0	0	0
W*W*V	0	0	0	0	0
W*U*U	0	0	0	0	-1.44e07
W*U*V	0	0	-7.56e07	0	0

W*V*V	0	0	-9.79e07	0	0
U*U*U	0	0	0	-1.76e07	0
U*U*V	0	0	0	4.97e07	0
U*V*V	0	0	0	0	0
V*V*V	0	0	0	0	0

D.2 Open Ends Data

D.2.1 2 Mode Analysis

Model ($\lambda = 0.0080$)

	AdotOverA	Zdot
1	0	-4.11e-02
X	0	0
Y	0	-1.18e00
Z	-1.32e04	-1.40e01
X*X	0	6.60e03
X*Y	-2.28e07	0
X*Z	0	0
Y*Y	0	6.60e03
Y*Z	0	0
Z*Z	0	0
X*X*X	0	0
X*X*Y	0	0
X*X*Z	0	-1.03e+06
X*Y*Y	0	0
X*Y*Z	0	-1.18e+06
X*Z*Z	0	0
Y*Y*Y	0	0
Y*Y*Z	0	0
Y*Z*Z	0	0
Z*Z*Z	0	0

D.2.2 3 Mode Analysis

Model ($\lambda = 0.0054$)

	AdotOverA	Zdot	Wdot
1	0	-2.75e-02	0
X	0	0	-4.89e01
Y	0	0	1.49e02
Z	-8.80e03	0	0
W	0	0	0
X*X	0	4.40e03	0
X*Y	-2.42e07	0	0
X*Z	0	0	1.98e04
X*W	3.23e07	-1.98e04	0
Y*Y	0	4.40e03	0
Y*Z	0	0	-1.47e04
Y*W	3.95e07	1.47e04	0
Z*Z	0	0	0
Z*W	0	0	0
W*W	-6.17e07	0	0
X*X*X	0	0	6.10e06
X*X*Y	0	0	-1.62e07
X*X*Z	0	7.16e05	0
X*X*W	0	0	1.09e07
X*Y*Y	0	0	0
X*Y*Z	0	0	0
X*Y*W	0	0	-1.57e07
X*Z*Z	0	0	0
X*Z*W	0	4.54e06	0
X*W*W	0	0	0
Y*Y*Y	0	0	-1.61e07
Y*Y*Z	0	-1.89e06	0
Y*Y*W	0	0	-4.49e05

Y^*Z^*Z	0	0	-4.48e07
Y^*Z^*W	0	-6.87e06	0
Y^*W^*W	0	0	0
Z^*Z^*Z	0	0	0
Z^*Z^*W	0	0	0
Z^*W^*W	0	0	0
W^*W^*W	0	0	0

D.2.3 5 Mode Analysis

Model ($\lambda = 0.0027$)

	AdotOverA	Zdot	Wdot	Udot	Vdot
1	0	-2.46e-02	0	2.69e-02	-2.35e-02
X	0	0	-4.44e01	0	0
Y	0	0	1.42e02	0	0
Z	-7.15e03	0	0	0	0
W	0	0	0	3.96e00	0
U	7.41e03	-9.14e00	0	0	2.33e01
V	-5.57e03	3.70e01	0	-7.71e00	0
X^*X	0	3.57e03	0	-3.71e03	2.78e03
X^*Y	-2.41e07	0	0	0	0
X^*Z	0	0	1.89e04	7.22e02	0
X^*W	3.14e07	-1.89e04	0	1.26e04	-1.50e04
X^*U	0	-7.22e02	-1.26e04	0	0
X^*V	0	0	1.50e04	0	0
Y^*Y	0	3.57e03	0	-3.71e03	2.78e03
Y^*Z	0	0	-2.00e04	0	0
Y^*W	3.88e07	2.00e04	0	-6.42e03	1.69e03
Y^*U	0	0	6.42e03	0	0
Y^*V	0	0	-1.69e03	0	0
Z^*Z	0	0	0	-7.58e03	0
Z^*W	0	0	0	0	0

Z*U	0	7.58e03	0	0	0
Z*V	0	0	0	8.88e03	0
W*W	-6.20e07	0	0	0	0
W*U	0	0	0	0	0
W*V	0	0	0	0	0
U*U	0	0	0	0	3.17e04
U*V	0	-8.88e03	0	-3.17e04	-2.04e04
V*V	0	0	0	2.04e04	0
X*X*X	0	0	5.77e06	0	-8.03e04
X*X*Y	0	0	-1.54e07	0	0
X*X*Z	0	1.09e06	0	0	1.95e06
X*X*W	0	0	1.14e07	0	0
X*X*U	0	0	0	0	0
X*X*V	0	1.27e06	0	0	0
X*Y*Y	0	0	0	0	0
X*Y*Z	0	-1.41e06	0	1.38e06	-4.68e05
X*Y*W	0	0	-1.61e07	0	0
X*Y*U	0	0	0	-2.58e06	0
X*Y*V	0	-6.93e06	0	0	5.68e06
X*Z*Z	0	0	0	0	0
X*Z*W	0	5.99e06	0	-1.26e06	0
X*Z*U	0	0	0	0	0
X*Z*V	0	8.27e06	-2.94e07	0	0
X*W*W	0	0	0	0	0
X*W*U	0	0	0	0	0
X*W*V	0	0	0	4.05e06	0
X*U*U	0	0	0	4.86e06	-1.16e07
X*U*V	0	0	0	-1.29e07	0
X*V*V	0	0	-6.66e07	0	0
Y*Y*Y	0	0	-1.51e07	0	0
Y*Y*Z	0	-1.28e06	0	2.37e06	0

Y*Y*W	0	0	-7.69e05	0	0
Y*Y*U	0	0	0	0	0
Y*Y*V	0	3.14e06	0	0	0
Y*Z*Z	0	0	-4.25e07	1.15e06	1.35e06
Y*Z*W	0	-5.89e06	0	0	5.97e06
Y*Z*U	0	0	-2.04e07	-6.37e06	2.42e06
Y*Z*V	0	0	0	0	0
Y*W*W	0	0	0	0	0
Y*W*U	0	0	0	0	0
Y*W*V	0	0	0	0	0
Y*U*U	0	0	7.50e06	0	0
Y*U*V	0	0	0	0	0
Y*V*V	0	0	-1.75e07	0	-1.69e07
Z*Z*Z	0	0	0	0	0
Z*Z*W	0	0	0	0	0
Z*Z*U	0	0	-1.28e07	0	0
Z*Z*V	0	-1.67e07	0	0	0
Z*W*W	0	0	0	0	-9.42e06
Z*W*U	0	2.87e07	0	0	0
Z*W*V	0	0	-5.83e07	0	0
Z*U*U	0	0	0	0	-1.81e07
Z*U*V	0	0	0	0	2.18e07
Z*V*V	0	-7.96e07	0	0	-3.85e07
W*W*W	0	0	0	1.02e06	0
W*W*U	0	0	0	0	-3.61e07
W*W*V	0	4.65e07	0	0	0
W*U*U	0	0	0	0	-2.73e07
W*U*V	0	0	0	0	0
W*V*V	0	0	0	0	0
U*U*U	0	-3.05e07	0	0	0
U*U*V	0	0	0	0	0

U*V*V	0	0	0	0	0
V*V*V	0	1.22e08	0	0	0

Appendix E

Spectrograms

E.1 Closed Ends Data

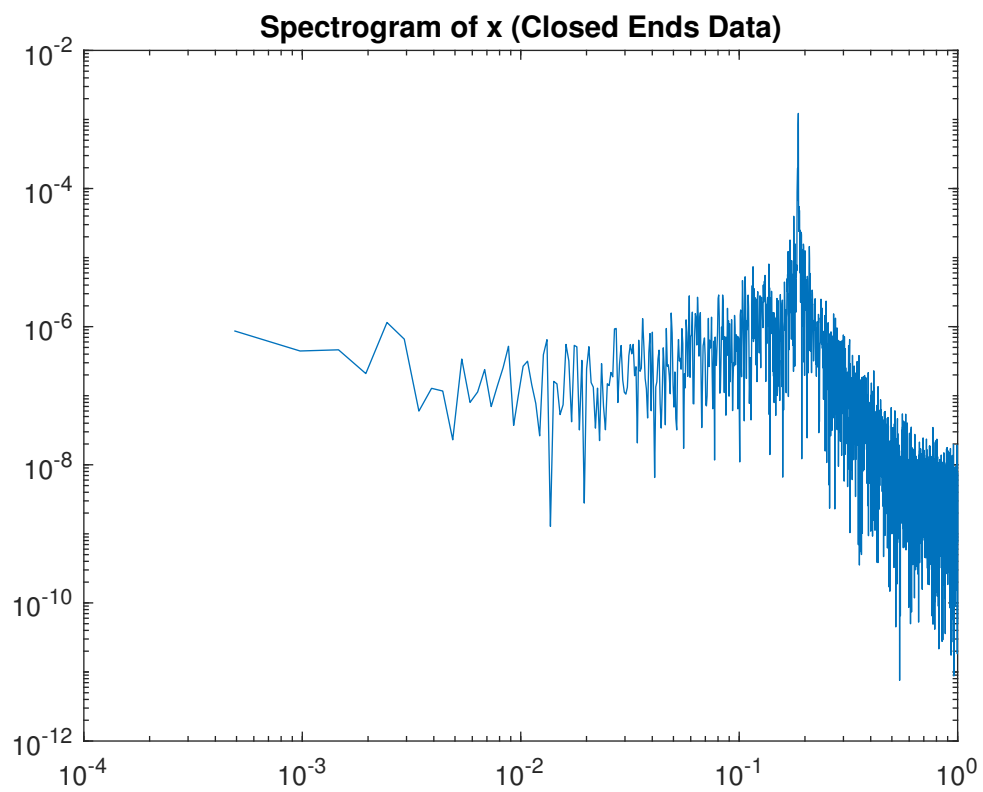


FIGURE E.1. Spectrogram of x , one of the modes that makes up our first harmonic oscillating pair

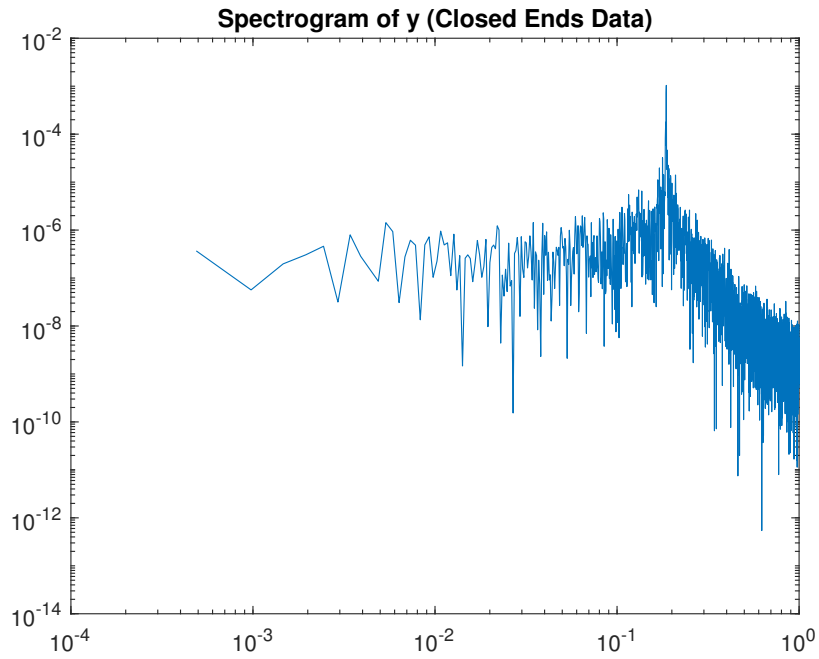


FIGURE E.2. Spectrogram of y , one of the modes that makes up our first harmonic oscillating pair

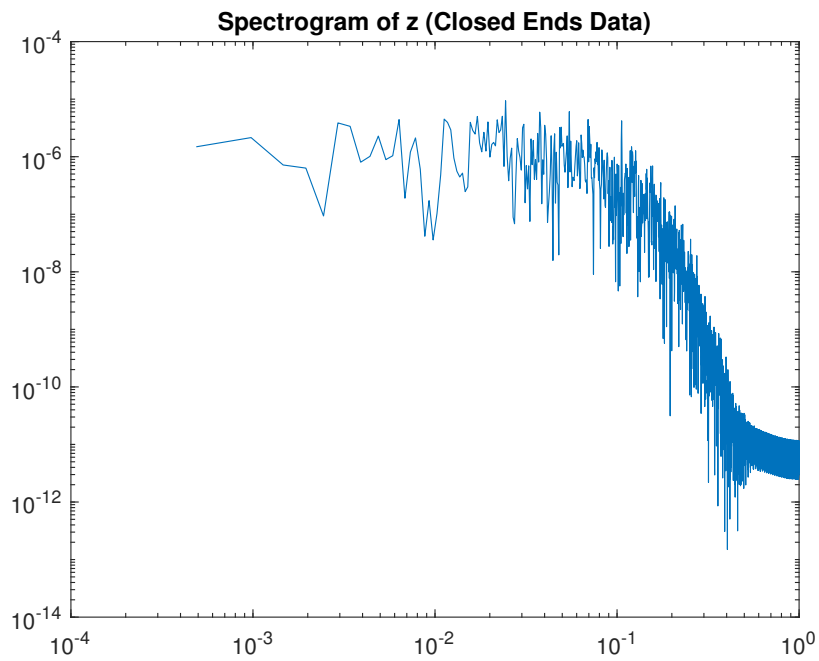


FIGURE E.3. Spectrogram of z , also referred to as our shift mode

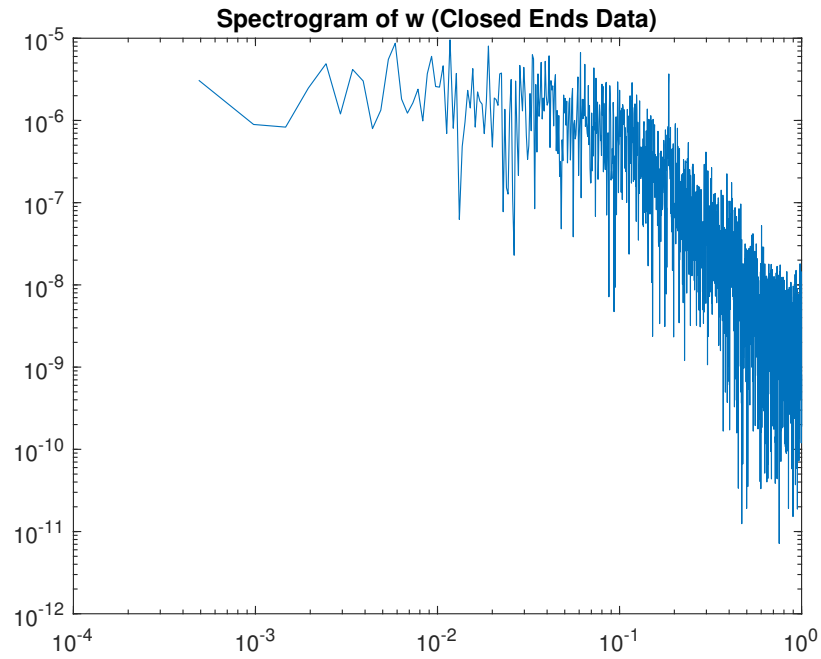


FIGURE E.4. Spectrogram of w , also referred to as our flapping mode

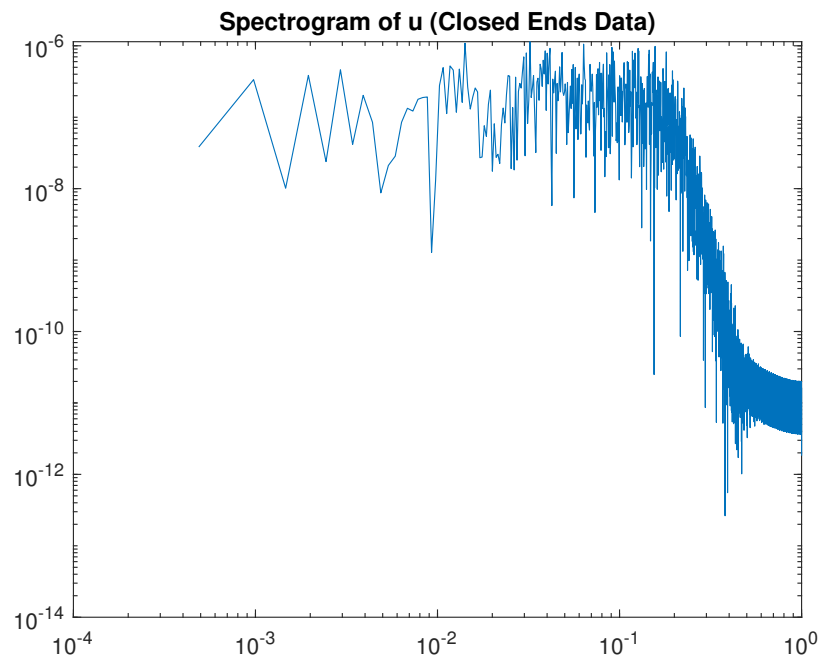


FIGURE E.5. Spectrogram of u , one of the modes that makes up our second harmonic oscillating pair

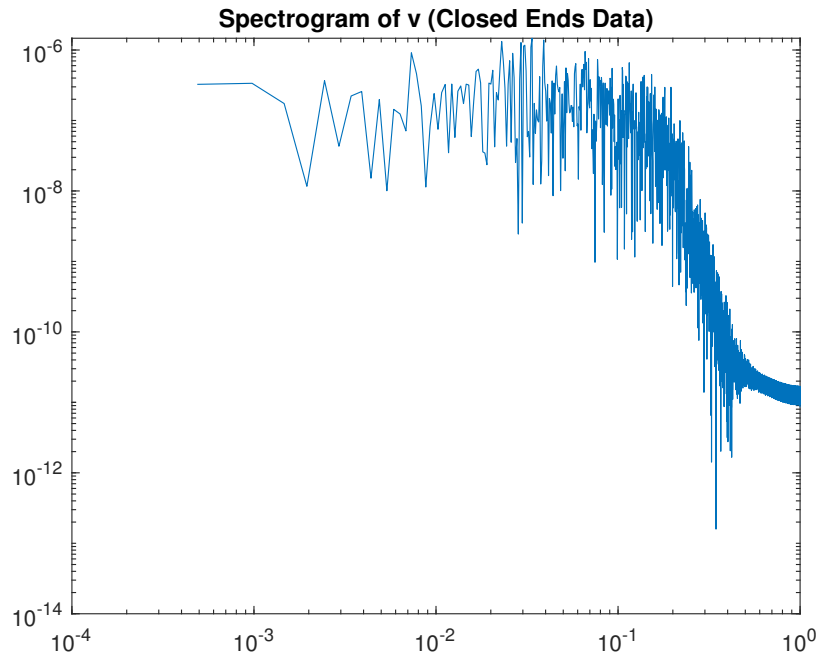


FIGURE E.6. Spectrogram of v , one of the modes that makes up our second harmonic oscillating pair

E.2 Open Ends Data

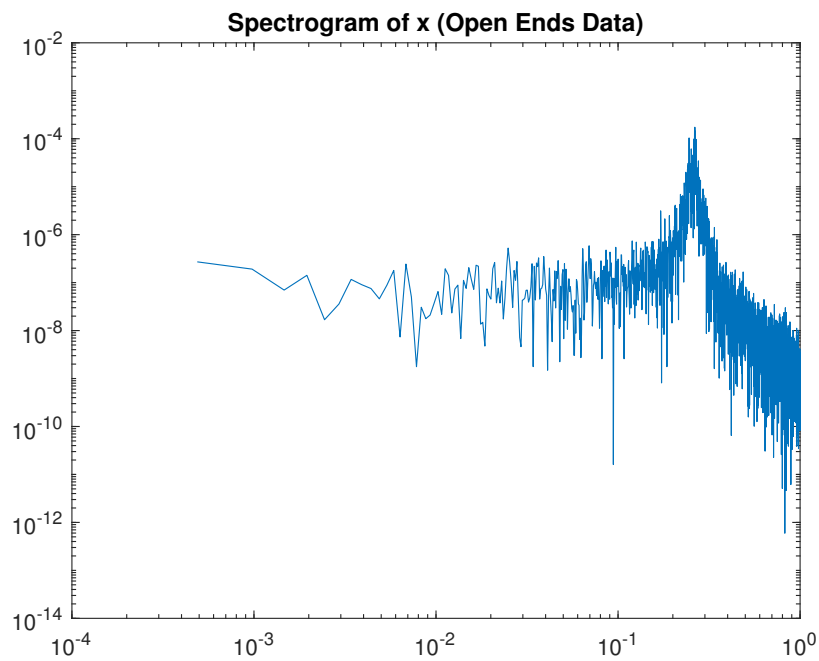


FIGURE E.7. Spectrogram of x , one of the modes that makes up our first harmonic oscillating pair

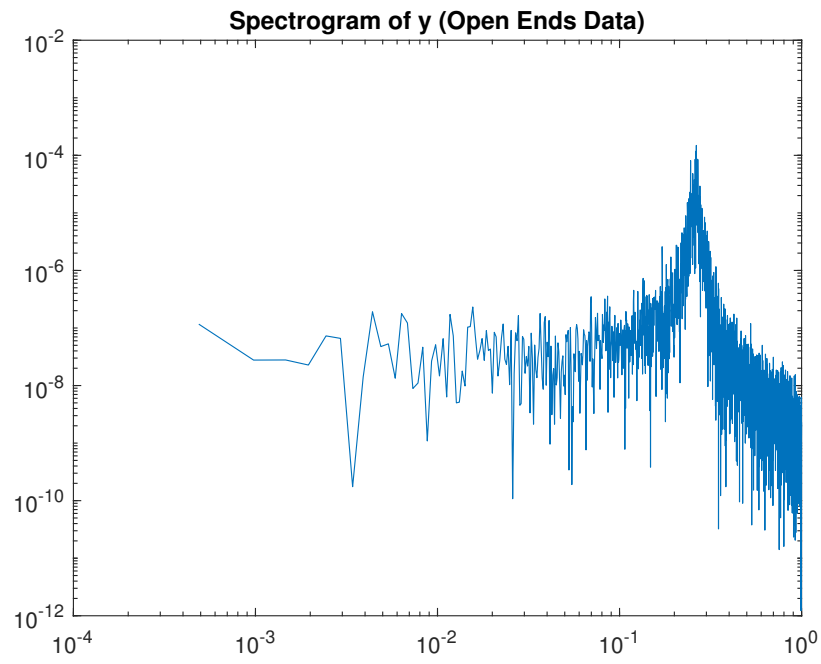


FIGURE E.8. Spectrogram of y , one of the modes that makes up our first harmonic oscillating pair

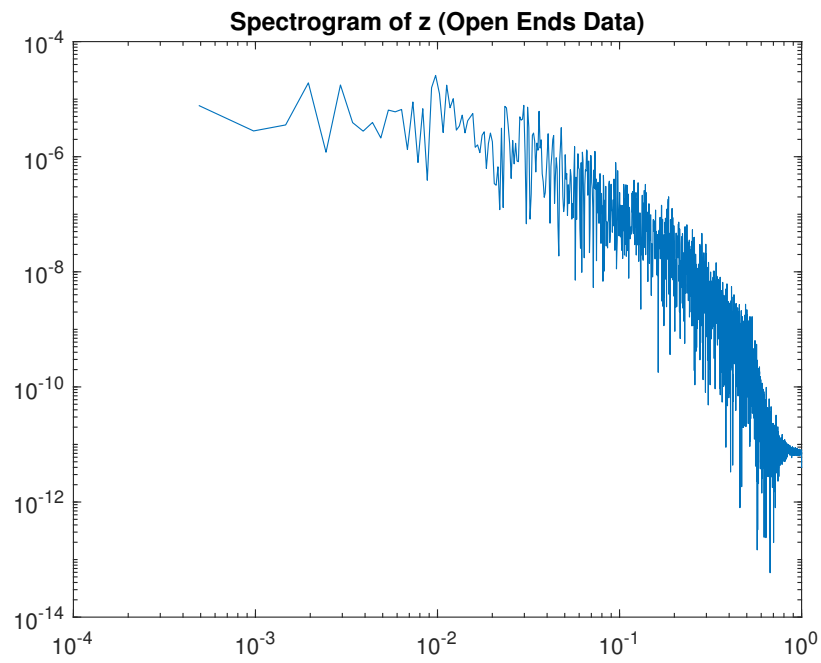
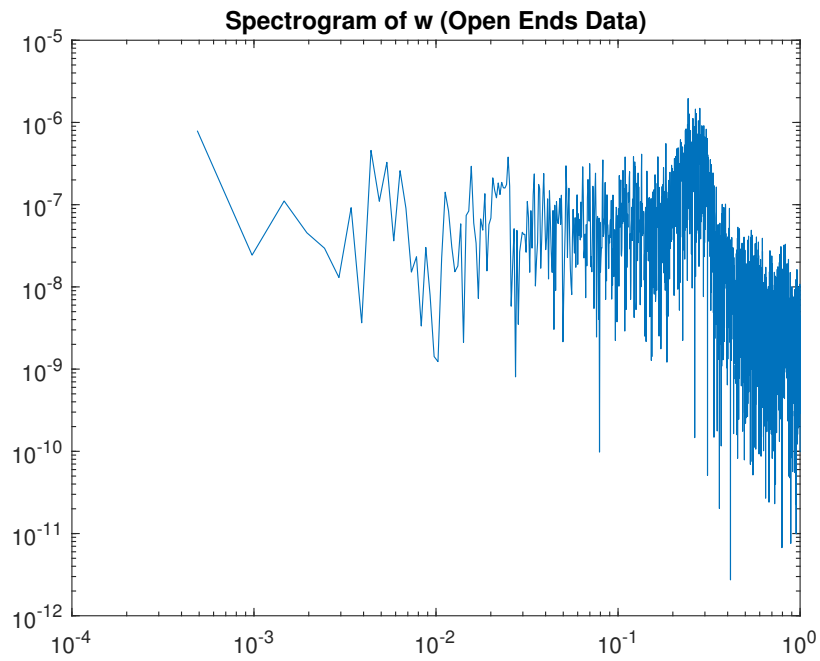
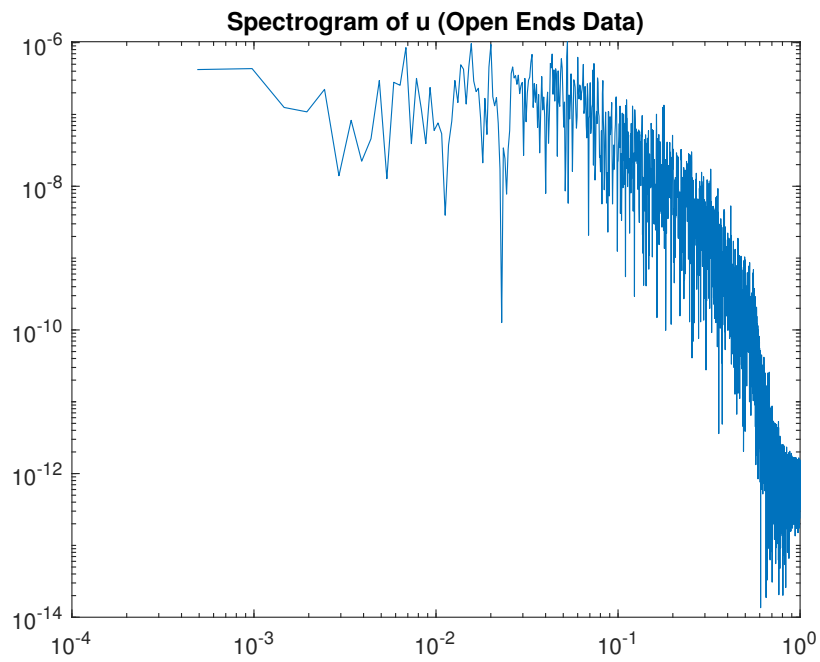


FIGURE E.9. Spectrogram of z , also referred to as our shift mode

FIGURE E.10. Spectrogram of w , also referred to as our flapping modeFIGURE E.11. Spectrogram of u , one of the modes that makes up our second harmonic oscillating pair

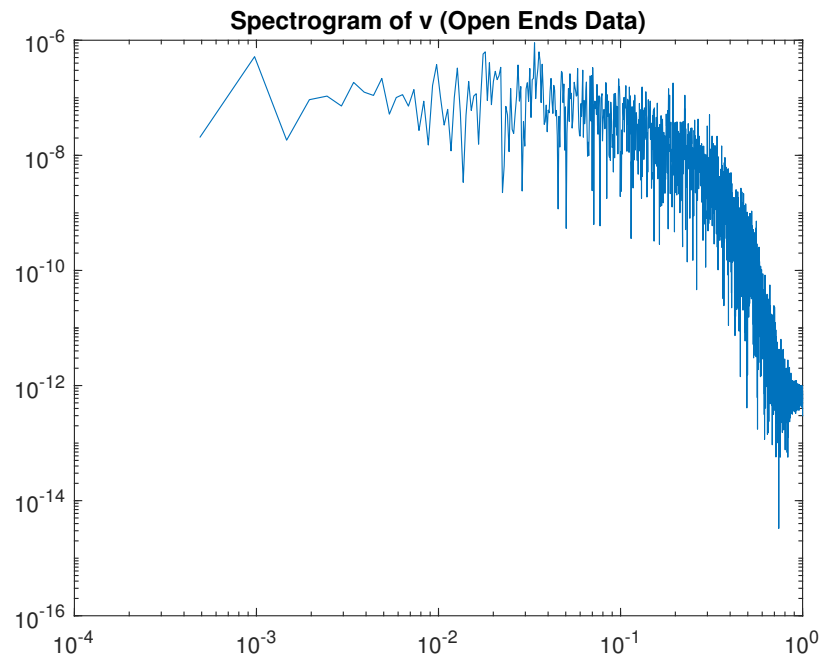


FIGURE E.12. Spectrogram of v , one of the modes that makes up our second harmonic oscillating pair

Bibliography

- Agrey, Kaden et al. (2022). "On Lasso with constraints for the sparse identification of the flow behind 2d bluff bodies" [Unpublished Manuscript]. eng.
- Braun, Eric, Kaden Agrey, and Robert J Martinuzzi (2020). "End effects of nominally two-dimensional thin flat plates". eng. In: *The International journal of heat and fluid flow* 86. ISSN: 0142-727X.
- Brunton, Steven L, Joshua L Proctor, and J Nathan Kutz (2016). "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". eng. In: *Proceedings of the National Academy of Sciences - PNAS* 113.15, pp. 3932–3937. ISSN: 0027-8424.
- Bühlmann, Peter (2011). *Statistics for High-Dimensional Data Methods, Theory and Applications*. eng. Springer Series in Statistics. ISBN: 3-642-20192-X.
- Efron, Bradley et al. (2004). "Least Angle Regression". In: *The Annals of Statistics* 32.2, pp. 407–451. ISSN: 00905364. URL: <http://www.jstor.org/stable/3448465>.
- Friedman, Jerome H., Trevor Hastie, and Rob Tibshirani (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software, Articles* 33.1, pp. 1–22. ISSN: 1548-7660. DOI: [10.18637/jss.v033.i01](https://doi.org/10.18637/jss.v033.i01). URL: <https://www.jstatsoft.org/v033/i01>.
- Ghaoui, Laurent El, Vivian Viallon, and Tarek Rabbani (2011). *Safe Feature Elimination for the LASSO and Sparse Supervised Learning Problems*. arXiv: [1009.4219](https://arxiv.org/abs/1009.4219) [cs.LG].
- Hastie, Trevor, Juanyang Qian, and Kenneth Tay (2021). "An Introduction to glmnet". eng. In: URL: <https://glmnet.stanford.edu/articles/glmnet.html>.
- Loiseau, Jean-Christophe and Steven L Brunton (2018). "Constrained sparse Galerkin regression". eng. In: *Journal of fluid mechanics* 838, pp. 42–67. ISSN: 0022-1120.

- NOACK, BERND R et al. (2003). "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake". eng. In: *Journal of fluid mechanics* 497, pp. 335–363. ISSN: 0022-1120.
- Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288. ISSN: 00359246. URL: <http://www.jstor.org/stable/2346178>.
- Tibshirani, Robert et al. (2012). "Strong rules for discarding predictors in lasso-type problems". eng. In: *Journal of the Royal Statistical Society. Series B, Statistical methodology*. Journal of the Royal Statistical Society Series B 74.2, pp. 245–266. ISSN: 1369-7412. URL: <https://doi.org/10.1111/j.1467-9868.2011.01004.x>.
- Wu, Tong Tong and Kenneth Lange (2008). "Coordinate descent algorithms for lasso penalized regression". In: *The Annals of Applied Statistics* 2.1, pp. 224 –244. DOI: [10.1214/07-AOAS147](https://doi.org/10.1214/07-AOAS147). URL: <https://doi.org/10.1214/07-AOAS147>.