

Syracuse University

SURFACE

Dissertations - ALL

SURFACE

5-30-2014

APPLICATION OF DATA FUSION TO FLUID DYNAMIC DATA

Christopher John Ruscher
Syracuse University

Follow this and additional works at: <https://surface.syr.edu/etd>



Part of the [Engineering Commons](#)

Recommended Citation

Ruscher, Christopher John, "APPLICATION OF DATA FUSION TO FLUID DYNAMIC DATA" (2014).
Dissertations - ALL. 114.
<https://surface.syr.edu/etd/114>

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Abstract

In recent years, there have been improvements in the methods of obtaining fluid dynamic data, which has led to the generation of vast amounts of data. Extracting the useful information from large data sets can be a challenging task when investigating data from a single source. However, most experiments use data from multiple sources, such as particle image velocimetry (PIV), pressure sensors, acoustic measurements, and computational fluid dynamics (CFD), to name a few. Knowing the strengths and weaknesses of each measurement technique, one can fuse the data together to improve the understanding of the problem being studied. Concepts from the data fusion community are used to combine fluid dynamic data from the different data sources. The data is fused using techniques commonly used by the fluid dynamics community, such as proper orthogonal decomposition (POD), linear stochastic estimation (LSE), and wavelet analysis. This process can generate large quantities of data and a method of handling all of the data and the techniques in an efficient manner is required. To accomplish this, a framework was developed that is capable of tracking, storing, and, manipulating data.

With the framework and techniques, data fusion can be applied. Data fusion is first applied to a synthetic data set to determine the best methods of fusing data. Data fusion was then applied to airfoil data that was obtained from PIV, CFD, and pressure to test the ideas from the synthetic data. With the knowledge gained from applying fusion to the synthetic data and airfoil data, these techniques are ultimately applied to data for a Mach 0.6 jet obtained from large-window PIV (LWPIV), time-resolved PIV (TRPIV), and pressure.

Through the fusion of the different data sets, occlusion in the jet data were estimated within 6% error using a new POD based technique called Fused POD. In addition, a technique called Dynamic Gappy POD was created to fuse TRPIV and LWPIV to generate a large-window time-resolved data set. This technique had less error than other standard techniques for accomplishing this such as pressure-based stochastic estimation.

The work presented in this document lays the groundwork for future applications

of data fusion to fluid dynamic data. With the success of the work in this document, one can begin to apply the ideas from data fusion to other types of fluid dynamic problems, such as bluff bodies, unsteady aerodynamics, and other. These ideas could be used to help improve understanding in the field of fluid dynamics due to the current limitations of obtaining data and the need to better understand flow phenomena.

APPLICATION OF DATA FUSION TO FLUID DYNAMIC DATA

by

Christopher John Ruscher

B.S. Syracuse University, May 2009

M.S. Syracuse University, May 2011

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
MECHANICAL AND AEROSPACE ENGINEERING
AT
SYRACUSE UNIVERSITY
SYRACUSE, NEW YORK, U.S.A.

May 2014

Copyright © 2014 by Christopher John Ruscher
All Rights Reserved

Acknowledgements

I would like to, at this point, acknowledge all of the people who made it possible for me to get to this point. Without the help of these people I would not be who I am today.

First I would like to thank my family, especially my Mother, Father, and my brother Kevin who have supported me unconditionally in everything I have done.

I would also like to thank my advisor, Dr. John Dannenhoffer for working with me for the past five years and always pushing me to do better and be better. He was always available to help especially early in the morning when no one else was in Link. Thanks to his help and input I have been able to succeed.

I would also like to thank my co-advisor, Dr. Mark Glauser for allowing me to work with his research group and always making me feel like part of the team. Dr. Glauser was also a key player in my success.

I want to thank the members of my committee, Dr. Pramod Varshney, Dr. Jacques Lewalle, Dr. Melissa Green, and Dr. Benjamin Akih-Kumgeh, for taking time out of their busy schedules to read my thesis, listen to my defense, and provide feedback to improve my research.

I wish to thank Dr. Zachary Berger for being one of my closest friends for the past nine years. He has shared in my good times and helped me get through the bad times and come out a better person. We have had many discussions including engineering, physics, sports, and life in general that have made me a better engineer, better friend, and better person. Congratulations Zach, we did it!

To Matt Berry for being a good friend and colleague and for all of the fruitful conversations Matt, Zach, and I have had in our living room the past two years about pretty much everything and anything including engineering.

To Dr. Patrick Shea for always being willing to take time to explain something that was confusing me and always having a willingness to help with anything.

I would also like to thank everyone who has been a part of Dr. Glauser's research team during my time here including, Dr. Guannan Wang, Dr. Kerwin Low, Dr. Ryan Wallace, Jakub Walczak, Sirkar Kaligotla, Zhe Bai, Pingqing Kan, Andy Magstadt, John-Michael Velarde and all of the undergraduates who have been part of the group

at one point or another. All of them have helped me in one way or another by letting me use their data, helping explain experimental methods, helping me learn about the different data processing tools, and helping me to grow as an engineer and person.

I would also like to thank everyone who has been a part of Dr. Dannenhoffer's research group including Bridget Dixon, Dale Ringham, and Ben Henry for helping me become a better presenter, helping me to see the big picture and helping me prepare for my quals and defense.

I also am grateful for all of the advice from Dr. Gina Lee-Glauser and having me and all of my colleagues to her house and summer house in the Adirondacks.

I also thank Barry Kiel and Balu Sekar for giving me the opportunity to work with them at AFRL in Dayton for two summers.

To all of the friends I have made over the years who spent countless hours working on homework or projects with me and making my time at Syracuse fun and enjoyable including David Stokes, Jerry Vo, Omar Assaf, Jon Kim, and Dallas Conway.

I also want to thank the L.C. Smith staff who have helped me with all of the paper work and clerical task over the years including Debbie Brown, Kim Drumm-Underwood, Kathy Datthyn-Madigan, Colleen Patterson and Linda Lowe.

To everyone I worked with in the PRIDE office as an undergraduate including Kate Pynn and Karen Davis.

I am also grateful for the funding from General Electric and ISSI over the last two years which made it possible for me to do this work.

I want to also thank Syracuse University for being my home for the past nine years and all of the support it has given me over the years.

Lastly, I want to thank Kathleen Joyce. She is the main reason I choose to come to Syracuse ten years ago when she and then Dean Spina came to Rochester to recruit students in the area. This ultimately led to me coming to Syracuse to do my undergraduate studies and eventually my graduate studies.

Syracuse, NY
April 2, 2014

Christopher Ruscher

Contents

Abstract	
Acknowledgements	v
List of Figures	xi
List of Tables	xviii
Nomenclature	xix
Abbreviations	xxiv
1 Introduction	1
1.1 Objective	1
1.2 Data Fusion	2
1.2.1 Business Intelligence	4
1.2.2 Navigation	5
1.2.3 Target Identification	6
1.2.4 Image Fusion	8
1.2.5 Data Fusion in Nature	11
1.3 Strengths and Weaknesses of Data Sources	12
1.4 Analysis Tools	15
1.4.1 Fourier Analysis	15
1.4.2 Proper Orthogonal Decomposition	16
1.4.3 Stochastic Estimation	25
1.4.4 Modeling	30
1.4.5 Kalman Filtering	32
1.4.6 Wavelets	34
1.4.7 Application to Fluid Dynamics	40

2	Orange High Speed Data (HSD) Framework	42
2.1	ORANGE High Speed Data	43
2.1.1	Data Management	44
2.1.2	ORANGE POD	46
2.1.3	ORANGE Wavelets	48
2.1.4	ORANGE Correlate	49
2.1.5	ORANGE LSE	51
2.2	Examples	53
3	Synthetic Data	59
3.1	One-Dimensional Test Case	59
3.1.1	Test Case 1: Downsampled/Dispersed Signal	60
3.1.2	Test Case 2: Noisy Downsampled/Dispersed Signal	61
3.1.3	Test Case 3: Gappy/Dispersed Signal	62
3.1.4	Test Case 4: Gappy, Multiple Windows, Noisy, Downsampled/ Dispersed Signal	62
3.2	Discrete Wavelet Fusion	63
3.2.1	Test Case 1	64
3.2.2	Test Case 2	66
3.2.3	Test Case 3	69
3.2.4	Comparison of Different Daubechies Wavelets	70
3.3	Gappy POD Fusion	71
3.3.1	Test Case 1	72
3.3.2	Test Case 2	72
3.3.3	Test Case 3	73
3.4	LSE Fusion	75
3.5	Fusing Techniques Together for Test Case 4	78
3.6	Summary of Fusion Techniques	80
3.7	Inspiration for the Application of Fusion on Real Fluid Dynamic Data	81
4	Fusion for Airfoil Data	83
4.1	Definition of an Airfoil	84
4.2	Flow Over an Airfoil	86
4.3	Experimental Setup	87
4.4	Computational Setup	89
4.4.1	Grid Generation	89
4.4.2	Computational Setup	92
4.5	Data Fusion Applied to the Airfoil	98
4.6	Hole Filling	99

4.6.1	Fused POD	99
4.6.2	Application to Airfoil Data	103
4.7	Temporal Estimation	109
4.7.1	Stochastic Estimation	110
4.7.2	Stochastic Modeling	112
4.8	Summary of Data Fusion on a Flow Over an Airfoil	115
5	Fusion of Mach 0.6 Jet Data	117
5.1	Jet Flow	118
5.1.1	Jet Flow Structure	118
5.1.2	Aeroacoustics	120
5.2	Experimental Setup	122
5.2.1	Large Window PIV	123
5.2.2	Time-Resolved PIV	125
5.3	Summary of Data Fusion for the Jet	127
5.4	Hole Filling	128
5.4.1	Sensor Occlusion Estimation	129
5.4.2	Estimation of Spurious Vectors/Particle Dropout	150
5.5	Domain Extension/Estimation	159
5.5.1	Stochastic Estimation Results	161
5.5.2	Stochastic Modeling	168
5.5.3	Gappy POD	171
5.5.4	Dynamic Gappy POD	182
5.5.5	Comparison of Results	186
5.6	Summary of Data Fusion on a High Speed Jet Flow	189
6	Concluding Remarks	191
6.1	Summary	191
6.2	Conclusions	194
6.3	Suggested Future Work	196
A	One Dimensional and Unsteady (Shock Tube)	200
A.1	Shock Tube Theory	200
A.2	Experimental setup	204
A.3	Simulation Setup	206
A.3.1	Boundary and Initial Conditions	207
A.3.2	Computational Results	208
A.4	Fusion Results	210
A.4.1	Shock Arrival Time	210
A.4.2	Kalman Filter Fusion	212

A.4.3 Summary and Lessons Learned	216
B ORANGE HSD User Manual:	218
C Pre-submission of High Speed Optical Measurements in a Jet Flow:	266
Bibliography	278

List of Figures

1.1	<i>Data Fusion and Target Identification Used for Security (Image Obtained from ultra-ccs.com/business/bordersecurity)</i>	7
1.2	<i>Example of Image Fusion on a High Resolution Gray Scale Image and a Low Resolution Color Image</i>	8
1.3	<i>Block Diagram for Pyramid Decomposition</i>	10
1.4	<i>Diagram of the Image Fusion Technique</i>	11
1.5	<i>Reconstruction of a Square Wave using Fourier Modes</i>	15
1.6	<i>Diagram of the Gappy POD Method</i>	22
1.7	<i>Diagram of the Iterative Gappy POD Method</i>	24
1.8	<i>Mexican Hat Wavelet</i>	35
1.9	<i>Morlet Wavelet</i>	35
1.10	<i>Wavelet Image Fusion Process</i>	37
2.1	<i>Screenshot of ORANGE High Speed Data Main Screen</i>	44
2.2	<i>Screenshot of Data Management: The Data Management Tool Included in ORANGE HSD</i>	45
2.3	<i>Generating a Mask for the NACA 4412 Data</i>	46
2.4	<i>Screenshot of ORANGE POD: The Proper Orthogonal Decomposition Tool Included in ORANGE HSD</i>	47
2.5	<i>First POD Mode for the Streamwise Velocity Over the NACA 4412 Airfoil</i>	47
2.6	<i>Screenshot of ORANGE Wavelets: The Wavelet Tool Included in ORANGE HSD</i>	48
2.7	<i>Mexican Hat and Morlet Wavelet Map for NACA 4412 Surface Pressure</i>	49
2.8	<i>Screenshot of ORANGE Correlate: The Spectral Methods Tool Included in ORANGE HSD</i>	50
2.9	<i>Autospectrum for NACA 4412 Surface Pressure</i>	50
2.10	<i>Screenshot of ORANGE LSE: The Stochastic Estimation Tool Included in ORANGE HSD</i>	51
2.11	<i>Comparison of the Actual, POD Reconstruction, and LSE Velocity Fields</i>	52
2.12	<i>Extraction of Data from the TRPIV for Comparison with LDV Data</i>	54

2.13	<i>First Four Spatial Modes in the Radial Direction (Figure 6.54 in Berger [1])</i>	56
2.14	<i>Mode One for the Split POD Between Baseline and Control for a Mach 0.6 Jet</i>	57
3.1	<i>Test Case 1: Downsampled Signal and Dispersed Signal</i>	61
3.2	<i>Test Case 2: Downsampled and Noisy Signal and Dispersed Signal</i>	62
3.3	<i>Test Case 3: Gappy Signal and Dispersed Signal</i>	62
3.4	<i>Test Case 4: Gappy, Multiple Window, Noisy, Downsampled Signal and Dispersed Signal</i>	63
3.5	<i>Wavelet Fusion of a Downsampled Signal and Dispersed Signal without Secondary Compressions</i>	64
3.6	<i>Wavelet Fusion of a Downsampled Signal and Dispersed Signal with One Secondary Compression</i>	65
3.7	<i>Wavelet Fusion of a Downsampled Signal and Dispersed Signal with Five Secondary Compressions</i>	65
3.8	<i>Effect of Secondary Compressions on a Downsampled Signal and Dispersed Signal</i>	66
3.9	<i>Wavelet Fusion of a Noisy Downsampled Signal and Dispersed Signal with no Secondary Compressions</i>	67
3.10	<i>Wavelet Fusion of a Noisy Downsampled Signal and Dispersed Signal with One Secondary Compression</i>	67
3.11	<i>Wavelet Fusion of a Noisy Downsampled Signal and Dispersed Signal with Five Secondary Compressions</i>	68
3.12	<i>Effect of Secondary Compressions of Wavelet Fusion for a Noisy Downsampled Signal and Dispersed Signal</i>	69
3.13	<i>Wavelet Fusion Applied to Gappy Data</i>	70
3.14	<i>Wavelet Fusion Applied to Gappy Data by Inserting Another Signal in the Gappy Region</i>	70
3.15	<i>RMS Error Using Different Wavelets for Fusion of a Noisy Downsampled Signal and Dispersed Signal</i>	71
3.16	<i>Gappy POD Fusion Applied to a Downsampled Signal</i>	72
3.17	<i>Gappy POD Fusion Applied to a Downsampled and Noisy Signal</i>	73
3.18	<i>Gappy POD Fusion Applied to a Gappy Signal</i>	73
3.19	<i>Gappy POD Fusion Applied to a Gappy Signal with a “Bad” CFD-inspired Signal</i>	74
3.20	<i>Convergence Rate of POD vs. the RMS Error</i>	75
3.21	<i>LSE Fusion for a Downsampled Signal Using the First 12% of the PIV-inspired Signal</i>	76

3.22	<i>LSE Fusion for a Downsampled Signal Using the First 25% of the PIV-inspired Signal</i>	76
3.23	<i>LSE Fusion for a Downsampled Signal Using the First 50% of the PIV-inspired Signal</i>	77
3.24	<i>Correlation Between the CFD- and PIV-Inspired data</i>	77
3.25	<i>Diagram of a Technique that uses Wavelet Fusion and Gappy POD</i>	79
3.26	<i>Fusion of Downsampled, Noisy, Multiple Window, Gappy Data and Dispersed Data using the Fused Technique</i>	80
4.1	<i>Airfoil Nomenclature</i>	84
4.2	<i>Diagram of Stall on a NACA 4412 Airfoil</i>	86
4.3	<i>Diagram of the Experimental Setup for the NACA 4412 Test</i>	88
4.4	<i>Schematic of the Pressure Sensors on the NACA 4412 Airfoil</i>	88
4.5	<i>Definition of the Coordinate System for the TFI Algorithm</i>	90
4.6	<i>Airfoil Computational Grid</i>	91
4.7	<i>Airfoil Boundary Conditions</i>	96
4.8	<i>Instantaneous Streamwise Mach Number for a NACA 4412 Airfoil at 16° AOA Obtained with CFD and PIV</i>	97
4.9	<i>Comparison of Mean Streamwise Mach Number for a NACA 4412 Airfoil at 16° AOA Obtained with CFD and PIV</i>	97
4.10	<i>Venn Diagram Showing the Different Fusion Techniques applied to the Airfoil Data</i>	99
4.11	<i>Errors That Can Arise in PIV Data</i>	100
4.12	<i>Diagram of the Fused POD Technique</i>	102
4.13	<i>Gappy POD and Wavelet Fusion for a NACA 4412 Airfoil</i>	102
4.14	<i>Correlation between CFD and PIV POD Modes</i>	104
4.15	<i>Effect of Hole Size on Fused POD for PIV and CFD of a NACA 4412 Airfoil</i>	105
4.16	<i>Effect of Number of Modes on the Estimation of Missing Data for a NACA 4412 Airfoil using Fused POD with CFD Modes</i>	106
4.17	<i>POD Convergence Rate of a CFD simulated NACA 4412 Airfoil</i>	107
4.18	<i>Instantaneous Repaired Snapshot for a NACA 4412 Airfoil</i>	108
4.19	<i>Distribution of Error using Fused POD or CFD to Repair a Snapshot</i>	108
4.20	<i>Effect of Number of Modes on the Estimation of Missing Data for a NACA 4412 Airfoil using Fused POD with Exact Modes</i>	109
4.21	<i>Correlation between the Surface Pressure and the Velocity for a NACA 4412 Airfoil</i>	110
4.22	<i>Fluctuating Streamwise Velocity Estimated using sLSE</i>	112

4.23	<i>Comparison of the First Two POD Modes for the PIV Data and the Stochastic Model Estimate for a NACA 4412 Airfoil</i>	113
4.24	<i>Comparison of the First Two POD Modes for the CFD Data and the Stochastic Model Estimate for a NACA 4412 Airfoil</i>	114
4.25	<i>Cumulative Error Between the Estimated and Actual POD coefficients for Modes 1 and 2</i>	114
4.26	<i>Comparison of the First Two POD Modes for the CFD Data and the Stochastic Model Estimate with a Kalman Filter for a NACA 4412 Airfoil</i>	115
5.1	<i>Schematic of an Axisymmetric Jet Flow Field</i>	118
5.2	<i>Syracuse University Anechoic Chamber at the Skytop Facility</i>	123
5.3	<i>2013 LWPIV Experimental Setup for the Jet at the Syracuse University Anechoic Chamber at the Skytop Facility</i>	124
5.4	<i>Kulite Array Configuration for the Jet at the Syracuse University Anechoic Chamber at the Skytop Facility</i>	125
5.5	<i>2011 Time-Resolved PIV Experimental Setup at the Syracuse University Anechoic Chamber at the Skytop Facility</i>	126
5.6	<i>Data Fusion and Technique Fusion used on the Mach 0.6 Jet Data</i>	127
5.7	<i>Sensor Occlusions in the LWPIV data of a Mach 0.6 Jet</i>	128
5.8	<i>Reflections in the LWPIV Data for a Mach 0.6 Jet</i>	129
5.9	<i>Comparison of POD Modes Calculated With and Without a Mask</i>	130
5.10	<i>Three Ways of Splitting the Jet Data into Symmetric Top and Bottom Halves</i>	132
5.11	<i>Correlation Plots Between POD Modes from Top and Bottom of an Axisymmetric Jet Using Different Splitting Methods</i>	133
5.12	<i>Error Distribution for Fused POD using Different Splitting Methods in an Axisymmetric Jet</i>	136
5.13	<i>Instantaneous Streamwise Velocity Estimated with Fused POD using Different Splitting Methods on an Axisymmetric Jet</i>	137
5.14	<i>RMS of the Streamwise Velocity Estimated with Fused POD using Different Splitting Methods in an Axisymmetric Jet</i>	138
5.15	<i>RMS of Streamwise Velocity Estimated with Fused POD Before and After using Wavelet Image Fusion on a Mach 0.6 Jet</i>	139
5.16	<i>The Effects of Using the Wavelet Fusion in Fused POD</i>	140
5.17	<i>The Different Artificial Hole Configurations Used to Test the Effect of Hole Size</i>	141
5.18	<i>Distribution of Error for Different Size Holes and Different Domain Sizes</i>	142
5.19	<i>Effect of Hole Size on Fused POD Estimate</i>	143

5.20	<i>Effect of POD Modal Convergence When using Fused POD to Fill-in Missing Data for an Axisymmetric Jet</i>	145
5.21	<i>The Effects of using the Average in Fused POD</i>	146
5.22	<i>Error Distribution for Fused POD using Instantaneous vs. Fluctuating Velocity</i>	147
5.23	<i>Noisy Sine Wave Repaired with Sinusoidal Basis</i>	148
5.24	<i>Low-Dimensional Representation of the Streamwise Velocity</i>	149
5.25	<i>Low-Dimensional Representation of the RMS of the Streamwise Velocity</i>	150
5.26	<i>PIV Data for a Mach 0.6 Jet with no Post Processing to Compensate for Errors</i>	152
5.27	<i>PIV Data for a Mach 0.6 Jet with PIV Post Processing to Compensate for Errors</i>	153
5.28	<i>PIV Data with Bad Vectors Masked Out</i>	153
5.29	<i>PIV Data with Gappy POD Corrections</i>	154
5.30	<i>Statistics for the TRPIV Before and After using Fused/Gappy POD for a Mach 0.6 Jet</i>	155
5.31	<i>Error Between Fused POD and Iterative Gappy POD at the Cumulative Sum of the Percent of “Bad” Points</i>	156
5.32	<i>The Error Associated with Using Fused POD Instead of Iterative Gappy POD to Repair PIV Snapshots with Missing Data</i>	158
5.33	<i>The Computational Time Savings from Using Fused POD Instead of Iterative Gappy POD to Repair PIV Snapshots with Missing Data</i>	159
5.34	<i>The Ultimate Goal of the Estimation Techniques: Combine Pressure, TRPIV, and LWPIV to Obtain a Large-Window Time-Resolved Data Set</i>	160
5.35	<i>The Ultimate Goal of the sLSE: Combine Pressure and LWPIV to Obtain a Large-Window Time-Resolved Data Set</i>	162
5.36	<i>Actual POD Modes 1 and 2 Compared with Estimated Mode 1 and 2 Using the sLSE Approach</i>	163
5.37	<i>Error Between the Actual Velocity and the Estimated Velocity Using the sLSE Approach with 5 POD Modes</i>	164
5.38	<i>The Average Error in Velocity as a Function of Modes used in the sLSE Estimate</i>	165
5.39	<i>The Effect of Using a Different Number of Pressure Signals in the sLSE Estimate</i>	166
5.40	<i>Correlation Between the Pressure Sensor at 5 Diameters Downstream of the Jet Nozzle and the Velocity</i>	167
5.41	<i>Correlation Between the Pressure Sensor at 6 Diameters Downstream of the Jet Nozzle and the Velocity</i>	167

5.42	<i>The Ultimate Goal of the Stochastic Modeling, Gappy POD, and Dynamic Gappy POD: Combine Pressure and LWPIV to Obtain a Large-Window Time-Resolved Data Set</i>	169
5.43	<i>Method of Using Stochastic Modeling to Perform Data Fusion using TRPIV and LWPIV</i>	170
5.44	<i>Comparison of the Estimated Mode 1 and Mode 2 Using Stochastic Modeling and the Actual Mode 1 and Mode 2</i>	170
5.45	<i>Method of using the TRPIV Data to Verify the Effectiveness of Gappy POD as an Estimation Technique</i>	172
5.46	<i>Comparison of the Estimated Mode 1 and Mode 2 using Gappy POD and the Actual Mode 1 and Mode 2</i>	173
5.47	<i>Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach with 5 POD Modes in Space</i>	174
5.48	<i>The First 5 POD Modes and Mode 20 for a Mach 0.6 Jet</i>	175
5.49	<i>Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach with 20 POD Modes in Space</i>	175
5.50	<i>Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach with 5 POD Modes in Time</i>	177
5.51	<i>Fluctuations Larger than 5% per 100 Snapshots Using the Gappy POD Approach</i>	177
5.52	<i>Convergence of the Gappy POD Estimation Technique for the Jet Data using Only 50 Snapshots</i>	179
5.53	<i>Computational Time to Run the Gappy POD Estimation Technique for the Jet Data using Only 50 Snapshots</i>	179
5.54	<i>Correlation Between the POD Modes Calculated from the Large-Window PIV and the Time-Resolved PIV</i>	180
5.55	<i>Error Between the Estimated Velocity using LWPIV Modes and the Actual Velocity using Differing Number of Modes</i>	181
5.56	<i>Diagram for the Dynamic Gappy POD Technique</i>	183
5.57	<i>Comparison of the Estimated Mode 1 and Mode 2 using Dynamic Gappy POD and the Actual Mode 1 and Mode 2</i>	184
5.58	<i>Error Between the Actual Velocity and the Estimated Velocity using the Dynamic Gappy POD Approach with 5 POD Modes</i>	185
5.59	<i>Fluctuations Larger than 5% per 100 Snapshots using the Dynamic Gappy POD Approach</i>	185
5.60	<i>Error for Different Estimation Techniques for a Mach 0.6 Jet Flow</i>	187
5.61	<i>Comparison of Fluctuations Larger than 5% per 100 Snapshots using the Gappy POD Approach and the Dynamic Gappy POD Approach</i>	187

5.62	<i>Comparison of Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach and the Dynamic Gappy POD Approach with 5 POD Modes in Space at $r/D=0.8$</i>	188
6.1	<i>Overview of the Fused Techniques</i>	192
6.2	<i>Overview of the Fusion Test Cases</i>	193
6.3	<i>Wavy POD</i>	198
A.1	<i>Diagram of a Stationary and Moving Shock</i>	201
A.2	<i>Diagram of a Shock Tube Before the Shock Wave is Released, Before the Wave is Reflected, and After the Wave is Reflected</i>	202
A.3	<i>Control Volume Around a Shock</i>	202
A.4	<i>Schematic of the Syracuse University Shock Tube</i>	205
A.5	<i>Pressure Measurements at the Two Transducers</i>	206
A.6	<i>Computational Pressure Measurements at the Two Pressure Transducers</i>	208
A.7	<i>Comparison of Pressure Measurements obtained using CFD and Pressure Transducers</i>	209
A.8	<i>Level Zero Data Fusion for the Shock Tube Data</i>	211
A.9	<i>Fusion by Changing the Pressure Sensor Location</i>	211
A.10	<i>Stochastic Model Applied to the Shock Tube Data</i>	213
A.11	<i>Kalman Filter Applied to the Shock Tube Data</i>	213
A.12	<i>Gappy POD Fusion using a CFD Basis and Two Pressure Transducers</i>	214
A.13	<i>Gappy POD Fusion using a CFD Basis and Two Pressure Transducers and 5 CFD Virtual Sensors</i>	215
A.14	<i>Gappy POD Fusion using a CFD Basis and Two Pressure Transducers and 10 CFD Virtual Sensors</i>	216

List of Tables

1.1	Strength and Weaknesses of Current Measurement Techniques	12
5.1	<i>Effect of Modal Convergence on the Accuracy of Fused POD Estimates</i>	144
A.1	<i>Comparison of Shock Speed Predicted by CFD and Pressure Transducers</i>	209
A.2	<i>Comparison of Shock Speed Predicted by CFD, Theory, and Pressure Transducers After Data Fusion</i>	212
A.3	<i>Difference in Shock Speed Predicted by CFD and Pressure Transducers After Data Fusion</i>	212

Nomenclature

$[\hat{S}_{ppa_n}(f)]$	Cross spectral tensor for pressure-POD coefficient
$[\hat{S}_{ppp_k}(f)]$	Cross spectral tensor for pressure
\bar{U}_i	Part of the secondary data not contained in the baseline information
δ_{ij}	Dirac delta
η	Grid coordinate
$\frac{da_n}{dt}(t)$	Time derivative of the POD coefficient
$\hat{a}_n(f)$	Estimated POD coefficient in frequency space
$\hat{B}_j^n(f)$	Linear mapping coefficient in frequency space
$\hat{p}_j(f)$	Pressure in frequency space
λ	Eigenvalue/energy
\langle, \rangle	Averaging/inner product
μ	Kinematic viscosity
ν	Dynamic viscosity
$\overline{u'_i u'_j}$	Reynolds stress tensor
$\overrightarrow{u(x, t)}$	Fluctuating velocity field
$\Phi^{(b)}$	Baseline basis functions for split POD
$\phi^{(n)}(x)$	Basis function
ϕ_i^h	Large window eigenfunctions evaluated in the small window
ϕ'_n	Eigenfunctions evaluated from the small window

ρ	Density
σ	Standard deviation
σ_s	Wavelet scale
τ	Time lag
τ_{ij}	Shear stress
θ	Azimuthal direction
θ	Slope of the camber line
$\tilde{a}_n(t)$	Estimate of the time-dependent POD coefficient
\tilde{g}	Estimate of the gappy field
ξ	Grid coordinate
A_{ij}	Linear mapping coefficient
A_{ni}	Linear POD mapping coefficient
a_n	Fourier cosine coefficient
$a_n(t)$	Time dependent POD coefficients
a_o	Mean Fourier/POD coefficient
b_{ij}	Projection coefficient
B_{nij}	Quadratic mapping coefficient
b_n	Fourier sine coefficient
c	Chord
$C(t, t')$	Temporal correlation tensor
C_{DES}	DES switching coefficient
c_o	Propagation speed/speed of sound
C_{SGS}	LES coefficient
D	Jet diameter
d_{ij}	Distance from the diagonal

E	Conditional event
e	Energy
$F(k)$	State space model
f_o	Fundamental frequency
g	Gappy field
$G(\mathbf{x}, \mathbf{x}', \Delta)$	LES filter function
$G(k)$	Model for the input
$H(k)$	Measurement matrix
L_{ni}	Linear mapping term for Galerkin modeling
L_t	Integral length scale
m	Maximum camber
M_b	Number of baseline modes
$mask_i^t$	Mask vector for gappy POD
μ_{SGS}	Sub-grid scale viscosity
N	Total number of modes
n	Mode number
n_m	Number of modes used to reconstruct
N_{ROM}	Number of modes used in the low-dimensional model
N_s	Number of samples
P	Pressure
p	Location of maximum camber
$Q(k)$	Process noise
Q_{njk}	Quadratic mapping term for Galerkin modeling
r	Radial direction
$R(k+1)$	Measurement noise

R_{Dk}	Information in the diagonal direction
R_{Hk}	Information in the horizontal direction
$R_{i,j}(x, x')$	Two-point spatial correlation tensor
r_{ij}	Correlation
R_{k-1}	Compressed image
R_{Vk}	Information in the vertical direction
T	Period
t	Time
t_a	Airfoil thickness
T_s	Number of snapshots
$u(\vec{x}, t)$	Streamwise component of velocity (“x” direction)
$U(k)$	Model input
$U_{f_{ij}(t)}$	Estimated velocity
U_{max}	Maximum velocity
U_{min}	Minimum velocity
$U_{o_{ij}(t)}$	Actual velocity
$v(\vec{x}, t)$	Velocity in the “y” direction
$V(k + 1)$	Residual
$w(\vec{x}, t)$	Velocity in the “z” direction
$W(\sigma, t - t')$	Wavelet function
$W(k + 1)$	Optimal gain
X_L	X-Coordinate for the lower surface
X_U	X-Coordinate for the upper surface
y_c	Camber line
Y_L	Y-Coordinate for the lower surface

y_t	Airfoil thickness at a point
Y_U	Y-Coordinate for the upper surface
z	Axial direction
$Z(k + 1)$	Actual measurement

Abbreviations

AOA	=	Angle of Attack
CCD	=	Charge Coupled Device
CFD	=	Computational Fluid Dynamics
DES	=	Detached Eddy Simulation
DGP	=	Dynamic Gappy POD
DNS	=	Direct Numerical Simulation
FP	=	Fused POD
JDL	=	Joint Directors of Laboratories
LDA	=	Laser Doppler Anemometry
LDV	=	Laser Doppler Velocimetry
LES	=	Large Eddy Simulation
LSE	=	Linear Stochastic Estimation
LWPIV	=	Large-Window Particle Image Velocimetry
LQE	=	Linear Quadratic Estimation
mtLSE	=	Multi-time Linear Stochastic Estimation
NACA	=	National Advisory Committee for Aeronautics
Nd:YAG	=	Neodymium-doped Yttrium Aluminum Garnet
PIV	=	Particle Image Velocimetry
POD	=	Proper Orthogonal Decomposition
QSE	=	Quadratic Stochastic Estimation
RANS	=	Reynolds Average Navier-Stokes
sLSE	=	Spectral Linear Stochastic Estimation
TFI	=	Transfinite Interpolation
TRPIV	=	Time-Resolved Particle Image Velocimetry
WP	=	Wavy POD

Chapter 1

Introduction

With the improvements in data collection techniques in the field of fluid dynamics, very large data sets are obtained. Determining how to use the data efficiently to extract information is imperative given the large amounts of data for an experiment. For a given test case, data can come from multiple sources such as particle image velocimetry (PIV) [2, 3], pressure sensors, computational fluid dynamics (CFD)[4], and acoustic measurements. It is a challenging task to extract information from the typical amount of data collected for a given experiment. One needs to apply a vast number of analyses to fully comprehend the meaning of this data, which can be time consuming and tedious.

1.1 Objective

The objective of this work is to use data more efficiently and extract as much information as possible from the given data. This is accomplished using the idea of fusion. Data and technique fusion are two different types of fusion used to obtain this goal. Data fusion is the combination of data from multiple or single sources. This idea has been used successfully in other fields and is surveyed throughout [section 1.2](#) to learn how others have used data fusion in the past. The strengths and weaknesses of the

different data sources to be fused are then discussed in [section 1.3](#).

The other type of fusion, technique fusion, is the combination of multiple techniques. This idea uses the strengths of different techniques to overcome the weaknesses in those techniques. The main techniques are explained in [section 1.4](#) and are fused together in [subsection 4.6.1](#) and [subsection 5.5.4](#) to perform data fusion.

The large number of data sets and the many techniques can become overwhelming and require an efficient way of handling this. *A secondary objective is to design a framework that is able to efficiently handle large data and perform the analysis techniques.* The framework developed is essential in successfully performing fusion, and is discussed in [chapter 2](#).

1.2 Data Fusion

As stated, ideas from the data fusion community are used to fuse fluid dynamic data. Before applying data fusion to fluid dynamic data, it is important to understand the definition of data fusion and to survey the different applications in which data fusion has been successful.

To start, the idea of data fusion must be defined, but a formal definition is difficult to find. Hall and Llinas [5] states data fusion, data integration, and sensor fusion are ideas for taking diverse data sets and combining them to extract information that cannot be obtained from any sensor alone. The Joint Directors of Laboratories (JDL) [6] define data fusion as:

“a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process

itself, to achieve improved results”

In short, data fusion is the process of combining data to refine, estimate, and achieve a greater understanding of the data. However, as stated by Steinberg *et. al.* [6] and Wald [7] there are many different definitions of data fusion. The above just happens to be the most widely accepted and an attempt to homogenize the data fusion community.

The JDL defines five levels where data fusion can be performed. The levels according to the JDL are stated below [6].

- **Level 0** (Sub-Object Data Assessment): estimation and prediction of signal/object observable states on the basis of pixel/signal level data association and characterization
- **Level 1** (Object Assessment): estimation and prediction of entity states on the basis of observation-to-track association, continuous state estimation (e.g. kinematics) and discrete state estimation
- **Level 2** (Situation Assessment): estimation and prediction of relations among entities, to include force structure and cross force relations, communications and perceptual influences, physical context, etc.;
- **Level 3** (Impact Assessment): estimation and prediction of effects on situations of planned or estimated/predicted actions by the participants; to include interactions between action plans of multiple players (e.g. assessing susceptibilities and vulnerabilities to estimated/predicted threat actions given one’s own planned actions);
- **Level 4** (Process Refinement): adaptive data acquisition and processing to support mission objectives

However, these definitions are geared toward target identification and threat assessment . Like many others, such as Synnergren *et. al.* [8], Abdelgawad [9], and

Giacobbe [10]) who have modified the above levels to fit the fields of bioinformatics, wireless networks, and cyber security respectively, the definitions that fit within the field of aerodynamics are:

- **Level 0** is the preprocessing of the data. This consists of using the data to repair itself, such as repairing spurious vectors or particle drop out in a PIV measurement.
- **Level 1** involves looking at relationships between the data to fuse them together. An example of this is using CFD and PIV to create better data.
- **Level 2** involves using the data to extract meaningful physics such as noise sources and coherent structures in the case of jet noise.
- **Level 3** involves determining the type of control to use — closed loop or open loop — suction, blowing or synthetic jet actuators, etc. — and predicting the effects of the control.
- **Level 4** involves applying the control technique, measuring the outputs, and adjusting the control to meet the desired results.

This work focuses on performing data fusion on levels zero, one, and two, which are concerned with merging and interpreting data.

As stated, data fusion techniques have been applied in many fields, including business intelligence [11], navigation [12], and target recognition [13]. The application of data fusion is discussed in these fields in the following section.

1.2.1 Business Intelligence

The first field to discuss in which data fusion has been successfully implemented is business intelligence. Dayal *et al.* [11] discuss the use of data integration in business intelligence, which they state is the tools and methods for collecting and analyzing

data. The example used in the work by Dayal *et al.* is data flow for an online retail company. The retail company collects data that includes customer viewing history, purchase history, inventory, shipping schedules, etc. This data can then be used to determine prices based on availability and demand, or provide offers to customers based on the products they have viewed or purchased. Data integration also helps process this information more quickly and provide real-time feedback, such as dynamic pricing and special offers to customers before purchase. Using data integration, the retail company uses all of this information to serve the customer better, ultimately leading to a more profitable business.

An example of this is amazon.com [14]. If one purchases a computer on amazon.com a week ago and is now viewing keyboards, amazon.com might suggest buying a mouse. The mouse they suggest will be one that is in stock in a warehouse near the customer, which allows the customer to receive the product with minimal delay and shipping cost.

Another area in which data integration is used for business intelligence is in shipping [11]. Shipping is affected by weather, traffic, and product availability. Using all of this data together, a shipping company can determine the best time to send out a shipment and what route to take. In addition, using data integration, a shipping company could dynamically change routes leading to shipping that is more efficient by avoiding heavy traffic, bad weather, and construction.

1.2.2 Navigation

Like business intelligence, data fusion is also successfully used in navigation. One example presented by Herrera *et al.* [12], is the use of data fusion for a personal positioning system (PPS). The system uses multiple sensors to determine the location of an object. The sensors include inertial measurements (accelerometer and gyroscopes), magnetic sensors such as a compass, and a global positioning system (GPS). The inertial measurements have propagation error since they start from an initial

location and use the sensors to estimate the new location based on how the object has moved. Compasses are subject to magnetic fluctuations, and GPS can suffer from signal loss and signal corruption. Fusing this information together using a Kalman filtering technique and a dead reckoning algorithm, one can have a much better estimate of the location of an object [12]. For example, if one is navigating through the woods using only inertial measurements and had to travel 1 mile, then turn 10° , and move another half mile but in reality traveled 0.9 miles, turned 11° , and then traveled 0.6 miles, one would not be where one expected. In fact, all future movements would be effected by these errors. Using the GPS, one can update the position and remove the errors created by the inertial measurements if the GPS signal is reliable enough to be fused.

Sasiadek *et al.* [15] discuss the use of data fusion in navigation as well. In their work, GPS and inertial measurements are used to control an unmanned vehicle. The system uses accelerometer and gyros to navigate, which produces propagation error, but use the GPS to correct the inertial measurements. Sasiadek demonstrates that this method is more robust and accurate than using only one sensor on its own.

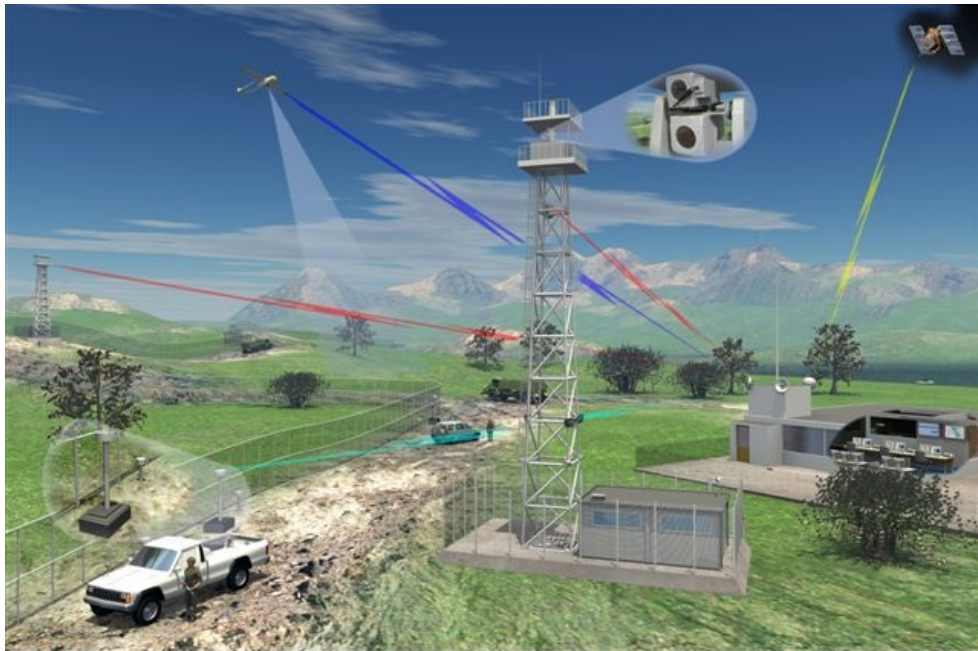
1.2.3 Target Identification

Target identification is another field that benefits from using data fusion. Fay *et al.* [13] discuss target identification by adding color to night vision goggles. They state that targets usually stand out better in color than they do in the standard green light of night vision. The system that Fay *et al.* describe uses combinations of four different sensors (low light camera, short wave infrared camera, long wave infrared camera, and medium wave infrared camera). Each camera contributes complementary and unique information. For example, the short wave is much more adept at picking out camouflage in low light situations than the other sensors [13]. The model they use can also enhance the contrast in such a way that targets stand out. Through the combination of the images, Fay *et al.* show using all four images are better than

using any single image.

Data fusion is also used by Abdel-Aleem *et al.* [16] to locate underground pipes and determine whether the pipe is damaged. The data comes from ultrasound, ground penetrating radar, video data, etc. They show through data fusion, one can determine if damage has occurred, location of the damage, and the type of damage. This is something that could not have been done with any one piece of data without an invasive measurement, which could be expensive and time consuming.

The application of data fusion for target identification is also demonstrated in figure 1.1 [17].



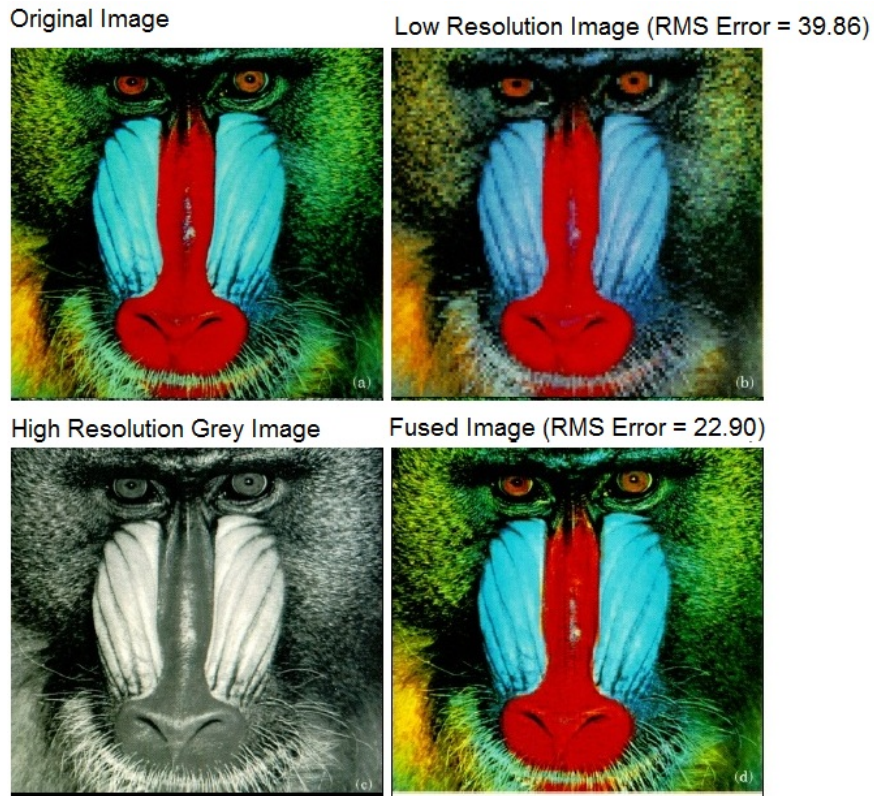
<http://www.ultra-ccs.com/business/bordersecurity>

Figure 1.1: *Data Fusion and Target Identification Used for Security (Image Obtained from [ultra-ccs.com/business/bordersecurity](http://www.ultra-ccs.com/business/bordersecurity))*

The figure demonstrates the use of multiple sensors to detect and locate a threat. Through the fusion of satellite imagery, motion sensors, and airborne and land-based radar, a threat can be identified and understood.

1.2.4 Image Fusion

Data fusion is also used to merge images; a technique to do so is discussed in detail by Yocky [18]. In this work, multiple images are combined into one, more useful image using wavelet transforms (which are discussed in subsection 1.4.6). This technique is useful because one can take advantage of the strengths of the different measurement techniques. In his work, an image (top left image in figure 1.2) was modified such that the spatial resolution was decreased (top right image in figure 1.2) and the color was removed (bottom left image in figure 1.2).



David A. Yocky J. Opt. Soc. Am. A/Vol. 12, No. 9/September 1995

Figure 1.2: *Example of Image Fusion on a High Resolution Gray Scale Image and a Low Resolution Color Image*

Using the wavelet data fusion technique, one can combine the low resolution image and black and white image such that the fused image has both high spatial resolution and color (bottom right of figure 1.2).

Image fusion can also be used for detecting hidden weapons as shown by Varshney [19]. He states the concealed weapon would be difficult to identify using only an infrared sensor or a millimeter wave sensors. The two images can be fused together to take advantage of the penetrating power of the millimeter wave sensor and the higher resolution of the infrared sensor. The fused image clearly shows the concealed weapon. Image fusion is also used by Chen *et al.* [20] for detecting concealed weapons. They used normal light sensors and millimeter wave sensors. These are fused together to give a human operator a better idea of who is holding a weapon and how to identify the target.

The fusion is accomplished using a pyramid method first proposed by Burt [21]. A block diagram for the process is shown in figure 1.3. The algorithm works by high and low pass filtering every row of the original image (R_k) and then removing every other row. This produces two intermediate images. The filtering and downsampling process is applied to the columns of the two intermediate images. The image that was high pass filtered twice produces the details along the diagonal direction of the image (R_{Dk}). High pass filtering the rows and then low pass filtering the columns gives details in the horizontal direction (R_{Hk}). Low pass filtering the rows and the high pass filtering the columns produces an image with detail in the vertical direction (R_{Vk}). Performing the low pass filtering on the rows and the columns gives the compressed image (R_{k-1}). The resolution of all of these images are reduced by a factor of 4. The compressed image is the representation of the original image after the resolution is decreased. The detailed images show the information needed to reconstruct the original image in the horizontal, vertical and diagonal directions.

The process can be repeated for further compressions if R_{k-1} is used as the starting image. The pyramid scheme can be done in reverse to decompress the image and replacing the downsampling with upsampling (add a row/column between every row/column).

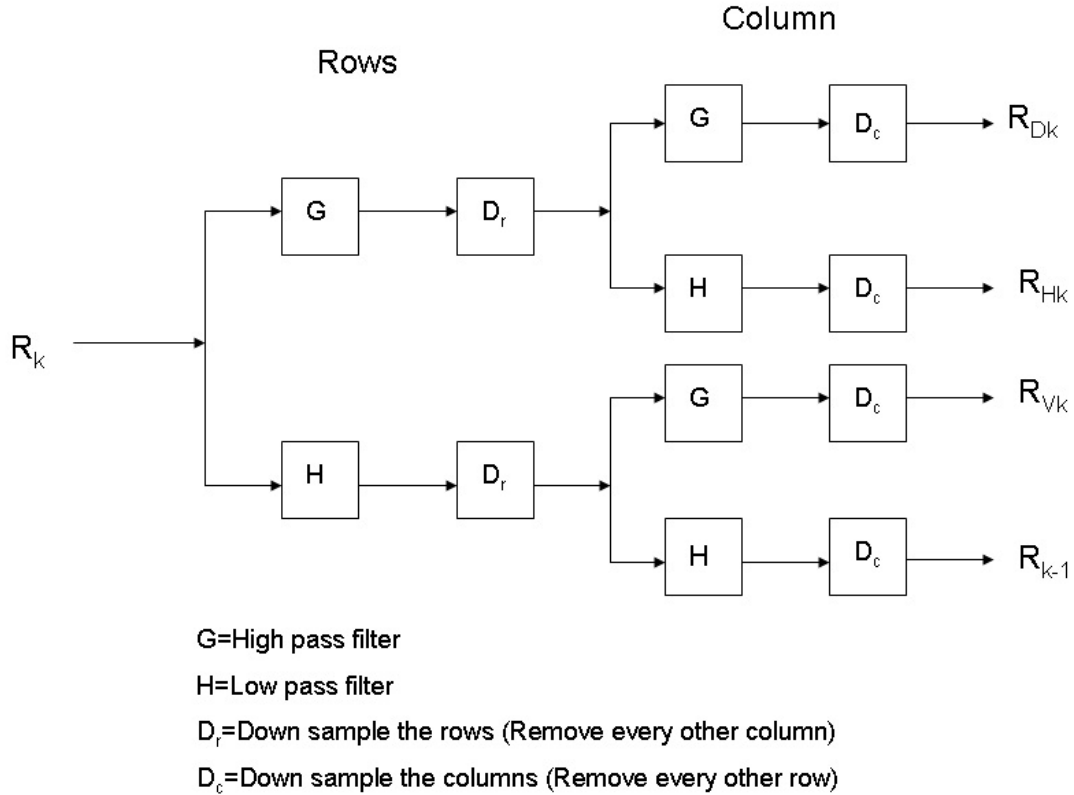


Figure 1.3: Block Diagram for Pyramid Decomposition

A common filtering technique uses the Haar wavelet [22] or the Daubechies wavelets [18]. These wavelets are discussed in subsection 1.4.6.

Using the pyramid technique, multiple representations of the same image are decomposed by filtering and splitting the images into high and low pass information. Then a final image is constructed using low pass information from one image and high pass information from another image. Some techniques for doing this are discussed by Pajares *et al.* [22] and Zhang *et al.* [23]. These include averaging the compressed images, taking the maximum absolute value of each pixel, or taking the maximum energy levels. The technique creates an image that has information from the different images. This process is shown in the diagram in figure 1.4 for one-dimensional data. The process in figure 1.4 works by decomposing the data represented by the largest black box into compressed data and detailed data. The compression process would be

repeated until the resolution of the data is equal to that of the data in the largest red box. Then the compression can be repeated again for both the red and black data. The data are fused by combining the information from the compressed image and the detail information for both images. For the case in figure 1.4 the detail information for the red data is replaced with the detail information from the black data. This is not the only method for doing the fusion; more complex methods have been used in the past to do the fusion as stated earlier. The detailed information from the black data and the compressed red data are then used to decompress the data, which is also the fused data. It must be stressed though, that this fusion process only works when the two data sets are different representations of the same image.

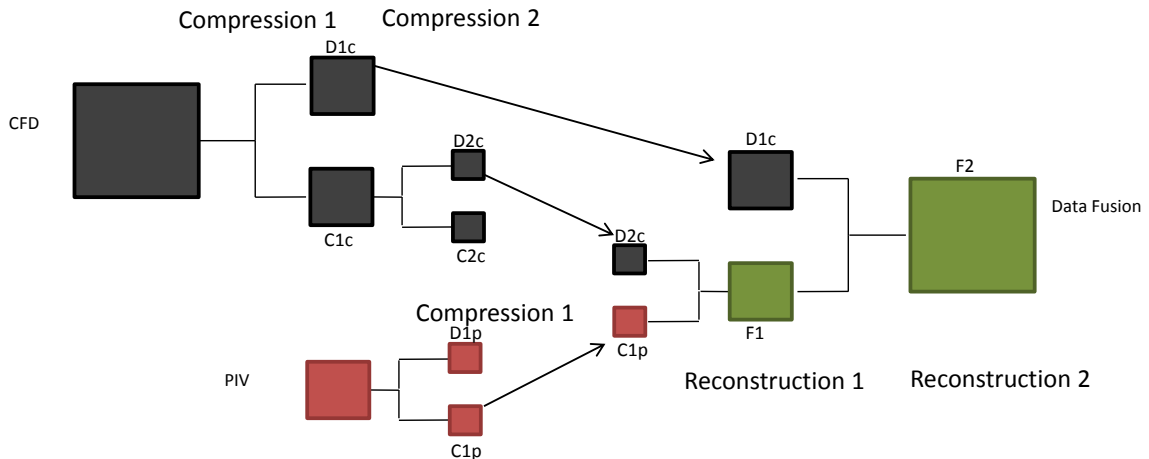


Figure 1.4: *Diagram of the Image Fusion Technique*

1.2.5 Data Fusion in Nature

Data fusion also occurs in nature. According to Hall and Llinas [5] human senses are an example. They describes how all five of our senses work together to keep us safe. Humans use sight, smell, touch, and taste to identify if an object is edible. Using a single sense alone, one may not be able to determine the edibility of the food [5]. Hall and Llinas also use the example of identifying threats through bushes or around corners. Sight may not be able to identify the target, so humans need to use hearing

and smell to determine the threat. Hall and Llinas argue that data fusion is a natural process that humans do all the time without realizing.

1.3 Strengths and Weaknesses of Data Sources

Now that data fusion and its application to other fields has been discussed, one can analyze fluid dynamic data in the context of data fusion. The first step is understanding the different data sources such as time-resolved PIV (TRPIV), large-window PIV (LWPIV), pressure sensors, and computational fluid dynamics (CFD). The data from each source has information that cannot be obtained from the others. Furthermore, each of the techniques have distinct advantages and disadvantages. This work focuses on range (spatial and temporal), resolution (spatial and temporal), and accuracy. Table 1.1 summarizes the strengths and weaknesses of the measurement techniques.

Data Source	Temporal	Spatial	Issues
CFD	Small Δt for convergence, Large domain \rightarrow large computational resources	small Δx for convergence, large spatial domain \rightarrow large computational resources	Modeling error, Dissipation, Dispersion
LWPIV	4Hz sample rate, Statistically long time domain	Multi-point planar data (multiple cameras to extend domain)	Noise, Particle drop out, Reflection
TRPIV	10kHz sample rate, $\mathcal{O}(1s)$ record length	Multi-point planar data (single camera)	Noise, Particle drop out, Reflection
Pressure	$\mathcal{O}(10kHz)$ sample rate $\mathcal{O}(10s)$ record length	Single point data (Measurement at each sensor only)	Noise, Uncertainty in placement

Table 1.1: Strength and Weaknesses of Current Measurement Techniques

The first measurement technique to discuss is the pressure sensor. There are

many different types of pressure sensors that each have different characteristics. The pressure sensor has temporal resolution on the order of 10 – 100kHz and can take a large record length, on the order of 10s depending on the particular sensor. Due to the high sampling rates and large record lengths, the pressure transducer is effective at measuring temporal data. However, pressure sensors are limited in spatial resolution since they only measure pressure at a single location in space. Some spatial resolution can be obtained by using arrays of pressure sensors, but the information would still be sparse. Uncertainty and noise are associated with the measurement.

PIV is another measurement technique, used to obtain flow velocity, by illuminating particles in the flow with a laser sheet and taking an image of the flow at two instances in time to determine the velocity. The two types used in this work are TRPIV and LWPIV. These systems are able to capture data on a two-dimensional plane. There is a tradeoff between resolution and domain size since the number of pixels is predefined based upon the camera. When the camera is focused on a smaller region, the spatial resolution is higher than if the camera is focused on a larger region. One way to increase the domain without sacrificing resolution is to use multiple cameras [24–26]. LWPIV is obtained in this manner. The windows must be stitched together in space, since the data in two adjacent windows does not always match correctly, despite the cameras sampling simultaneously. These differences are caused by the cameras not being aligned perfectly, rotation of the camera, the data being overlapped, and different viewing angles which change the strength of the velocity. Care must be taken when stitching the windows together or the seam may show up in the statistics. An algorithm to find the optimal oversight and scaling was developed by Shea *et al.* [25] to stitch the windows together to eliminate a seam.

The sampling rate of the PIV is low relative to that of the pressure sensor. One way to increase the sampling rate is to use a TRPIV system, which is currently capable of sampling at 10kHz. However, the camera is only able to collect on the order of 1s worth of data, which is an order of magnitude less than the pressure

sensors. Also, TRPIV systems are expensive and using multiple cameras to acquire a larger window is not always feasible due to the cost of the system. The energy in each laser pulse is less than the energy in a standard PIV pulse. This means the laser sheet for the time-resolved system cannot be made as large as the LWPIV system. These two factors limit the domain for the TRPIV measurement.

The PIV system, like the pressure sensors, is also plagued by uncertainty and noise [27] [28]. Usually the uncertainty of the PIV measurement is larger than the uncertainty of a pressure sensor measurement. In addition, seed dropout, reflection, or occlusions are some issues that can corrupt the PIV data. Some of these issues are addressed using data fusion and are discussed throughout this document.

The last data source discussed in this document is CFD, which is a simulation method. The CFD is able to get a full three-dimensional, time-resolved data set. The temporal and spatial resolution can be defined as well as the domain size and temporal range. Increasing the resolution and/or domain in both time and/or space requires more computational power, either longer run times or a “bigger” machine. Therefore, there is a tradeoff between resolution, domain size, and computational cost when using CFD.

One advantage of CFD over experimental data is CFD calculates all values i.e. velocity, pressure, density, etc. at all times and at all locations in the domain, whereas, experiments usually only measures a single quantity. Knowing more of the quantities potentially gives better understanding of the flow physics.

One issue is CFD requires discretization of the Navier-Stokes equations (discussed in [subsection 4.4.2](#)), which introduces errors such as dispersion and dissipation. In addition, turbulence, combustion, non-Newtonian fluids, etc require modeling, unlike the other methods that are actual measurements. In other words, a well set up experiment gives the exact state of the flow within the uncertainty, while the CFD gives a model of the flow field. However, when setting up experiments one needs to be careful when taking intrusive measurements, which can alter the flow.

1.4 Analysis Tools

Now that the idea of data fusion has been discussed as well as the strengths and weaknesses of different fluid dynamic sources, methods of fusing these sources together are required. In this section, the standard tools used in fluid dynamic analysis are discussed. These include proper orthogonal decomposition (POD), linear stochastic estimation (LSE), and wavelets.

1.4.1 Fourier Analysis

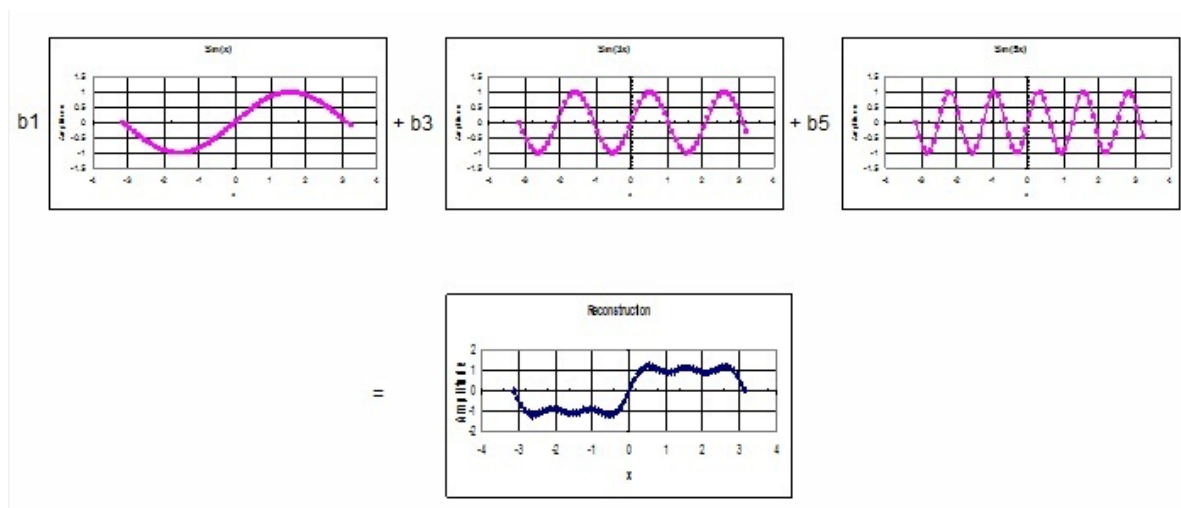


Figure 1.5: *Reconstruction of a Square Wave using Fourier Modes*

The first technique to discuss is Fourier analysis, which works by using a summation of sinusoids with varying coefficients, frequency, and phase to construct a more complicated waveform. This is shown graphically in figure 1.5, where a square wave is created by the summation of sinusoids with different amplitudes and frequencies.

Mathematically, the original signal can be reconstructed using

$$f(x) = a_o + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_o x) + b_n \sin(2\pi n f_o x) \quad (1.1)$$

where $f(x)$ is the original signal, n is the mode number, and

$$a_o = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) dx \quad (1.2)$$

$$a_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos(2\pi n f_o x) dx \quad (1.3)$$

$$b_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin(2\pi n f_o x) dx \quad (1.4)$$

Why do we use Sinusoids?

Sinusoids are used for reconstruction for multiple reasons. One is that when sinusoids are used to decompose a signal, the coefficients represent the frequency content of that signal. Another reason is that sinusoids form a basis, which is required for the reconstruction process. Functions form a basis if they are orthogonal to one another.

$$\int_{-\frac{T}{2}}^{\frac{T}{2}} \phi^{(i)}(x) \phi^{(j)}(x) dx = \delta_{ij} \quad (1.5)$$

Sinusoids might not always be the best basis to use. In the case of electronics where pulse signals and switching functions are used, square waves can be used for reconstruction [29] since the data looks more like a square than a sinusoid. For solutions to equations in cylindrical coordinates, perhaps Bessel functions would be appropriate.

1.4.2 Proper Orthogonal Decomposition

There are so many different basis functions that can be used and choosing the best one may be difficult. This is where POD is useful because it calculate the set of basis functions that are the most efficient in terms of energy.

Links between Fourier and POD

There are many links between Fourier and POD. In some cases, such as homogenous turbulence, POD produces Fourier modes, which means that sinusoids are in fact the best basis representation of the flow field in terms of energy. This occurs in the r - θ plane of an axisymmetric jet [30]. In addition, both techniques decompose a signal into different building blocks, which are usually much easier to understand than the raw data itself and because of this, the most important physics can be determined. Similarly, both techniques allow the signal to be reconstructed using the building blocks as seen in equations 1.6.

$$f(x) = \sum_{n=1}^{n_m} a_n \phi^{(n)}(x) \quad (1.6)$$

Proper Orthogonal Decomposition

POD is a reduced order modeling technique first introduced to fluid dynamics by Lumley in 1967 [31]. It is also known as Karhunen-Loève expansion or principal component analysis depending on the field in which the method is being used. This technique is used in a variety of applications, anywhere from turbulence [31, 32] to facial recognition [33]. POD is a way of building a basis for a flow that is based on the energy content or RMS error, whereas in Fourier analysis the basis is chosen to be a combination of sines and cosines. One may chose Fourier for various reasons, including transforming differential equations into algebraic equations, transforming a convolution into multiplication, or because sines and cosines give the frequency content of a signal. However, as stated previously, POD give the best energy representation of the flow.

The first POD mode is the most energetic, and the energy content of higher modes decrease. The modes (bases) are created by maximizing the mean square projection

of the velocity [24] or in other words trying to find the best fit for the data set using

$$\frac{\langle \overrightarrow{|u(x, t)}, \phi | \rangle^2}{\|\phi\|^2} \quad (1.7)$$

where u is the velocity, ϕ are the eigenvectors, and the inner product ($\langle \overrightarrow{|u(x, t)}, \phi | \rangle$) is defined as

$$\int u(x)\phi^*(x)dx \quad (1.8)$$

where $*$ is the complex conjugate. The maximization problem is solved using calculus of variation, which simplifies to an integral eigenvalue problem where the eigenvectors form the basis.

There are many different variations of POD. One is split POD [34], which looks for differences in the flow structure of two different flows. Another is Filtered POD [35], which filters out high frequency noise. A third is snapshot POD [36], which is a modification to classical POD [31] used to decrease the size of the eigenvalue problem for data sets with more spatial data than temporal data. Gappy POD, which is used to estimate missing data [37], is another variation of POD. The methods for performing some of these techniques are explained in the following sections.

Classical POD

As stated previously, the maximization of the mean squared projection of the velocity simplifies to an eigenvalue problem

$$\int R_{ij}(x, x')\phi_j^{(n)}(x)dx' = \lambda^{(n)}\phi_i^{(n)}(x) \quad (1.9)$$

where ϕ is the eigenvector, λ is the eigenvalue (energy), and $R_{ij}(x, x')$ is the time averaged spatial two-point velocity correlation.

$$R_{ij}(x, x') = \langle u_i(\overrightarrow{x}, t_o), u_j(\overrightarrow{x}', t_o) \rangle \quad (1.10)$$

In equation 1.10, u is the velocity of the flow or any other variable, depending on what is being studied, t_o is time, and x is the position vector. If the flow variable is velocity, the kernel for this problem ($R_{ij}(x, x')$) can be written as a block matrix where $u_1 = u$, $u_2 = v$, and $u_3 = w$

$$\begin{bmatrix} \langle u(\vec{x}, t_o), u(\vec{x}', t_o) \rangle & \langle u(\vec{x}, t_o), v(\vec{x}', t_o) \rangle & \langle u(\vec{x}, t_o), w(\vec{x}', t_o) \rangle \\ \langle v(\vec{x}, t_o), u(\vec{x}', t_o) \rangle & \langle v(\vec{x}, t_o), v(\vec{x}', t_o) \rangle & \langle v(\vec{x}, t_o), w(\vec{x}', t_o) \rangle \\ \langle w(\vec{x}, t_o), u(\vec{x}', t_o) \rangle & \langle w(\vec{x}, t_o), v(\vec{x}', t_o) \rangle & \langle w(\vec{x}, t_o), w(\vec{x}', t_o) \rangle \end{bmatrix} \quad (1.11)$$

Solving for the eigenvalues and eigenvectors of equation 1.11 gives the modes (basis functions) for the flow. The POD coefficients can then be found by projecting the velocity field onto the modes.

$$a_n(t) = \int u_i(\vec{x}, t) \phi_i^{(n)}(\vec{x}') d\vec{x}' \quad (1.12)$$

With the POD modes and the POD coefficients, the flow can be reconstructed

$$u_i(\vec{x}, t) = \sum_{n=1}^{n_m} a_n(t) \phi_i^{(n)}(\vec{x}') \quad (1.13)$$

where n_m is the number of mode used in the reconstruction. The flow field is completely reconstructed if all of the POD modes ($n_m = N$) are used in the reconstruction, where N is the total number of modes.

Snapshot POD

Snapshot POD (a modification of Classical POD) was introduced in 1987 by Sirovich [36]. The method transforms the eigenvalue problem from a spatial problem to a temporal problem. The spatial problem has dimension of the number of spatial points multiplied by number of velocity components and the temporal problem has dimension of temporal points. Snapshot POD is preferred over Classical POD when

analyzing a data set with more spatial data than temporal data, such as two camera PIV [24], LES [38, 39], or DNS data [40]. This method calculates the eigenfunctions using a temporal correlation tensor.

$$\int C(t, t') a_n(t') dt' = \lambda^{(n)} a_n(t) \quad (1.14)$$

In equation 1.14, $a_n(t)$ are the time dependent POD coefficients and $C(t, t')$ is the correlation tensor.

$$C(t, t') = \frac{1}{T_s} \int u_i(\vec{x}, t) u_i(\vec{x}', t) d\vec{x} \quad (1.15)$$

In equation 1.15, T_s is the total number of snapshots and u is the velocity field. Notice in equation 1.15 that the kernel ($C(t, t')$) is now the summation of the autocorrelations of the velocity components in the flow meaning the eigenvalue problem is no longer coupled. The coupling is taken care of when the spatial modes are calculated. The uncoupling of the eigenvalue problem along with doing a temporal correlation instead of a spatial correlation reduces the problem size significantly for data sets with more spatial data than temporal data.

The time dependent POD coefficients are orthogonal and have a magnitude equal to the square root of the eigenvalues, which is consistent with Classical POD.

$$\langle a_n \cdot a_m \rangle = \delta_{mn} \lambda^{(m)} \quad (1.16)$$

The POD modes are calculated by projecting each component of velocity onto the temporal POD coefficients.

$$\phi^{(n)}(\vec{x}) = \frac{1}{T_s \lambda^{(n)}} \int a_n(t) u_i(\vec{x}, t) dt \quad (1.17)$$

The spatial eigenfunctions are orthonormal, as are the spatial eigenfunctions for Classical POD.

Split POD

Another variant of POD is known as Split POD. In this version of POD, one can use two different data sets and find modes that correspond to differences in the data. This has been used by Camphouse *et al.* [34] to determine the effects of flow control. The idea for this came from the fact that the controller is adding very little energy and the control information is dominated by baseline information. Using the split POD procedure, the baseline information can be removed from the control data. The procedure is as follows: perform POD on the baseline data to get basis functions $\Phi^{(b)}$. The control data (secondary data set) is then projected onto the baseline basis functions.

$$b_{ij} = \langle U_i, \Phi_j^{(b)} \rangle \quad (1.18)$$

U_i are snapshots from the secondary data set and \langle, \rangle denotes an inner product.

The part of the secondary data that is not contained in the baseline data is extracted.

$$\bar{U}_i = U_i - \sum_{j=1}^{M_b} b_{ij} \Phi_j^{(b)} \quad (1.19)$$

\bar{U}_i is the part of the secondary data not contained in the baseline information, and M_b is the number of baseline modes to use. POD is then performed on the new data (\bar{U}_i), giving basis functions for information not contained in the baseline data. The secondary data can now be expressed in terms of both baseline and secondary information.

Gappy POD

Another variant of POD, Gappy POD was created by Everson and Sirovich in 1995 [37] to handle “gappy” data. Gappy data is data that has gaps or missing information and the technique estimates this missing information. Everson and Sirovich first demonstrated this technique using marred photographs and repairing them with Gappy POD. Specifically, they showed missing data from human faces could be de-

terminated from a basis formed by similar types of faces. They also observed that if not careful, errors can arise. This was shown by attempting to repair an image of a monkey’s face with a human face basis. This created a repaired image that looked like a human face.

Gappy POD works by performing POD on non-gappy data. The data with missing information is then reconstructed using a linear combination of the POD basis calculated from the non-gappy data. The coefficients of the linear combination are then determined by solving a least squares problem that minimizes the difference between the linear reconstruction and the gappy data. The process is shown pictorially in figure 1.6. The “O” box represents the original data set, which can contain a few gappy snapshots. The non-gappy data (“A”) is then used to generate the POD modes (“ Φ ”) of the data set. The gappy snapshot (“B”) is represented as a linear combination of the POD modes and the coefficients are determined by fitting the linear combination with the actual data. With the coefficients and the modes, the gappy data is estimated and then inserted anywhere there is a hole.

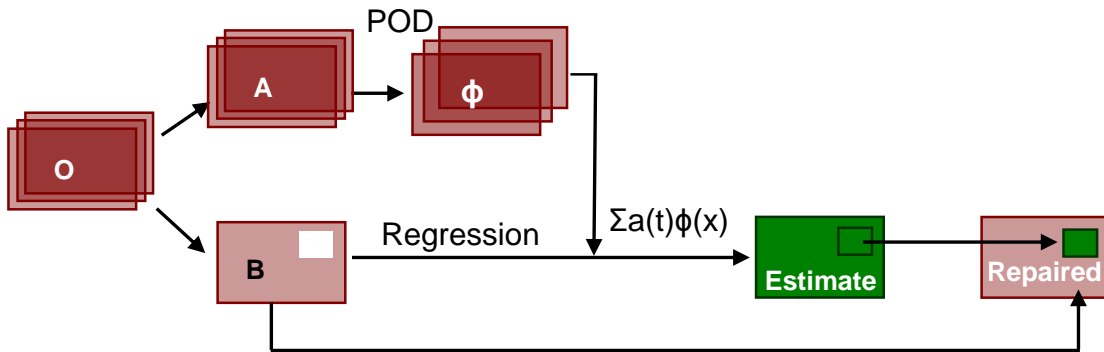


Figure 1.6: *Diagram of the Gappy POD Method*

The mathematical formulation of Gappy POD is presented below. First, the gappy points must be located and stored in a mask vector. The mask vector is defined such that if the data is missing or corrupt the mask is zero and if the data is not missing

it is one.

$$mask_i^t = \begin{cases} 1 & \text{if data is not missing} \\ 0 & \text{if data is missing} \end{cases} \quad (1.20)$$

With the mask vector, a ‘‘gappy inner product’’ is defined as:

$$\langle u, v \rangle_{gappy} = \langle \langle a, mask \rangle, \langle b, mask \rangle \rangle \quad (1.21)$$

In equation 1.21, a and b are two vectors. The estimate of the gappy data is then defined as a linear combination of the non-gappy modes.

$$\tilde{g} = \sum_{n=1}^{n_m} b_n \phi^{(n)} \quad (1.22)$$

In equation 1.22, b_n is the linear combination coefficient, $\phi^{(n)}$ is the POD basis from the non-gappy data and \tilde{g} is the POD estimated gappy data. Then the error between the estimate and original is minimized at the non-gappy points.

$$\langle g - \tilde{g}, g - \tilde{g} \rangle_{gappy} \quad (1.23)$$

In equation 1.23, g is the original gappy data. The minimization problem is solved by taking the derivative of the system with respect to b_i and setting it equal to zero. This produces the following linear system that can be solved for b_i .

$$\langle \phi^{(i)}, \phi^{(j)} \rangle_{gappy} b_i = \langle g, \phi^{(i)} \rangle_{gappy} \quad (1.24)$$

Now, the repaired data can be constructed using the following equation.

$$g = \begin{cases} g & \text{if mask} = 1 \\ \tilde{g} & \text{if mask} = 0 \end{cases} \quad (1.25)$$

Iterative Gappy POD

In addition to repairing a single marred image, Gappy POD can be extended to repair a set of marred images using an iterative process. Notice the differences between figure 1.7 and figure 1.6. In the iterative version of Gappy POD (figure 1.7), all of the data is gappy, but the gaps are distributed in time and space and therefore all of the data (“O”) is said to be gappy data (“B”). The missing data in “B” is estimated to be a spatial average which produces the data in “A”. POD is performed on the data in “A” and the linear combination coefficients are calculated using equation 1.24. The estimated data is then inserted into the gaps using equation 1.25 and the process repeats until the data converges.

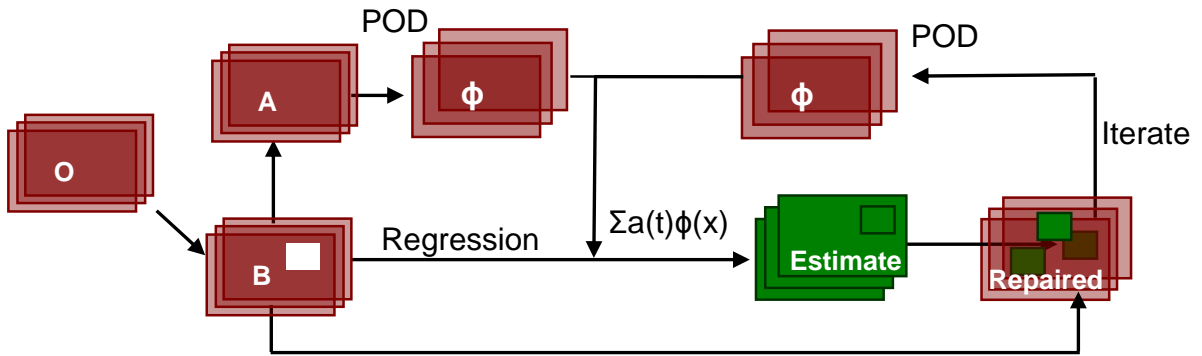


Figure 1.7: *Diagram of the Iterative Gappy POD Method*

Gappy POD has been used by many different researchers in the past to repair fluid dynamic data and is discussed in the following paragraphs. Gappy POD was first used for aerodynamic applications by Bui-Thanh *et al.* [41] to reconstruct a flow field based on airfoil surface data and to design optimal airfoils. The Gappy POD method was successfully used to determine the shape of an airfoil given a desired pressure distribution. Similar work was performed by Wilcox [42].

Ruscher *et al.* [43] also applied Gappy POD to fluid dynamic data to perform POD on a set of snapshots with an airfoil moving through the image. This work attempted to estimate the POD modes around a plunging airfoil using different techniques. It

was determined that Gappy POD produced the most accurate results, but other techniques could be performed orders of magnitude faster with little difference in the results. However, one should note that the data was generated using a source panel method, so the faster techniques may not work as well with experimental or high fidelity CFD data.

Gappy POD was also used by Murray and Ukeiley [44] to repair marred PIV images in a cavity flow. They saw that using more modes decreased the error in the estimation. In fact, it was demonstrated that the error in the estimate approached the experimental uncertainty as more modes were used. Ultimately they concluded that Gappy POD had improved their data so they could increase their confidence in the calculated POD modes.

Raben *et al.* [45] made modifications to slightly improve the results of Gappy POD by using an adaptive method to determine the optimal number of modes to use locally. Using this adaptive method, Raben *et al.* were able to increase the accuracy of the estimated data. This new technique allowed one to estimate the large scale structures and the small scale structures separately, which was shown to significantly reduce the estimation error. All of these works demonstrate that Gappy POD is an effective method to estimate missing fluid dynamic data.

1.4.3 Stochastic Estimation

Another technique is stochastic estimation, which was first proposed by Adrian in 1977 [46]. In this work, Adrian shows that the velocity at one point can be estimated from information at another point. Adrian states that the data at a point is a function of time (t), position (x), and conditional event E .

$$u(x, t) = F(E, x, t) \tag{1.26}$$

When given E , the best mean square representation is given by a conditional average [47]. The conditional average can be estimated as a linear combination of the conditional data. Minimizing the error between the actual velocity and the linear estimate produces the following equation, as stated by Bonnet *et al.* [48].

$$u_i(x') = A_{ij}(x')u_j(x) \quad (1.27)$$

In equation 1.27, A_{ij} is given by the following relation: $\overline{u_j(x)u_k(x)}A_{ij} = \overline{u_j(x)u_i(x')}$ where $\overline{u_j(x)u_k(x)}$ is the Reynolds stress and $\overline{u_j(x)u_i(x')}$ is the two-point correlation. This technique was expanded upon by Bonnet *et al.* [48] by performing POD and then using another signal to estimate the POD coefficients. Then the field is rebuilt using the POD basis and estimated coefficients. The modified stochastic estimation, as it is called, requires that the conditional signal be well correlated with the POD coefficients. The technique works by assuming the POD coefficients can be written as a power series expansion of the conditional signal. This was further expanded by Taylor *et al.* to use pressure as the conditional signal [49].

$$\tilde{a}_n(t) \approx A_{ni}P_i(t) + B_{nj}P_j(t)P_k(t) + \dots \quad (1.28)$$

The coefficients (A_{ni} , B_{nij} , ...) are determined by minimizing the mean square error between the estimate and actual value of the POD coefficients.

$$(\tilde{a}_n(t) - a_n(t))^2 \quad (1.29)$$

The power series estimate is then truncated. If only the first term is used, the estimate is known as linear stochastic estimation (LSE). If the first two terms are used, the estimate is known as quadratic stochastic estimation (QSE). Naguib *et al.* [50] demonstrated that keeping the quadratic term in the estimation was extremely

useful when using pressure to estimate velocity. Murray and Ukeiley [51] also showed this result for a cavity flow. However, Tung *et al.* [52] shows not much information is gained from higher order estimates, so this section only focuses on the linear and quadratic estimation. Using the linear truncation, the minimization problem reduces to a linear set of equations

$$\begin{bmatrix} \langle P_1 P_1 \rangle & \dots & \langle P_q P_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle P_1 P_q \rangle & \dots & \langle P_q P_q \rangle \end{bmatrix} \begin{bmatrix} A_{n1} \\ \vdots \\ A_{nq} \end{bmatrix} = \begin{bmatrix} \langle a_n P_1 \rangle \\ \vdots \\ \langle a_n P_q \rangle \end{bmatrix} \quad (1.30)$$

where $\langle P_i P_j \rangle$ are the pressure-pressure correlations, A_{ni} are the linear mapping coefficients, and $\langle a_n P_i \rangle$ are the pressure-POD correlations. Using the quadratic truncation, the minimization problem reduces to another system of linear equation.

$$\begin{bmatrix} \langle P_1 P_1 \rangle & \dots & \langle P_q P_1 \rangle & \langle P_1 P_1 P_1 \rangle & \dots & \langle P_q P_q P_1 \rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle P_1 P_q \rangle & \dots & \langle P_q P_q \rangle & \langle P_1 P_1 P_q \rangle & \dots & \langle P_q P_q P_q \rangle \\ \langle P_1 P_1 P_1 \rangle & \dots & \langle P_q P_1 P_1 \rangle & \langle P_1 P_1 P_1 P_1 \rangle & \dots & \langle P_q P_q P_1 P_1 \rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle P_1 P_j P_k \rangle & \ddots & \langle P_q P_j P_k \rangle & \langle P_1 P_1 P_j P_k \rangle & \ddots & \langle P_q P_q P_j P_k \rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle P_1 P_q P_q \rangle & \dots & \langle P_q P_q P_q \rangle & \langle P_1 P_1 P_q P_q \rangle & \ddots & \langle P_q P_q P_q P_q \rangle \end{bmatrix} = \begin{bmatrix} A_{n1} \\ \vdots \\ A_{nq} \\ B_{n11} \\ \vdots \\ B_{nj k} \\ \vdots \\ B_{nqq} \end{bmatrix} \begin{bmatrix} \langle a_n P_1 \rangle \\ \vdots \\ \langle a_n P_q \rangle \\ \langle a_n P_1 P_1 \rangle \\ \vdots \\ \langle a_n P_j P_k \rangle \\ \vdots \\ \langle a_n P_q P_q \rangle \end{bmatrix} \quad (1.31)$$

Multi-Time Stochastic Estimation

The LSE/QSE techniques can be limited in complex flows. The technique is improved by using the multi-time stochastic estimation (mtLSE), which uses multiple lags instead of using a single lag. The multi-time estimation takes advantage of the fact that

complex flows will have many different lags. If the appropriate lag is used, the correlation between the conditional signal and the signal being estimated is maximized. The improvement between the single-time correlation and multi-time correlation is shown in chapter 5.5. However, when using the mtLSE algorithm one needs to assign the lags a priori and is limited to a set number of lags. The estimation is performed using

$$\tilde{a}_n(t) = \sum_{k=1}^{N_m} B_j^n(\tau_k) p_j(t - \tau_k) \quad (1.32)$$

where $\tilde{a}_n(t)$ is the estimated POD coefficient, τ_k is the lag, p_j is the conditional signal, and B_j^n are the mapping coefficients.

This method was used by Tu *et al.* [53] for the flow behind a cylinder collected using a TRPIV system. They used the time resolved data to directly compare the stochastic estimation to the actual data and showed the estimation correctly predicted the flow field. This technique was also used by Ukeiley *et al.* [54] for a cavity flow. Using the technique, they were able to get an estimate of the time resolved velocity field. Durgesh and Naughton [55] applied this technique to a high-Reynolds number flow in the wake of a bluff body. They determined using the multi-time approach the flow dynamics could be estimated more accurately than with the single-time estimation. The work also showed, to perform the technique efficiently, one must determine the optimal time lags to use in the estimation.

Spectral Linear Stochastic Estimation

To deal with the issue of determining the optimal lags a priori, one can take the limit as the number of lags approaches infinity and use all lags. This then transforms equation 1.32 into a convolution.

$$\tilde{a}_n(t) = \int_{-\infty}^{\infty} B_j^n(\tau) p_j(t - \tau) d\tau \quad (1.33)$$

As suggested by Ewing and Citriniti [56] for LSE and Tinney *et al.* [57] for mLSE, if a Fourier transform is applied to both sides of equation 1.33, the convolution become a multiplication and results in the following equation:

$$\hat{a}_n(f) = \hat{B}_j^n(f)\hat{p}_j(f) \quad (1.34)$$

where “ $\hat{}$ ” represents the Fourier transform. One should note that the spectral coefficient is a function of frequency, meaning the equations need to be solved at every frequency. The process to solve for the spectral coefficients is similar to the process for the mLSE. The final solution for the spectral coefficients is:

$$\hat{B}_j^n(f) = [\hat{S}_{ppk}(f)]^{-1}[\hat{S}_{pa_n}(f)] \quad (1.35)$$

where $[\hat{S}_{ppk}(f)]$ is the cross spectral tensor for pressure

$$\hat{S}_{ppk}(f) = \lim_{T \rightarrow \infty} \frac{1}{T} \langle \hat{p}_p^*(f), \hat{p}_k(f) \rangle \quad (1.36)$$

and $[\hat{S}_{pa_n}(f)]$ is the cross spectral tensor for pressure and POD coefficients.

$$\hat{S}_{pa_n}^*(f) = \int_{-\infty}^{\infty} \langle a_n(t_o), p_p(t_o + \tau) \rangle e^{-2i\pi f\tau} d\tau \quad (1.37)$$

One should note that the pressure-velocity correlation (equation 1.37) is done in the temporal domain and not spectral domain like the pressure-pressure correlation. The pressure is obtained with a sampling rate on the order of 40kHz but the velocity is obtained with a sampling rate of around 4Hz due to limitations of non-time-resolved PIV systems. Since the velocity is not time-resolved for most problems, one cannot obtain the frequency content of the velocity data. This means the spectrum cannot be computed using spectral methods; it must be calculated in the time domain by lagging the pressure, which is time-resolved.

The stochastic estimation technique has been used in many different applications. Adrian [58] studied the application of stochastic estimation for many flows including plane and axisymmetric shear layers, pipe flow, and grid turbulence. The modified technique was used by Bonnet *et al.* [48] to estimate the shear layer in a jet. The measurements were obtained using crossed hot wires. The hot wires collect time resolved velocity measurements, which were used to validate the stochastic estimation technique in the shear layer of a jet. Stochastic estimation was also used by Gutmark *et al.* [59] to estimate the flow field in a swirling jet. They used stochastic estimation to reconstruct the entire field using hot wires and pressure sensors. The technique proved useful in identifying large structures in the flow. Another example is work done by Pinier *et al.* [24]. They used surface pressure measurements, which were time resolved, to estimate a time resolved velocity field from non-time resolved PIV. The time-resolved velocity estimate was then used to create a control system to help reduce separation on the surface of the airfoil. Tinney *et al.* [60] used the spectral method to generate an estimate of a time-resolved, 3-D velocity field for an axisymmetric jet. They used non-time-resolved PIV and time-resolved near-field pressure to obtain the time resolved estimate. The time resolved velocity was then used to obtain an estimate of the time-resolved far field noise to be compared with the actual noise measurements. The Spectral stochastic estimation was also used by Hall and Ewing [61] for a wall jet, and Garcia-Sagrado and Hynes [62] for a NACA0012 airfoil. The many version of the stochastic estimation have been used by multiple researchers to estimate data for different fluid dynamic applications.

1.4.4 Modeling

Another technique is modeling, which is usually used in flow control applications to determine how the flow evolves in time. Two typical types are Galerkin and stochastic models. Both attempt to determine an equation that shows how the flow changes in time. A Galerkin model estimates the change in time based on the current flow

characteristics with the following equation:

$$\frac{da_n}{dt}(t) = \sum_{i=1}^{N_{ROM}} L_{ni}a_i(t) + \sum_{j=1}^{N_{ROM}} \sum_{k=j}^{N_{ROM}} Q_{njk}a_k \quad (1.38)$$

where N_{ROM} is the number of modes used in the reconstruction, L_{ni} is the linear mapping term, Q_{njk} is the quadratic mapping term, and $\frac{da_n}{dt}(t)$ is the time derivative of the POD coefficients. This equation is derived by projecting the Navier-Stokes equation onto the POD modes, thus creating a low-dimensional version of the equation. The coefficients L_{ni} and Q_{njk} can be derived mathematically or can be determined by training the model. To train the models, data is collected and, then the time rate of change ($\frac{da_n}{dt}(t)$) is calculated from either time-resolved data or an estimate of time resolved data. The coefficients are then determined by solving a least squares problem to minimize the error between the actual data and the modeled data. This type of modeling has been done by Pinier *et al.* [24] for an airfoil flow. They created the model by first estimating the time-resolved POD coefficients using LSE. The coefficients were then used to create the model which was used to reduce the separation on the airfoil. Modeling was also done by Aubry *et al.* [63] for a turbulent boundary layer. In this work they were able to capture intermittencies in the flow structures using the model. From that they could capture the events that are associated with the intermittent structures. Rempfer [64] used Galerkin modeling for a transitional boundary layer. Rempfer used the model to study the different structures occurring at different downstream locations in the transitional boundary layer. Ukeiley *et al.* used modeling for a mixing layer [65], and many others [66–70] have also used this modeling technique.

As stated by Tu *et al.* [53] the stochastic modeling approach is applied for experimental data, because the Galerkin approach can diverge when experimental data is noisy. The stochastic model attempts to model the flow by directly calculating the

flow field based on the current state using the following equation:

$$a_n(t + dt) = \sum_{i=1}^{N_{ROM}} A_{ni} a_i + \sum_{j=1}^{N_{ROM}} \sum_{k=j}^{N_{ROM}} B_{njk} a_j(t) a_k(t) \quad (1.39)$$

where A_{ni} is the linear mapping term and B_{njk} is the quadratic mapping term. Tu *et al.* [53] used this technique to build a model for the flow behind a cylinder.

These modeling techniques work well over a short period of time, but start to diverge over a long period of time (this will be discussed in [section 5.5](#) in more detail). The flow field either goes to zero or diverges to plus or minus infinity. As such, these models need input to help keep them stable. This is generally done using a Kalman filter. [53, 67]

1.4.5 Kalman Filtering

A Kalman Filter [71] or linear quadratic estimation (LQE) is an algorithm that uses multiple noisy measurements to make estimates of the measurement with more accuracy than any of the measurements alone. The Kalman filter works by estimating the state of a system at a point in time and then applies the filter to the data based on the estimated uncertainty. The filtered value should be better than the model or any of the noisy measurements. This is a common data fusion tool. The Kalman filter algorithm is as follows: Predict the state using

$$\hat{X}(k + 1|k) = F(k)\hat{X}(k|k) + G(k)U(k) \quad (1.40)$$

where $F(k)$ is the model for the state ($\hat{X}(k|k)$), $\hat{X}(k + 1|k)$ is the predicted state, and $G(k)$ is the model for the input ($U(k)$). The measurement is predicted using

$$\hat{Z}(k + 1|k) = H(k)\hat{X}(k + 1|k) \quad (1.41)$$

where $H(k)$ is the measurement matrix and $\hat{Z}(k+1|k)$ is the predicted measurement. The residual between the actual measurement and predicted measurement

$$V(k+1) = Z(k+1) - \hat{Z}(k+1|k) \quad (1.42)$$

is calculated where $V(k+1)$ is the residual and $Z(k+1)$ is the actual measurement. The covariance is predicted at the next state by

$$P(k+1|k) = F(k)P(k|k)F(k)^T + Q(k) \quad (1.43)$$

where $P(k+1|k)$ is the estimate of the covariance at the next time, $P(k|k)$ is the current covariance, and $Q(k)$ is the process noise. The residual covariance is calculated by

$$S(k+1|k) = H(k+1)P(k+1|k)H(k+1)^T + R(k+1) \quad (1.44)$$

where $S(k+1|k)$ is the estimate of the residual covariance and $R(k+1)$ is the measurement noise. The optimal gain is then calculated

$$W(k+1) = P(k+1|k)H(k+1)^T S(k+1|k)^{-1} \quad (1.45)$$

where $W(k+1)$ is the optimal gain. The covariance is predicted for the next state

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1|k)W(k+1)^T \quad (1.46)$$

where $P(k+1|k+1)$ is the covariance at the next time step. Finally the state is estimated using

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + W(k+1)V(k+1) \quad (1.47)$$

where $\hat{X}(k+1|k+1)$ is the filtered estimate of the state.

This technique can only be applied to linear systems such as the Galerkin and stochastic models. Some successful attempts have been made to extend the Kalman filter to non-linear applications such as the Extended Kalman Filter (uses Taylor series to linearize the model), Ensemble Kalman Filter [72] which use previous time data to calculate the covariances, etc. These filters have been used in many applications such as navigation [73, 74], and aerodynamic control [53, 67].

1.4.6 Wavelets

Continuous Wavelets

Wavelet decomposition is a technique that has been used in signal processing in order to compress signals, but has also been used as a way of analyzing different scales in turbulent flows [75]. This tool has been used to extract coherent structures in turbulence such as work done by Lewalle *et al.* [76], Meneveau [77], Chainais [78], and Camussi [79].

There are many different wavelet functions. A couple are the Mexican Hat wavelet (figure 1.8)

$$W(\sigma, t) = \left(\frac{t^2}{4\sigma_s^2} - \frac{1}{2\sigma_s} \right) \frac{1}{2\sqrt{\pi}\sigma_s} e^{-\frac{t^2}{4\sigma_s}} \quad (1.48)$$

and the Morlet wavelet (figure 1.9).

$$W(t) = (e^{2i\pi t} - e^{-z_0^2}) e^{-2\pi^2 t^2 / z_0^2} \quad (1.49)$$

These are two common wavelets used in the analysis of fluid dynamic data.

Notice that the Mexican hat wavelet has one large hump and two smaller humps. This feature allows this particular wavelet to have good time resolution. The Morlet wavelet has many humps, which gives the wavelet good frequency resolution.

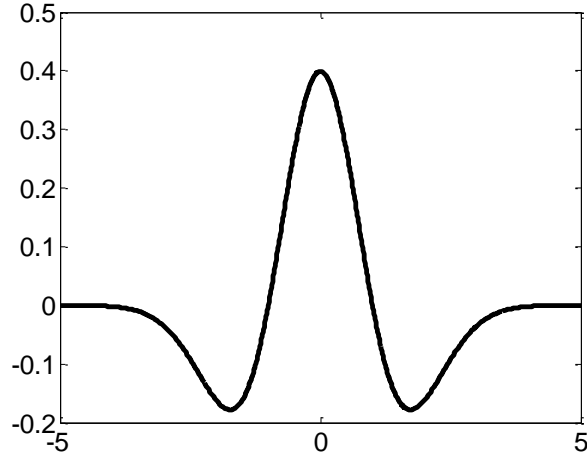


Figure 1.8: *Mexican Hat Wavelet*

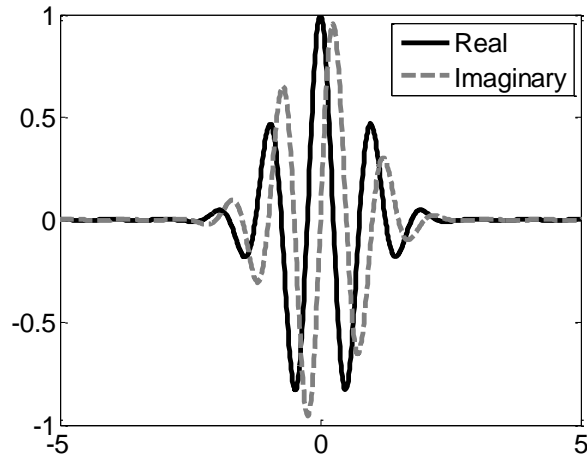


Figure 1.9: *Morlet Wavelet*

The mathematics defining the wavelet transform involves convolving a signal with the wavelet function for the scales of interest.

$$\int_{-\infty}^{\infty} u(t') * W(\sigma_s, t - t') dt' \quad (1.50)$$

where u is the signal being analyzed, W is the wavelet function and σ_s is the scale.

As stated earlier, wavelet analysis has been used in fluid dynamics by many researchers, especially to extract coherent structures in turbulence. One such example

is performed by Schobeiri *et al.* [80]. They use wavelet analysis to understand the flow structure in a boundary layer on a concave plate during transition . The wavelet analysis was chosen for this work due to the complex nature of the transitional flow. The wavelet analysis was able to pick out the different scales present. Meneveau [77] uses the wavelet to filter a turbulent flow field and showed that wavelet analysis is a good method for extracting intermittent features. Lewalle *et al.* [76] used wavelets to decompose a flow into coherent structures and a background field to show that they coexist. They state that the wavelet procedure extracts the coherent structure from the flow field in an robust manner. They also hypothesize that this may be used as a way of simulating a flow field using a coherent structure filter instead of an energy-based filter. A similar idea is discussed by Farge *et al.* [81]. Schneider *et al.* [82] use this idea to simulate a mixing layer flow. They showed that the resulting simulated data was better than that of the standard simulation methods because the small scale turbulence was extracted. This work showed wavelet filtering is a viable technique for simulating flow fields.

Discrete Wavelets

In addition to the continuous wavelets discussed previously, there are also discrete wavelets. Two common wavelets include the Haar wavelet [83] and the Daubechies series of wavelets [84]. As stated earlier, discrete wavelets can be used to fuse images as shown by the work of Yocky [18] Varsheny *et al.* [19] Ramac *et al.* [85] as discussed earlier. Image fusion was also performed by Nunez *et al.* [86] to show that an additive method for combining the wavelet components is able to correctly fuse high resolution black and white images with low resolution color images. Chen *et al.* [20] used image fusion to detect hidden weapons using different types of images of the same scene. A diagram outlining the image fusion procedure is shown again to reinforce this idea.

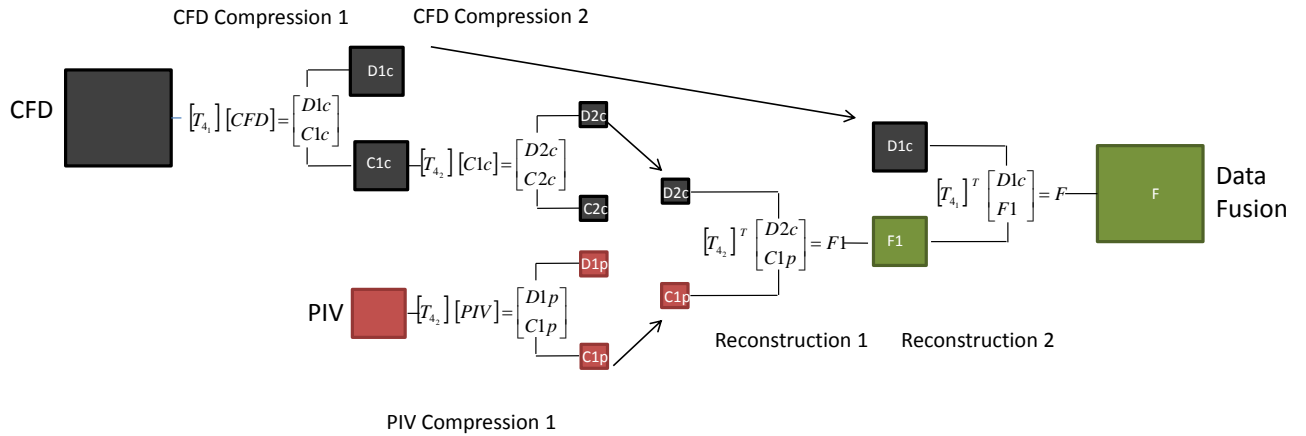


Figure 1.10: *Wavelet Image Fusion Process*

This is similar to figure 1.4, but shows how the wavelet filtering is done to compress the image by using the transformation matrix in equations 1.51. This method simply swaps the detail information, but there are more advanced methods of doing this as discussed earlier.

These wavelets have also been used for image compression by Lewis and Knowles [87] who used a 2-D wavelet to compress images and showed the images could be reconstructed with minimal losses. Calderbank *et al.* [88] created wavelets to perform lossless compression that also transformed an integer value to an value. Wavelets are also used for image de-noising as seen by the work of Chang *et al.* [89]. In this work they used lossy compression to remove noise from an image. Kivanc *et al.* [90] also used wavelets and statistical models to remove noise from images. Image stitching [91, 92] can also be accomplished using discrete wavelets.

The Daubechies wavelet transform is performed by multiplying the data (X) by an orthonormal matrix.

$$\begin{bmatrix}
c_4 & -c_3 & c_2 & -c_1 & & & & & & & \\
& & & c_4 & -c_3 & c_2 & -c_1 & & & & \\
& & & & & & \ddots & & & & \\
& & & & & & & c_4 & -c_3 & c_2 & -c_1 \\
c_2 & -c_1 & & & & & & & c_4 & -c_3 & \\
c_1 & c_2 & c_3 & c_4 & & & & & & & \\
& & & & c_1 & c_2 & c_3 & c_4 & & & \\
& & & & & & \ddots & & & & \\
& & & & & & & c_1 & c_2 & c_3 & c_4 \\
c_3 & c_4 & & & & & & & & c_1 & c_2
\end{bmatrix}
\begin{bmatrix}
X \\
X \\
X \\
X \\
X \\
X \\
X \\
X \\
X \\
X \\
X
\end{bmatrix}
=
\begin{bmatrix}
D_{ix} \\
\vdots \\
D_{ix} \\
C_{ix} \\
\vdots \\
C_{ix}
\end{bmatrix}
\tag{1.51}$$

The matrix in equation 1.51 is the transform for the Daubechies4 wavelet. Higher order Daubechies wavelets follow a similar pattern.

The bottom half of the transform matrix represents a low pass filter and acts as a smoothing operation and the top half acts as a high pass filter. These are known as a quadrature mirror filter [93]. The compressed data (C_{ix}) is given by the bottom half of the product of the transform matrix and the data (X). The top half of the product is removed and stored as detail information (D_{ix}), which is used to decompress the image.

The coefficients c_1, c_2, c_3, c_4 are determined by forcing the matrix to be normal,

$$c_1^2 + c_2^2 + c_3^2 + c_4^2 = 1 \tag{1.52}$$

forcing the columns to be orthogonal,

$$c_3c_1 + c_4c_2 = 0 \tag{1.53}$$

and forcing the first 2 moments to be zero.

$$\begin{aligned} c_4 - c_3 + c_2 - c_1 &= 0 \\ 0c_4 - 1c_3 + 2c_2 - 3c_1 &= 0 \end{aligned} \tag{1.54}$$

Solving the equations give the following coefficients: $c_1 = \frac{1+\sqrt{3}}{4\sqrt{2}} \approx 0.483$, $c_2 = \frac{3+\sqrt{3}}{4\sqrt{2}} \approx 0.837$, $c_3 = \frac{3-\sqrt{3}}{4\sqrt{2}} \approx 0.224$, $c_4 = \frac{1-\sqrt{3}}{4\sqrt{2}} \approx -0.129$.

The data set can be decompressed using an inverse transform. Since the matrix is orthonormal, the inverse of the matrix is also the transpose of the matrix. Before the data can be multiplied by the transpose matrix, the compressed image must be infused with detailed information.

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{1.55}$$

In equation 1.55, y_i is the compressed data and d_i is the detailed information.

The Haar wavelet is similar to the Daubechies wavelets; it actually follows the same constraints, but with two coefficients. The Haar wavelet ends up taking an average of adjacent points. The coefficients for the Haar wavelet are $c_1 = \frac{1}{\sqrt{2}}$ and $c_2 = \frac{1}{\sqrt{2}}$.

1.4.7 Application to Fluid Dynamics

Using the data from the sources discussed in [section 1.3](#) and the techniques explained in [section 1.4](#), researchers have already begun to fuse fluid dynamic data on some level. Stochastic estimation ([subsection 1.4.3](#)) may be thought of as a method of performing data fusion. As stated, the process essentially uses one piece of data to estimate another piece of data. For example, in standard PIV, one is most often unable to capture time dependent velocity fields as stated in [section 1.3](#). However, pressure sensors sample at a much higher rate, also as stated in [section 1.3](#). The stochastic estimation technique uses the pressure and PIV data in such a way that an estimate of the velocity data at a sampling rate equal to that of the pressure is produced. This is shown in work done by Pinier *et al.* [24] in which PIV and surface pressure for a NACA airfoil were used to estimate a time dependent velocity field. This method can be thought of as data fusion because two sources of data are being used together to produce information that is not obtainable from either measurement alone.

Another example of data fusion in fluid dynamics is presented by Planquart [94] in which the use of CFD and wind tunnel tests are used in conjunction to create a better design. Some examples presented by Planquart include the design of a sun protection system and a high-speed land vehicle. For the sun protection system, a prototype was placed in the wind tunnel and experimental values for the critical wind loading and deformation were recorded. That data was then used in a CFD simulation in which the geometry of the system was modified to create a better design. The system was then tested in the wind tunnel to confirm the CFD results [94].

Aerodynamic data fusion is also performed by Zimmermann *et al.* [95]. Zimmermann used a method similar to Gappy POD to fuse CFD and wind tunnel data. The CFD was used to generate basis functions and the wind tunnel was used to generate lift and drag measurements. The idea of this work was to generate the flow field around an airfoil given the lift and drag based on the CFD results. By generating the

flow field at many different flow conditions, Zimmermann could create a POD basis based on the CFD and the residual of the CFD. The data was fused by using the experimental lift, drag, etc. as a constraint. A similar approach was used by Vendl and Faßbender [96] to use Gappy POD to estimate the pressure on a surface of a wing. This was compared with the full CFD result and showed good comparison.

To accomplish the task of fusing fluid dynamic data, a framework to handle the vast amounts of data and analysis techniques is developed and explained in [chapter 2](#). Ultimately, data fusion is applied to data from a Mach 0.6 jet in [chapter 5](#), but the application of data fusion to a few other test cases are explored first to better understand how to apply data fusion to fluid dynamic data. The test cases include a synthetic data set ([chapter 3](#)), flow over a NACA 4412 airfoil ([chapter 4](#)), and the Mach 0.6 jet. The results of these test cases show data and technique fusion can improve fluid dynamic data.

Chapter 2

Orange High Speed Data (HSD)

Framework

As stated in [chapter 1](#), fluid dynamic research produces large amounts of data that come from many different sources such as PIV, LDA, Pressure, CFD, etc. Sometimes a large percentage of this is not used. Usually during an experiment, the researcher is focused on collecting as much data as possible in a given window of time. The researcher may not have the means to collect data again, making it important to get all of the data that is needed and more. Experimentalists try to collect enough data for the current project and possibly a few other projects [97]. In addition, the researcher may not have time to process the data while collecting it, other than an initial check. Errors could arise that corrupt the data, and the researcher most likely would not notice these until after the experiment is disassembled. These are two reasons why the vast amounts of data collected for fluid dynamic research are not used.

If all of the data collected is used, it may lead to a better understanding of the problem under investigation, making it advantageous to have a method that would increase the amount of data that can be processed. To attempt to create this method, it is important to understand why data is unused. It is fairly obvious as to why

corrupt data is unused, so having a method of repairing data accurately and in a timely manner is important and is discussed throughout this document. But why is much of the non-corrupt data not processed, especially if using it would be insightful? Each researcher usually writes his/her own codes to process all of the data, extract the useful data, perform the analyses, and track the data, which could potentially take a lot of time. If a standard method for data processing existed the researcher may spend more time analyzing the data and less time processing it, allowing for the use of more data.

2.1 ORANGE High Speed Data

One solution is to develop a framework capable of handling large amounts of fluid dynamic data, and capable of streamlining the repair and analysis process. Such a framework has been implemented in the tool ORANGE High Speed Data (HSD) developed using Matlab at Syracuse University by Ruscher [98] (figure 2.1). This allows one to use different analysis tools and compare them side-by-side allowing the user to draw conclusions from multiple analyses. ORANGE HSD includes POD (subsection 1.4.2), LSE (subsection 1.4.3), wavelets (subsection 1.4.6), spectral methods and other useful tools. ORANGE HSD allows the user to format data, perform analysis, and view results with the click of buttons and not much effort is required from the user. The user may have to spend some time reading the manual (Appendix B) to understand how to use the tool.

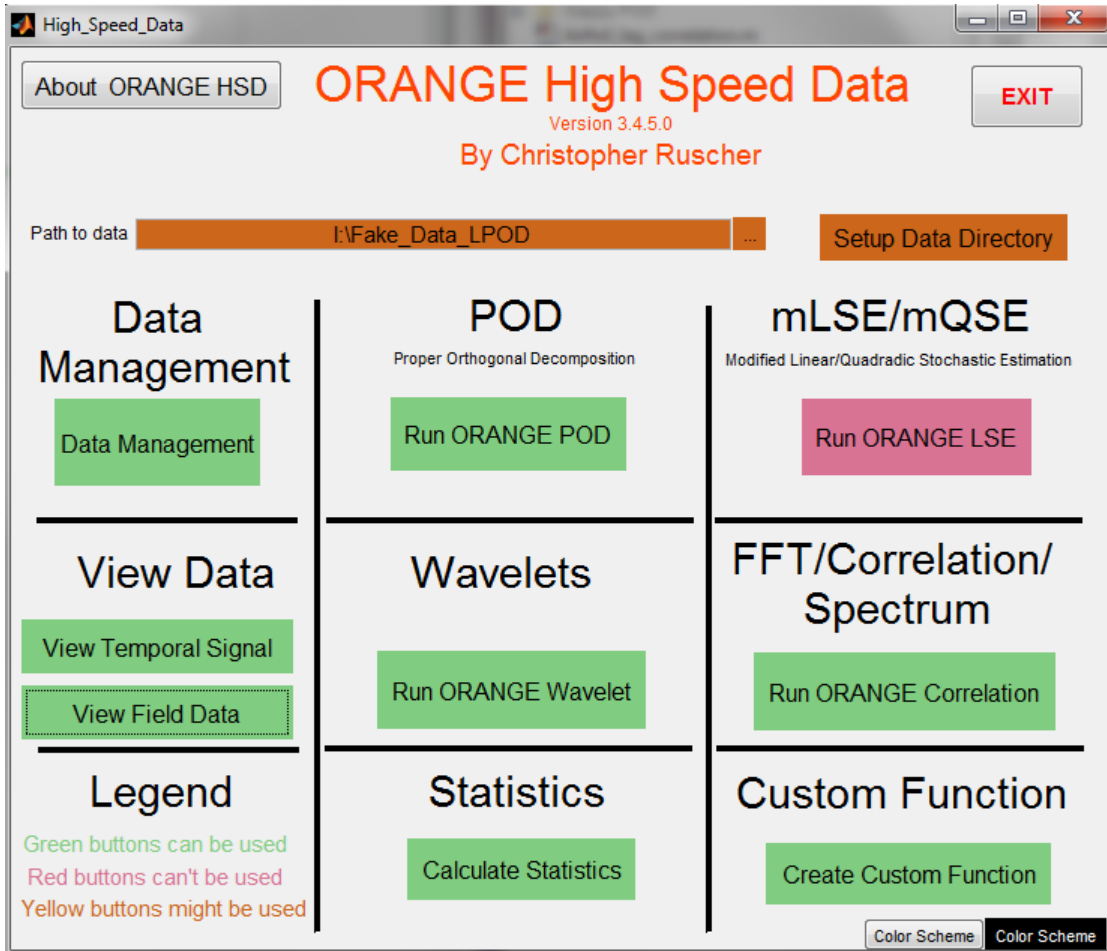


Figure 2.1: Screenshot of ORANGE High Speed Data Main Screen

Some of the different tools incorporated in ORANGE HSD are discussed in the following subsections. Examples of the output are displayed through out the subsections. The results come from data collected by Pinier *et al.* [24] for a NACA 4412 airfoil at an angle of attack of 16° .

2.1.1 Data Management

ORANGE HSD has a data management system (figure 2.2) that stores and displays relevant information about the data whenever a new data set is added or created instead of someone having to record what was done to the data. This allows the user

to spend less time keeping track of data manipulations and more time understanding the data. This process is automated and should have less chance for errors in the recording process. The framework tracks where the new data comes from so the user can always know where the data came from without having to do extensive searching through their notes.

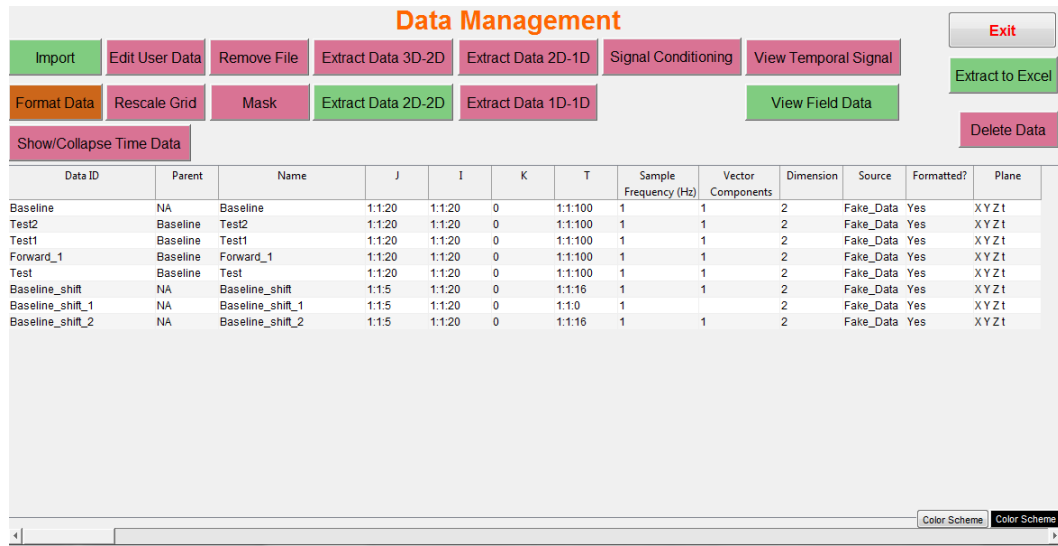


Figure 2.2: Screenshot of Data Management: The Data Management Tool Included in ORANGE HSD

ORANGE HSD also has a formatting tool, to read data from different sources and automatically put it into a standard format, allowing the user to spend time on other, more useful tasks, such as analyzing results. The standard format allows direct comparison of the data, which could require extra processing for the researcher without the standard format.

In addition, the data management tool allows the user to extract a new data set from an existing data set. This lets the user trim data or reduce the dimension of the data. This could be done without a framework, but having this ability coupled with the tracking system allows for an efficient way of extracting data and storing its origin. The extraction tool can extract single-point measurements from a plane of data, allowing for one-dimensional plots and possible comparisons to data collected

using a single-point measurement technique. For example, one could extract a single point from a PIV plane to compare it to an LDA measurement or extract a plane from a 3D CFD data set to compare with PIV data.

The data management tool also allows the user to mask out bad data or an object that may be in the image such as an airfoil, turret, or nozzle, which could corrupt some analyses. This is especially useful for POD and Gappy/Fused POD. The mask generation process is graphically based and only requires the user to click around the desired region, which could be faster and more efficient than finding the bad points and creating a mask manually. An example of the masking process is shown in figure 2.3.

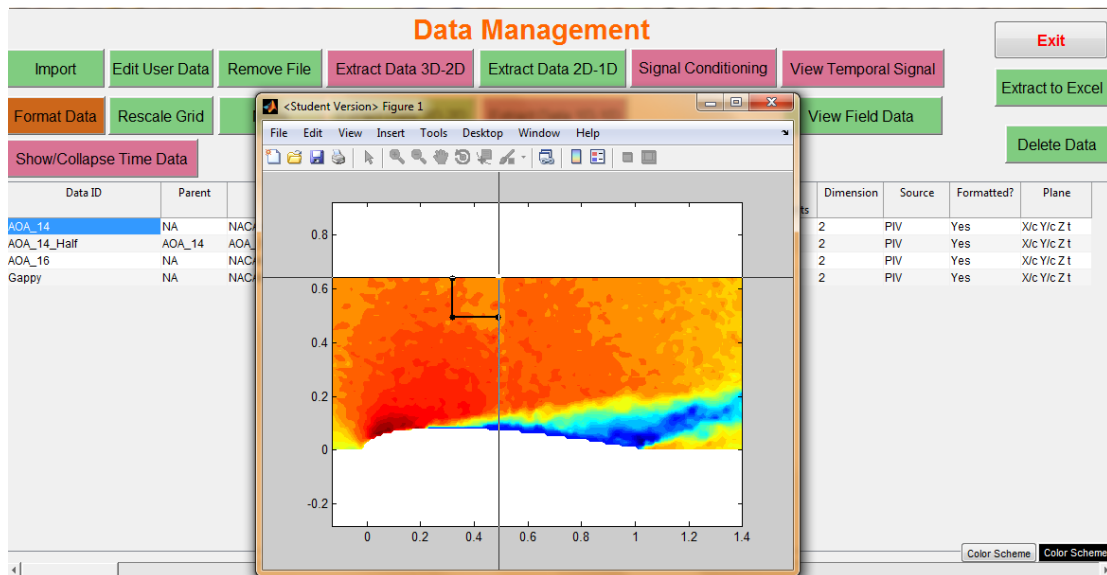


Figure 2.3: Generating a Mask for the NACA 4412 Data

2.1.2 ORANGE POD

The first of the four major analyses included in ORANGE HSD to discuss is ORANGE POD (figure 2.4), which is the POD tool for ORANGE HSD. The tool's main purpose is to compute the POD modes using the snapshot method (1.4.2). The user can then view the modes and temporal coefficients and compare them for the different data

sets. ORANGE HSD is capable of computing the POD modes for large data sets which is very attractive for TRPIV data sets and has been implemented by Low *et al.* [99], Berger *et al.* [100, 101], and Berry [102]. The tool also includes other variations of POD (subsection 1.4.2), such as iterative Gappy POD, Classical POD, Split POD, and Fused POD (subsection 4.6.1).

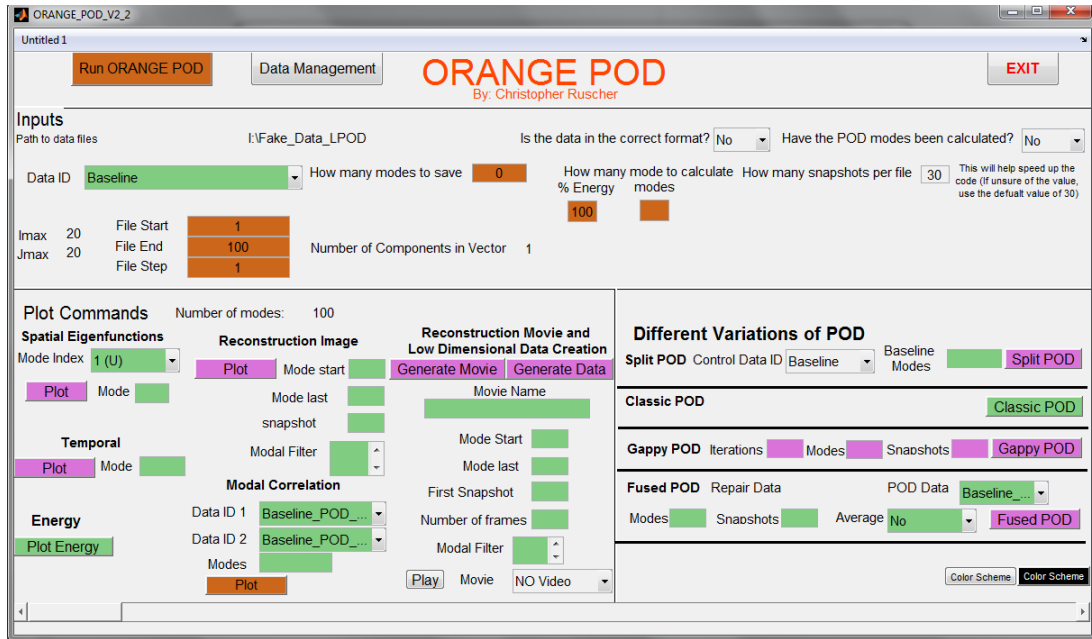


Figure 2.4: Screenshot of ORANGE POD: The Proper Orthogonal Decomposition Tool Included in ORANGE HSD

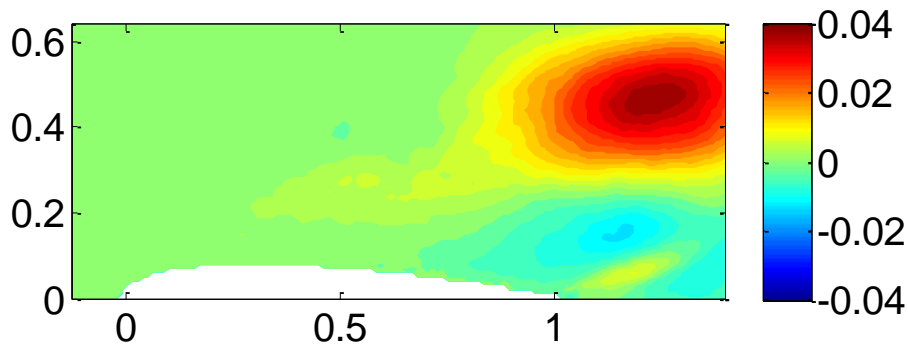


Figure 2.5: First POD Mode for the Streamwise Velocity Over the NACA 4412 Airfoil

The first POD mode for the NACA 4412 airfoil data is shown in figure 2.5. One can see the large structure in the wake which is the dominant feature in the flow field.

2.1.3 ORANGE Wavelets

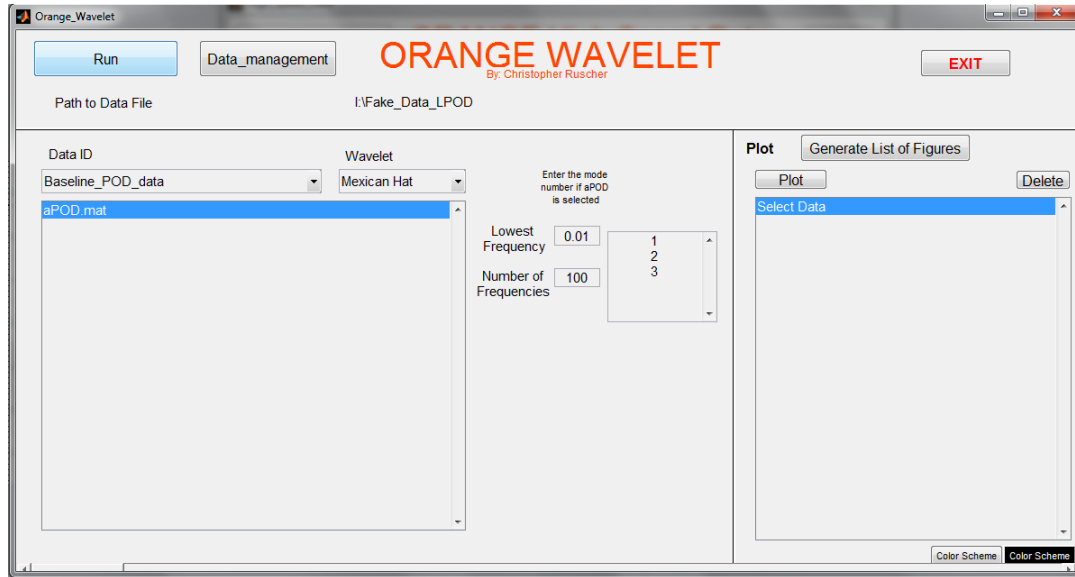


Figure 2.6: *Screenshot of ORANGE Wavelets: The Wavelet Tool Included in ORANGE HSD*

The next major tool included in ORANGE HSD is ORANGE Wavelets (figure 2.6), which is used to calculate the continuous wavelet transform for a signal. The tool is currently capable of using the Morlet transform (subsection 1.4.6) and the Mexican hat transform (subsection 1.4.6), which produce time-frequency relationships for signals. The code is designed in a such a way that other transforms can be added without much effort.

The wavelet tool has been used to generate the wavelet map for the pressure on the surface of the NACA 4412 airfoil. The results of this are shown in figure 2.7. One can see a time frequency representation of the data.

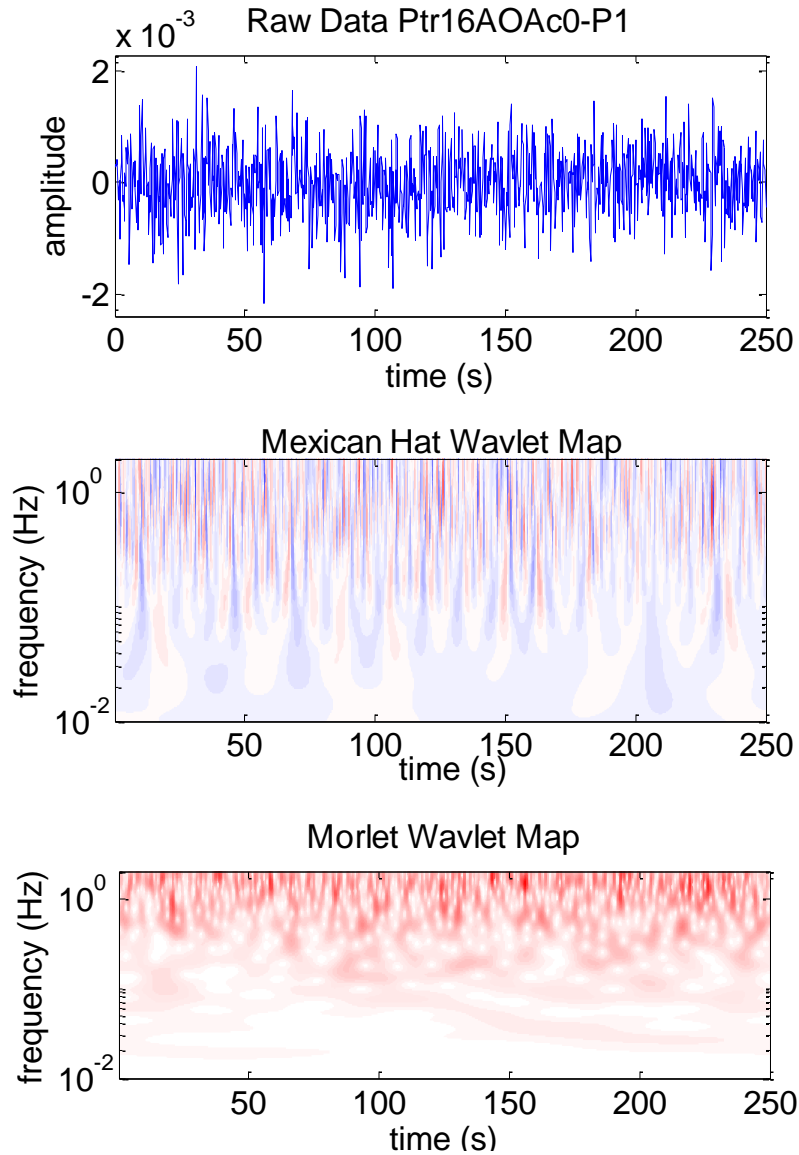


Figure 2.7: *Mexican Hat and Morlet Wavelet Map for NACA 4412 Surface Pressure*

2.1.4 ORANGE Correlate

The third major tool is ORANGE Correlate (figure 2.8), which performs auto- and cross-correlations, auto- and cross-spectrum, and Fourier transforms. The user again can compare the results of multiple data sets and determine how the spectral content and time lags in a signal compare with one another. The tool also allows the user to obtain smooth results by performing block averaging and using a moving filter.

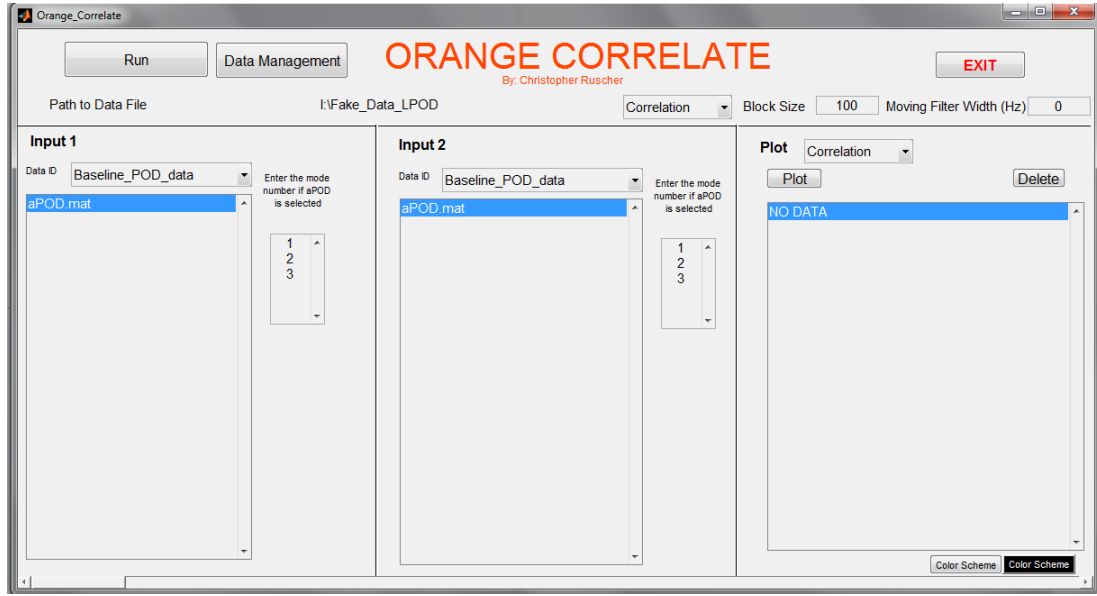


Figure 2.8: Screenshot of ORANGE Correlate: The Spectral Methods Tool Included in ORANGE HSD

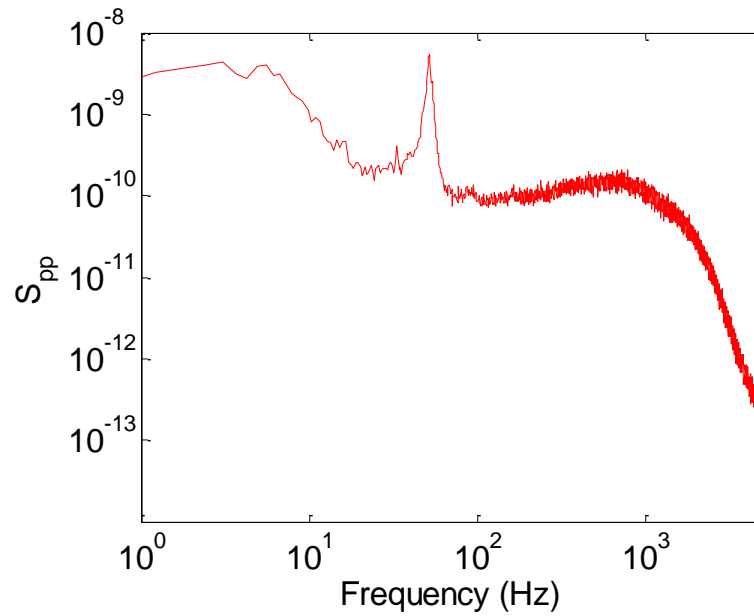


Figure 2.9: Autospectrum for NACA 4412 Surface Pressure

An autospectrum for the surface pressure data for the NACA 4412 airfoil was computed using ORANGE Correlate. One can see the dominant frequency in this

plot as well as the energy cascade. This is shown in figure 2.9.

2.1.5 ORANGE LSE

The last major tool included with ORANGE HSD is ORANGE LSE (figure 2.10), which performs LSE (subsection 1.4.3) on a data set. One can use a signal to generate a mapping from one signal to another that can be used as an estimation technique when one signal is known and the other unknown. This can be useful to estimate time-resolved velocity using high-speed pressure measurements and non-time-resolved PIV.

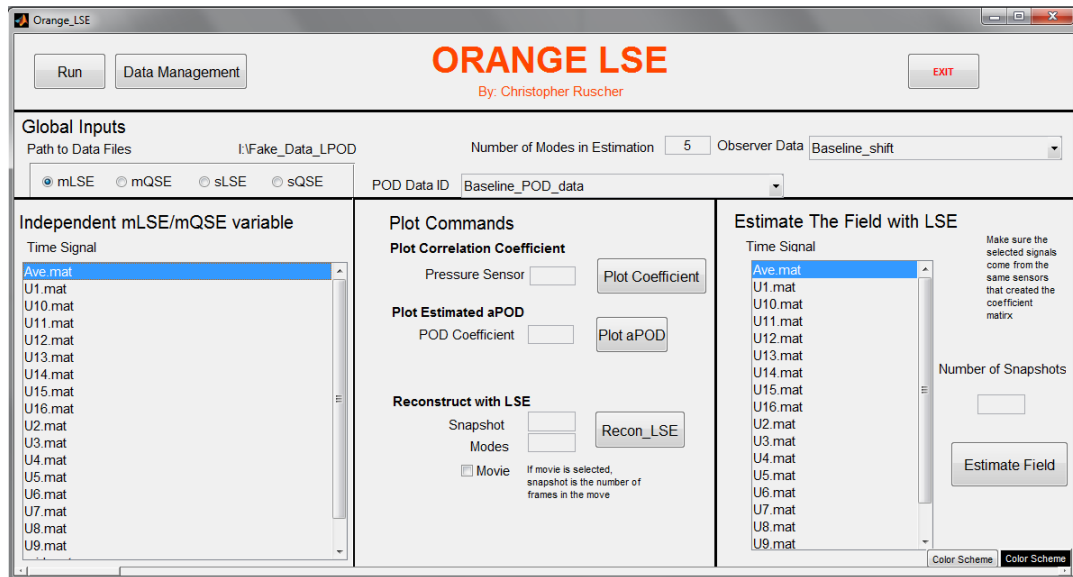


Figure 2.10: *Screenshot of ORANGE LSE: The Stochastic Estimation Tool Included in ORANGE HSD*

The surface pressure can be used to estimate the velocity field using ORANGE LSE. The results of this are shown in figure 2.11. One can see an original snapshot of the fluctuating velocity field on the top, the POD reconstruction of the same velocity field, and the LSE of the same field. It can be seen that that three images are similar and the LSE is estimating the large scale structures correctly.

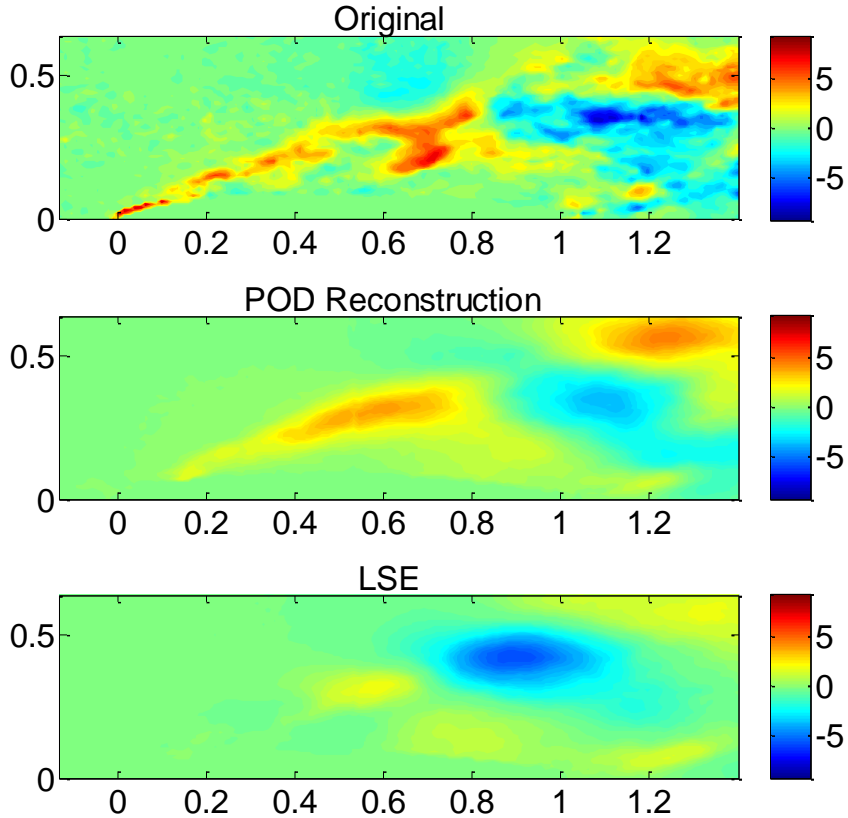


Figure 2.11: *Comparison of the Actual, POD Reconstruction, and LSE Velocity Fields*

ORANGE HSD allows the user to perform these analyses at a push of a few buttons, saving time because the user no longer has to search for codes they wrote and modify them for new data. One should note that even though the code makes it easy to run these analyses, the user still needs to understand the techniques and results. The user of the tool should have an understanding of the techniques before using them. The code is not intended to be a substitute for learning the techniques, but rather as an implementation of a framework to streamline a researcher’s analysis process, allowing the researcher to analyze more data. A more in depth description of ORANGE HSD is included in appendix B which is the user’s manual for ORANGE HSD.

As a final note, the main advantage of the framework is the automation. Some

of the analyses may take about the same time to run, but the user does not have to spend all of that time watching the code. In other words, the code is reducing the man hours required to perform the analyses allowing the researcher to spend more time understanding the data and less time processing.

2.2 Examples

The framework has been used in a few different cases including the comparison of LDV and TRPIV measurements in a high-speed jet taken in 2004 by Hall *et al.* [103] and in 2011 by Low *et al.* [99] respectively. These two experiments measured the same quantity (velocity) using two different techniques. Despite this, the two measurements were not compared since the researchers were focused on processing their own data and did not have the time to make the comparison. However, using this framework, the preprocessing required to compare the two data sets was accomplished in a day. The preprocessing included shifting the PIV grid to match the LDV data, extracting single point data from the PIV (figure 2.12 shows the mean Mach number contours and the black points are where the data was extracted), computing the auto-spectrum, and comparing the auto-spectrum on a single plot. Ruscher *et al.* [104] could focus on the analysis of the data sets instead of the processing the data. Since the framework was in place, the research shifted to understanding the differences in the data and the errors associated with the measurement techniques. Without the framework it is unlikely this work would have been done.

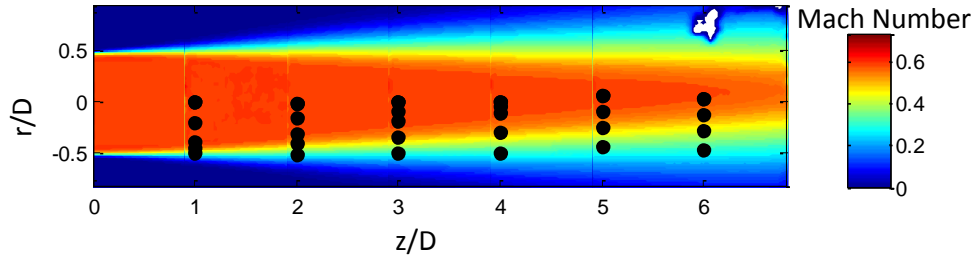


Figure 2.12: *Extraction of Data from the TRPIV for Comparison with LDV Data*

The framework also helps perform data fusion since many of the data fusion processes discussed in this document require data to be manipulated in such a way that the creation of a new data set make the processes more efficient. In the case of using Fused POD to estimate missing data where a sensor was located, ([section 5.4](#)) the data was rotated, split into two parts (the top and bottom halves) and the POD modes were calculated for the bottom half. Then Fused POD was performed on the top half of the data using the bottom half modes. The two halves were then joined together, creating a new data set, the repaired data. This required many manipulations and by using the framework this was accomplished with little effort and the newly created data files were stored by the framework for future analysis. Fused POD could have been done without the framework, but it made the task easier, allowing more time to focus on the analysis of the new data and the technique. This allowed for an increased understanding of the fusion tool by looking at the effects of hole size, where to split the data, and convergence of the POD modes. All of this is discussed in [section 5.4](#).

The framework also aided in the repair of TRPIV jet data that was corrupted by random dropouts of seeding ([subsection 5.4.2](#)). This can be repaired using the iterative Gappy POD, but Fused POD is able to repair the data faster albeit with slightly different results as shown in [subsection 5.4.2](#). To use Fused POD, a subset of the data was extracted and iterative Gappy POD was performed on the smaller data set, which was used to create a POD basis used to repair the entire data set

two order of magnitudes faster than using iterative Gappy POD alone. Again, the framework was helpful in extracting the necessary data and tracking the new data sets that were created. All of this could have been done without the framework, but would have taken more time and man hours.

The framework, also was used to estimate a time-resolved large-window data set using dynamic Gappy POD (subsection 5.5.4). In the Dynamic Gappy POD the LWPIV was shifted to match that of the (TRPIV). Then a smaller window, the desired size of the large-window time-resolved data, was extracted from the LWPIV. POD was then performed on the TRPIV windows, which are used to generate a model, and the LWPIV which gives the mode shapes for the new data set. With the modes, the model, and the TRPIV data, a large-window time-resolved data set is generated. The framework again was used to create and track new data sets that were required for DGP, streamlining the fusion process.

The framework provided by ORANGE HSD allows researchers to focus more on the actual analysis and understanding of the data instead of focusing on processing the data. This allows researchers to use more of the data that was collected as shown by the number of data sets processed by Berger [1], who used this framework, as opposed to Low [105], who did not. Low was able to process about 13 TRPIV data sets for a Mach 0.6 jet, which included 7 baseline data sets that range from the lip of the jet to 7.5D downstream and six open loop control cases for the same range. During the data collection phase, 7 data sets were taken for the jet at Mach 0.85 and Mach 1.0, which were not processed until a later time by Berger *et al.*[101, 106]. In addition to processing the data collected in 2011 by Low *et al.* [99], Berger also was able to collect and process data from another set of TRPIV experiments [100] as well as a set of LWPIV experiments [107]. Approximately 80 different data sets were acquired for the 2013 TRPIV experiments and five data sets were acquired for the LWPIV experiments. These included baseline, open- and closed-loop control, off-center planes, and asymmetric nozzles. According to private communication with Berger,

approximately 2/3 of this data have been processed and analyzed [97]. The processing includes formatting the data, computing the POD modes, data extraction, spectral tools, vorticity calculation, assessment of data quality, data repair/estimation, and more. The framework implemented in ORANGE HSD allowed Berger to process more data and make comparisons between the data, which would not have been possible in the given time frame of a typical Ph.D. thesis without this framework. Because Berger was required to spend less time processing data, more time was spent looking at results and understanding the data, leading to a deeper understanding of the jet flow.

An example of the processing done by Berger [1] is shown in figure 2.13. All of the mods were generated using ORANGE POD. These correspond to 10 different off center data sets that each had 15,000 snapshots. Using ORANGE POD Berger was able to process the data efficiently. As a note, figure 2.13 was not generated using ORANGE POD. It was generated using a code written by Berger. However, the calculation of all of the modes were done using ORANGE POD.

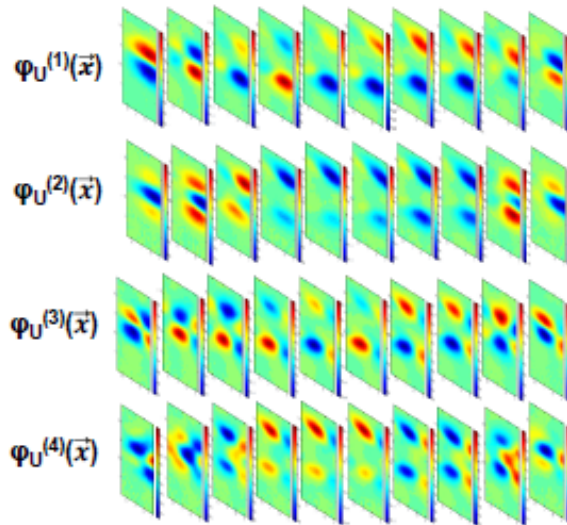


Figure 2.13: *First Four Spatial Modes in the Radial Direction (Figure 6.54 in Berger [1])*

In addition, some of the TRPIV data collected by Berger *et al.* [100] was corrupted due to particle drop out and/or laser misalignment, which caused non-physical velocities in all of the snapshots (subsection 5.4.2). The framework and the idea of Gappy POD/Fused POD was used by Berger to repair these snapshots in a reasonable time. If not for the framework and the fusion techniques, those few data sets most likely would not have been used.

Some of the 2011 TRPIV [99] data was processed by Berry [102] to see the effects of the flow control on the TRPIV experiments using split POD (subsection 1.4.2). Using the framework, Berry was able to perform the split POD in an efficient manner for multiple control cases, due to the frameworks ability to track and handle multiple data sets.

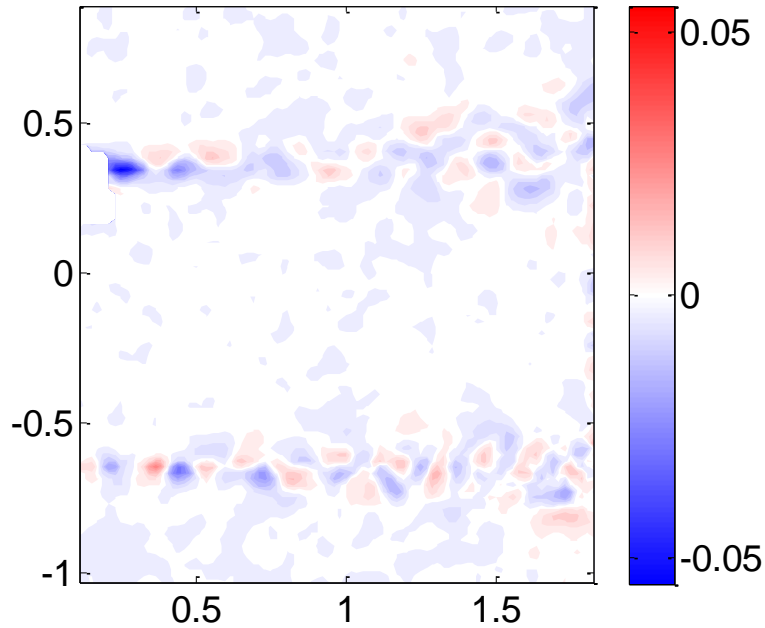


Figure 2.14: *Mode One for the Split POD Between Baseline and Control for a Mach 0.6 Jet*

A sample of this work is shown in figure 2.14. This shows the first split POD mode for the difference between the baseline and control case for a Mach 0.6 jet. One can see structures forming at $r/D = 0.5$ and -0.5 . This is the effect of the control

system which is a synthetic jet actuator. These results could be used to develop a more robust controller in the future.

In addition to the split POD, Berry [102] also processed the asymmetric nozzle data sets from the 2013 TRPIV experiments. The data sets had many “bad” data points on the edges of the plane due to the nozzle being in the images and the laser plane not illuminating the entire window. The framework allowed the images to be cut and then processed using POD. This was done for multiple configurations of asymmetric jet nozzles. The framework allowed Berry to process the data efficiently and focus his attention on understanding the results. This work provided insight for future asymmetric nozzle work that will be done at Syracuse University’s Skytop Anechoic facility.

Chapter 3

Synthetic Data

The techniques described in [section 1.4](#) are ultimately applied to jet data in [chapter 5](#), but to test the validity of the techniques and determine how they work for a given scenario they are first applied to synthetic data. The test cases described in [subsection 3.1.1](#) are designed to mimic problems that are faced in real data. Various techniques are applied to the synthetic data to determine which one works best for a given scenario. The findings from this work ultimately guide the creation of the fused techniques discussed and applied in [chapter 4](#) , and [chapter 5](#).

3.1 One-Dimensional Test Case

The synthetic test cases try to mimic the strengths and weaknesses of computational and experimental data, which are two different sources of data that could be used in the fusion process. The test signal is a combination of sinusoids with varying frequency and amplitudes and is given by

$$\sin(t-\tau)+0.7\sin(2(t-\tau))+0.5\sin(4(t-\tau))+0.3\sin(7(t-\tau))+0.2\sin(9(t-\tau)) \quad (3.1)$$

and represents non-corrupt data. This signal is modified to incorporate errors that can occur in PIV and CFD data. The different PIV errors include: downsampled data

(represents data with less resolution than the other), noisy and downsampled data (represents uncertainty in measurements), data that is missing (represents dropout, reflections, or sensor occlusions), and a data set that is downsampled, noisy, has missing data, and has multiple windows (represents two PIV windows adjacent to one another but not synchronized in time).

The original signal is modified to represent CFD by adding dispersion, which is a numerical error that occurs from the discretization of the flow field. The dispersion is added by running the original signal through the advection equation

$$\frac{\partial u}{\partial t} + U \frac{\partial u}{\partial x} = 0 \quad (3.2)$$

using a forward-time backward-space scheme

$$u_i^{n+1} = CFL(u_{i-1}^n - u_i^n) + u_i^n \quad (3.3)$$

with a Courant Friedrichs Lewy ($CFL = \frac{U\Delta t}{\Delta x}$) number of 0.5.

The PIV-inspired signal, CFD-inspired signal, and fused signal are compared to the original signal by the root mean square (RMS) error

$$\sqrt{\frac{\sum_{i=1}^{N_s} (x_i - y_i)^2}{N_s}} \quad (3.4)$$

where x_i represents the actual signal, y_i represents the signal being compared with the original, and N_s represents the number of data points in the signal. The RMS error is used as a metric to determine how well a signal matches the original.

3.1.1 Test Case 1: Downsampled/Dispersed Signal

The first test case uses a downsampled signal and a dispersed signal. The downsampled signal represents the fact that a PIV signal is limited in the number of pixels that can be captured. In this test, the actual signal is downsampled by a factor of

four to produce the PIV-inspired signal and separately run through the advection equation (equation 3.3) to obtain the CFD-inspired signal. The signals for this test case are shown in figure 3.1.

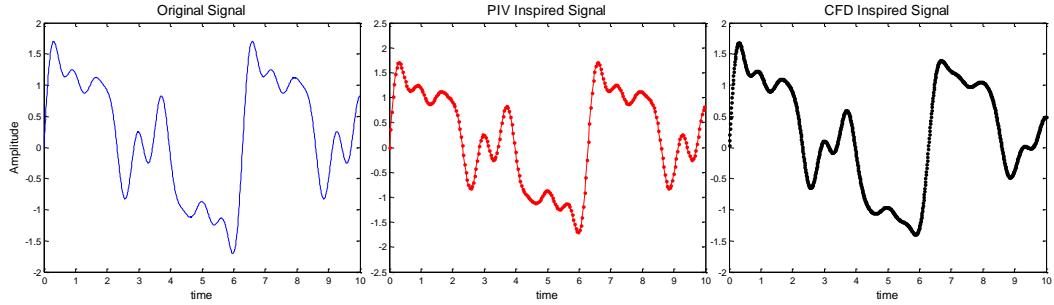


Figure 3.1: *Test Case 1: Downsampled Signal and Dispersed Signal*

Note the CFD-inspired signal becomes smoother than the original signal as time increases due to the dispersion.

3.1.2 Test Case 2: Noisy Downsampled/Dispersed Signal

The second test case uses a downsampled and noisy signal to represent PIV data and a dispersed signal to represent CFD data. This test case adds the complication of noise that may be experienced when taking experimental measurements. For this test case, the noise is 10% of the signal with a Gaussian distribution centered around zero. This is a conservative estimate of the uncertainty in a PIV measurement of a jet flow based on the method outlined by Raffel *et al.* [28] and the calculation performed by Ruscher *et al.* [104]. The signals for this case are in figure 3.2.

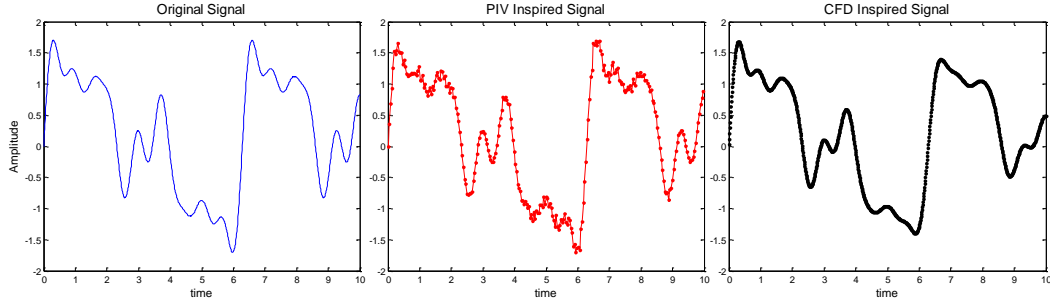


Figure 3.2: *Test Case 2: Downsampled and Noisy Signal and Dispersed Signal*

3.1.3 Test Case 3: Gappy/Dispersed Signal

The third test case has gaps or missing data to represent PIV data that has dropouts, reflections, or sensor occlusions that corrupted the data and a dispersed signal to represent CFD data, as shown in figure 3.3.

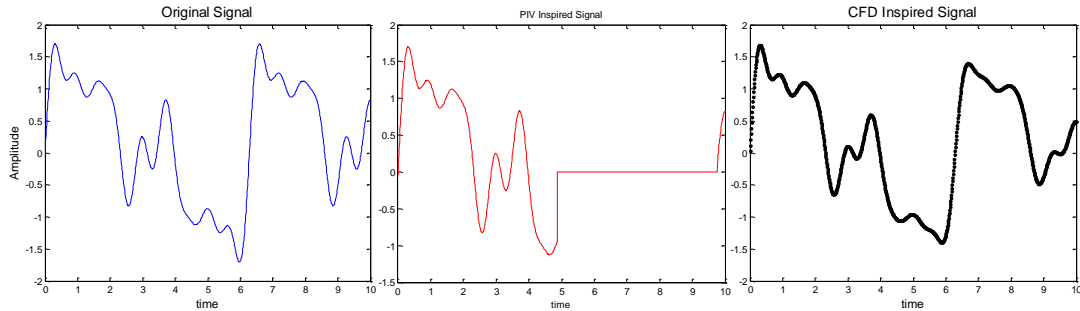


Figure 3.3: *Test Case 3: Gappy Signal and Dispersed Signal*

3.1.4 Test Case 4: Gappy, Multiple Windows, Noisy, Downsampled/ Dispersed Signal

The final test case investigated is a signal that has all of the errors discussed in the previous test cases and it has multiple windows that are not aligned to represent PIV data. A dispersed signal is used to represent CFD data. These signals are displayed in figure 3.4. Notice in figure 3.4, the PIV data has a dashed line which represent the boundary between the two pseudo-windows.

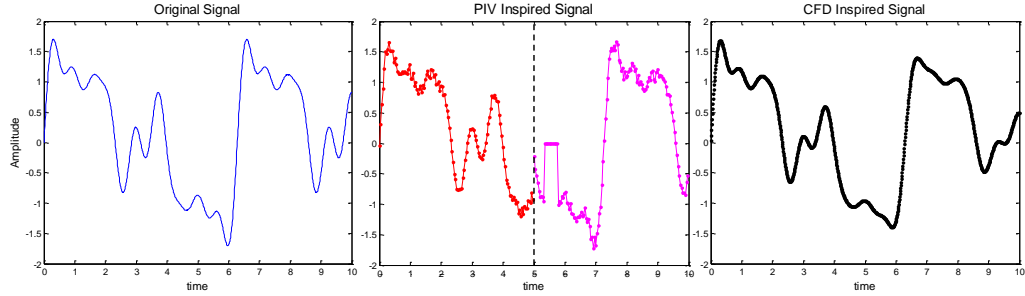


Figure 3.4: *Test Case 4: Gappy, Multiple Window, Noisy, Downsampled Signal and Dispersed Signal*

3.2 Discrete Wavelet Fusion

The first technique discussed is the discrete wavelet transform. The process starts by using the discrete wavelet transform to compress the CFD-inspired signal to match the same resolution as the PIV-inspired signal. Then the CFD-inspired signal and PIV-inspired signal can be compressed further if desired (secondary compression, described in [subsection 3.2.2](#)). The PIV-inspired signal is reconstructed using the detailed information from the CFD-inspired signal. This produces a signal that resembles the actual signal. The data in the test cases discussed in [section 3.1](#) are now fused together using the wavelet fusion outlined in the block diagram in [figure 1.10](#).

3.2.1 Test Case 1

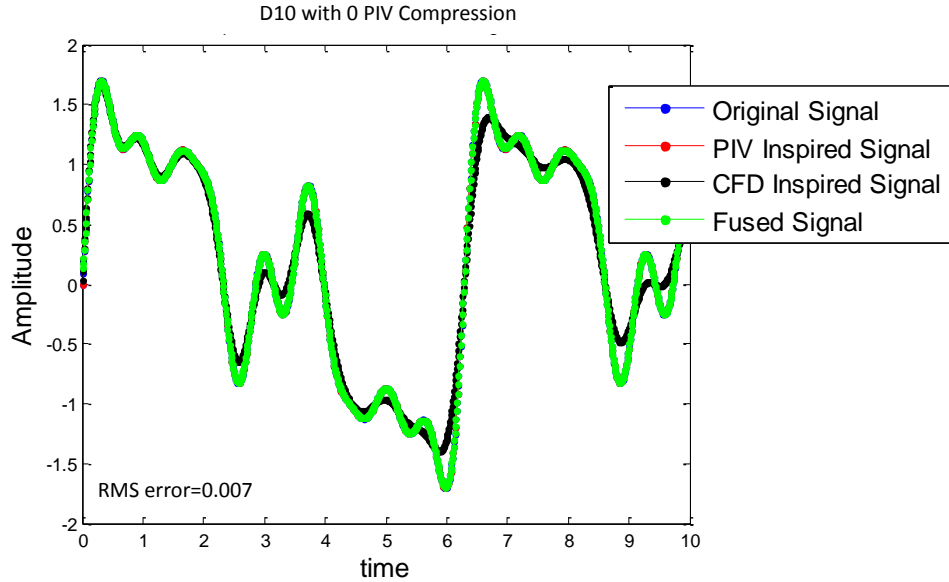


Figure 3.5: *Wavelet Fusion of a Downsampled Signal and Dispersed Signal without Secondary Compressions*

Using the Daubechies10 discrete wavelet transform, the CFD-inspired signal and PIV-inspired signal from test case one are fused together. Using zero secondary compressions, the fused signal looks similar to the original signal (figure 3.5).

Performing one secondary compression increases the RMS error slightly, but the signals still look the same (figure 3.6). Performing five secondary compressions on the data increases the RMS error, such that it is nearly the same as the RMS error of the CFD-inspired signal. In addition, the fused signal now closely resembles the CFD-inspired signal (figure 3.7). This is expected because the CFD-inspired signal and PIV-inspired signal are similar. The further one compresses the signals the more similar they will become. Using a compressed signal that is similar to the CFD-inspired signal and the detailed information from the CFD-inspired signal will create a fused signal that looks like the CFD-inspired signal.

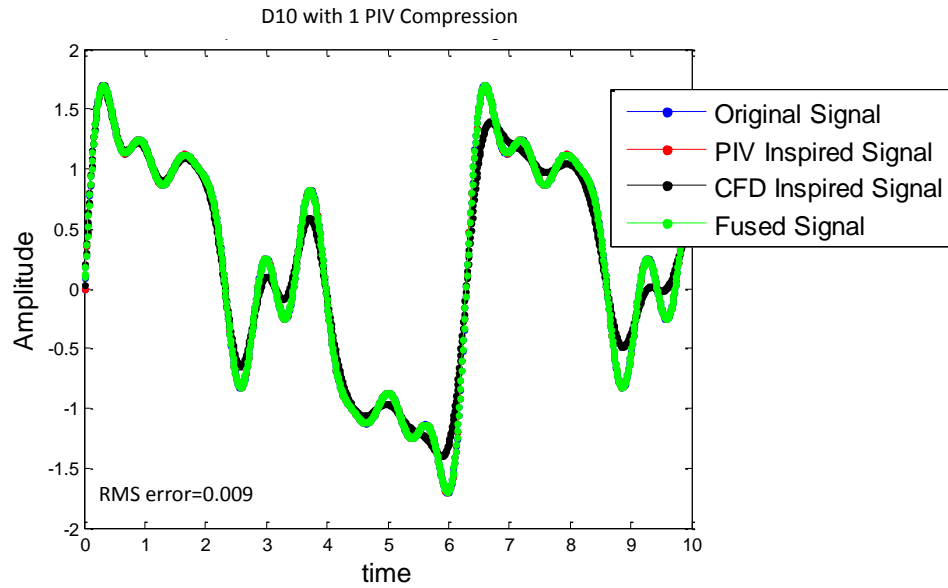


Figure 3.6: *Wavelet Fusion of a Downsampled Signal and Dispersed Signal with One Secondary Compression*

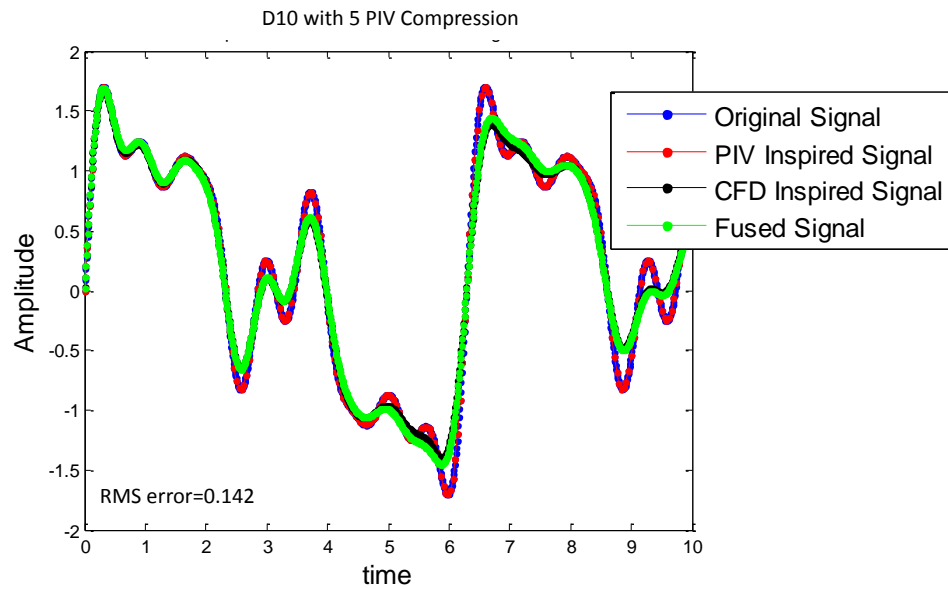


Figure 3.7: *Wavelet Fusion of a Downsampled Signal and Dispersed Signal with Five Secondary Compressions*

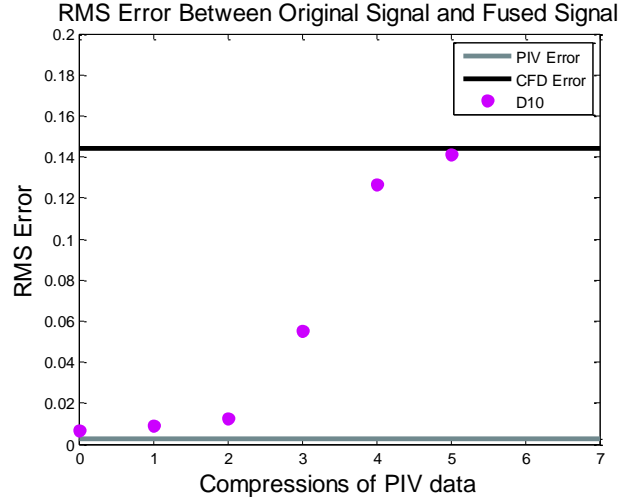


Figure 3.8: *Effect of Secondary Compressions on a Downsampled Signal and Dispersed Signal*

The RMS error increases as the number of secondary compressions increases, as is shown in figure 3.8. With no secondary compressions, the RMS error is equal to the RMS error for a linear interpolation of the downsampled data (the wavelet fusion is much smoother though). As the number of secondary compressions increase, the RMS error increases and converges to the RMS error of the CFD-inspired signal.

3.2.2 Test Case 2

The wavelet fusion technique is also used on the noisy downsampled PIV-inspired signal. The CFD-inspired signal and PIV-inspired signal are fused in the same manner as for test case 1. For test case 2 with no secondary compressions, the fused signal has some noise but still is close to the original signal (figure 3.9). Performing one secondary compression removes the noise from the signal and the RMS error decreases (figure 3.10). Five secondary compressions cause an increase in the RMS error (figure 3.11) for the same reasons as test case 1.

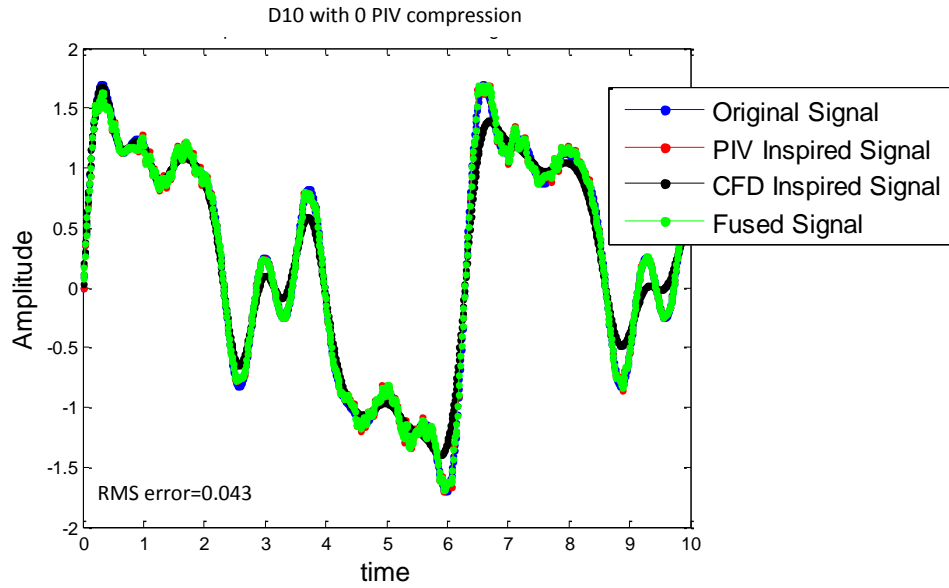


Figure 3.9: *Wavelet Fusion of a Noisy Downsampled Signal and Dispersed Signal with no Secondary Compressions*

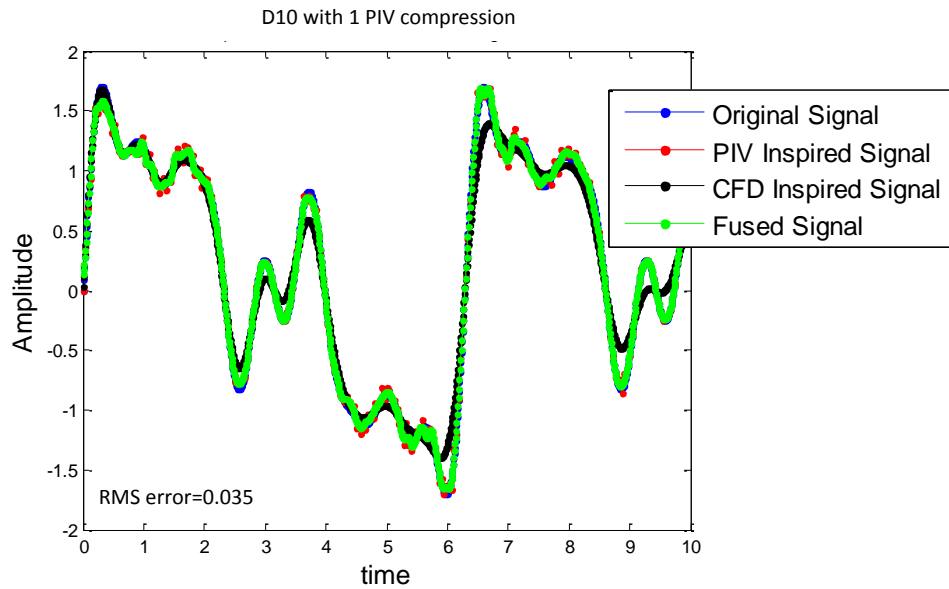


Figure 3.10: *Wavelet Fusion of a Noisy Downsampled Signal and Dispersed Signal with One Secondary Compression*

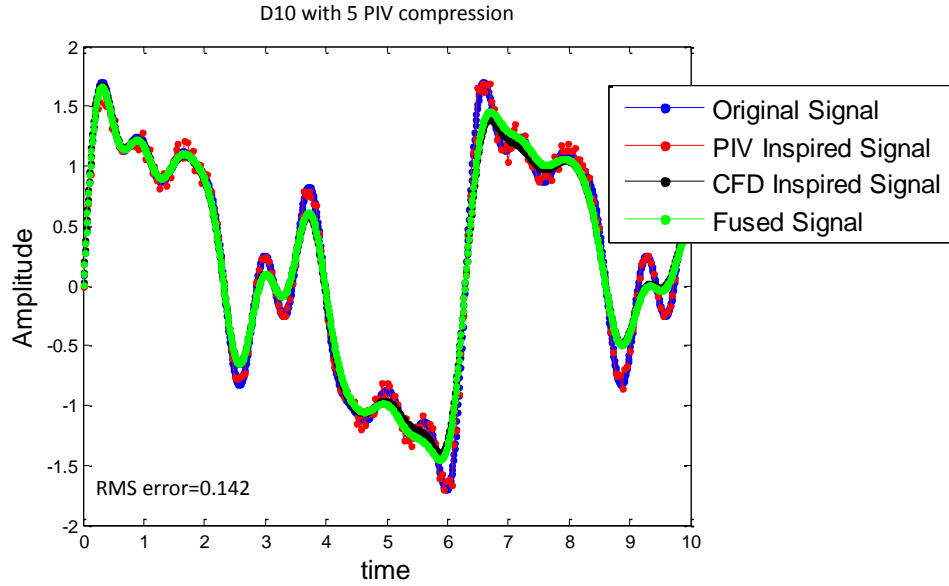


Figure 3.11: *Wavelet Fusion of a Noisy Downsampled Signal and Dispersed Signal with Five Secondary Compressions*

Performing secondary compressions on noisy data decrease the RMS error of the fused signal by filtering the noise. Performing too many secondary compressions will cause the RMS error to increase until the fused signal and the CFD-inspired signal are the same. This is because performing more compressions filters out information that is not noise. The fused signal created using one secondary compression is better than both the CFD- and PIV-inspired signals as shown in figure 3.12.

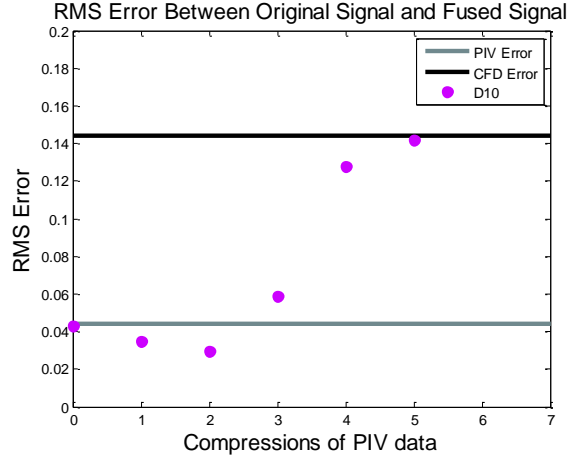


Figure 3.12: *Effect of Secondary Compressions of Wavelet Fusion for a Noisy Downsampled Signal and Dispersed Signal*

3.2.3 Test Case 3

The wavelet fusion technique is applied to the data set with missing data, which to the author’s knowledge is the first time this has been done. The two signals are fused in a similar manner as before. However, if care is not taken to treat the gappy region correctly, the missing information from the PIV-inspired signal will transfer over to the fused signal (figure 3.13). The fusion algorithm can be improved by only using CFD-inspired data in the gappy regions. This is equivalent to inserting the CFD-inspired data into the PIV-inspired data in the gappy regions (figure 3.14). This would help improve the PIV-inspired signal by having data in the field, but it is not taking advantage of fusion techniques.

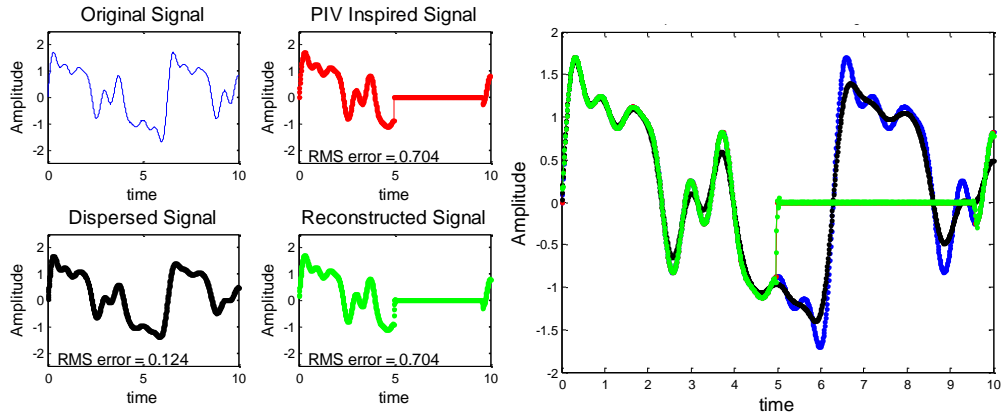


Figure 3.13: *Wavelet Fusion Applied to Gappy Data*

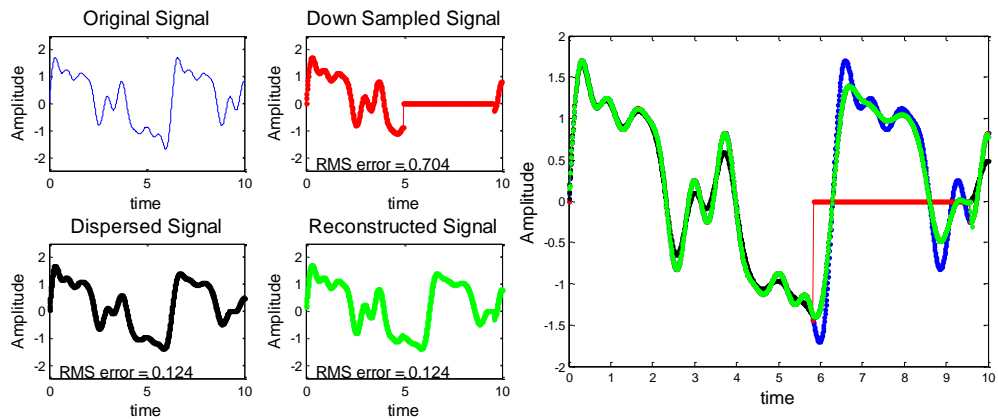


Figure 3.14: *Wavelet Fusion Applied to Gappy Data by Inserting Another Signal in the Gappy Region*

3.2.4 Comparison of Different Daubechies Wavelets

The Daubechies10 wavelet has been used for the fusion so far, but what about other wavelets? The differences in using various wavelets were studied for the noisy down-sampled case to determine the best wavelet to fuses the signals. This was performed for the Haar, Daubechies4, Daubechies6, Daubechies8, and Daubechies10 wavelets. It was found that as the complexity of the wavelet increased, the RMS error between the fused signal and original signal decreased. However, the RMS error reaches an asymptote, and using higher order wavelets does not improve the error as much (fig-

ure 3.15). This could be due to the fact that the higher order Daubechies wavelets use more of the neighboring points in the filtering process. The data points further away have less influence on the filtering process. If the signals have low frequency noise, perhaps higher order wavelets may be useful to smooth out the data.

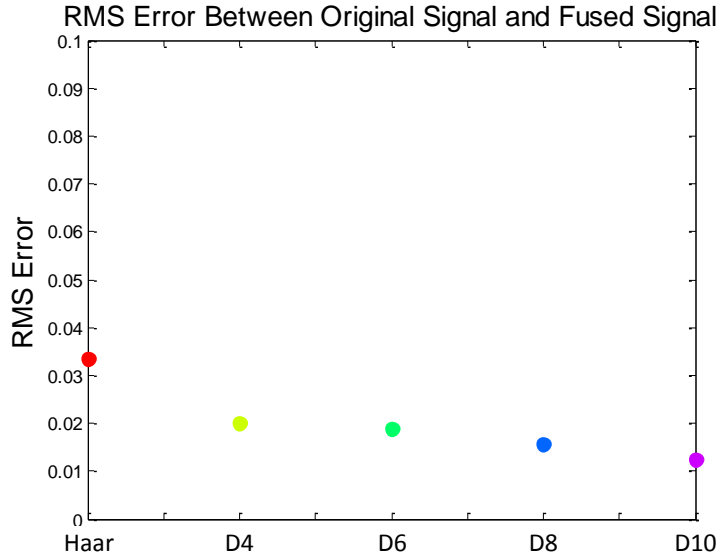


Figure 3.15: *RMS Error Using Different Wavelets for Fusion of a Noisy Downsampled Signal and Dispersed Signal*

3.3 Gappy POD Fusion

Wavelet fusion has been discussed in the previous section. The Gappy POD fusion technique is now explored in this section. The fusion process works by creating a POD basis for the CFD-inspired signal. The fused signal is then generated using a linear combination of the basis functions using the Gappy POD method outlined earlier (figure 1.6). This work was the initial inspiration for Fused POD.

3.3.1 Test Case 1

The test cases are fused with Gappy POD. The fusion requires that each image have the same resolution, which is not the case with the downsampled signal. However, if one assumes that the points missing from the downsampled signal are the gappy points, then Gappy POD can be used for the fusion process. The fused signal that is created is worse than the signal produced from the wavelet fusion. It has many discontinuities, which causes a large RMS error (figure 3.16).

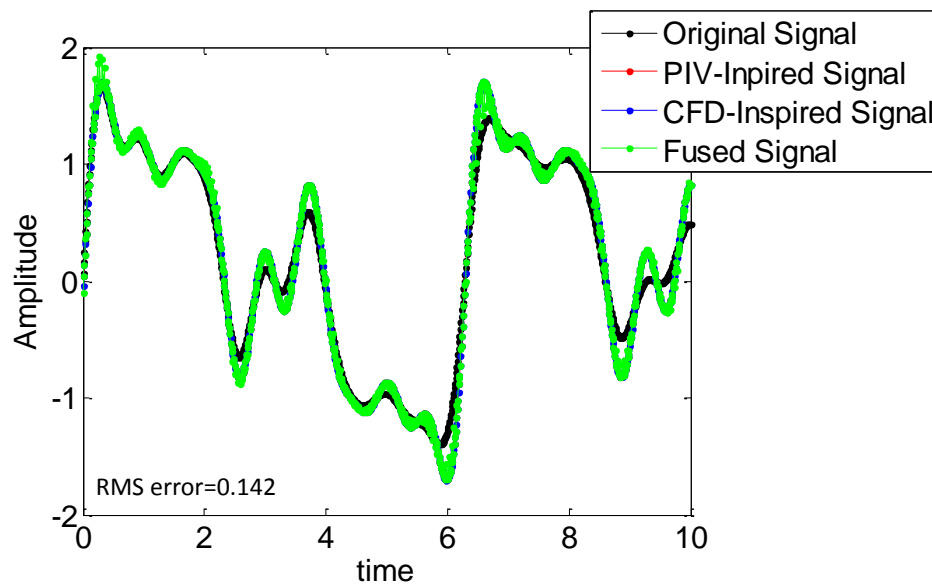


Figure 3.16: *Gappy POD Fusion Applied to a Downsampled Signal*

3.3.2 Test Case 2

The fusion of test case 2 has the same issue as test case 1 regarding the resolution of the signals; and the issue is solved using the same idea as in test case 1, where the missing resolution is treated as a gap. The Gappy POD fusion process again shows the fused signal has many discontinuities and the wavelet fusion technique is a better way to perform the fusion (figure 3.17).

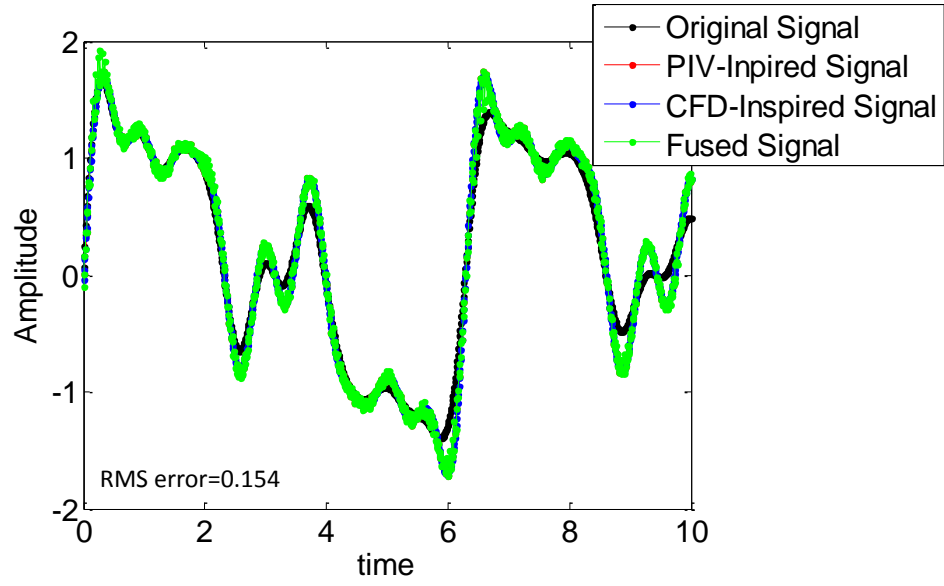


Figure 3.17: *Gappy POD Fusion Applied to a Downsampled and Noisy Signal*

3.3.3 Test Case 3

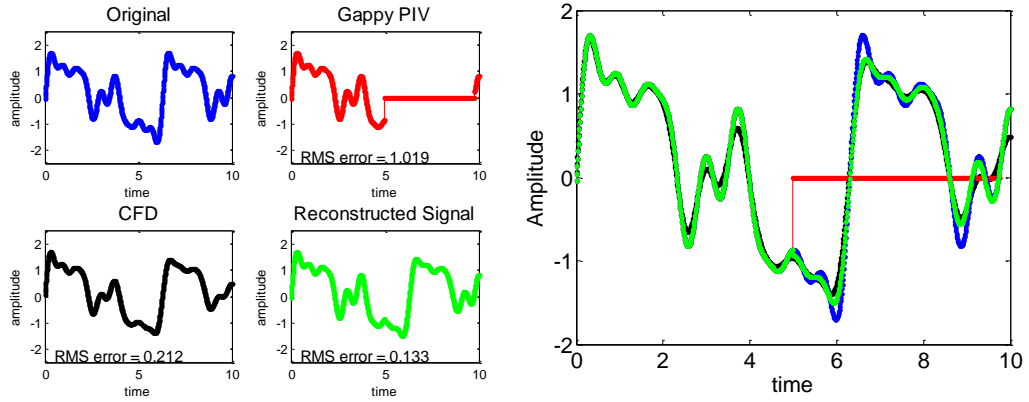


Figure 3.18: *Gappy POD Fusion Applied to a Gappy Signal*

Test case 3 benefits from the Gappy POD fusion technique the most. Using the Gappy POD fusion technique, the RMS error decreases. The fused signal is better than the PIV-inspired signal, which has no data in the gappy region, and it is also better than the CFD-inspired signal. The fused signal behaves more like the original

signal than the CFD-inspired signal (figure 3.18). One can see that the fused signal is capturing the peaks in the gappy region better than the CFD-inspired signal. The Gappy POD fusion technique works better than the wavelet fusion for this test case.

The Gappy POD fusion method works well in this case. However, if the CFD-inspired signal is very different from the PIV-inspired signal, then the region estimated by the Gappy POD looks too much like the CFD-inspired signal (figure 3.19). This is essentially what Everson and Sirovich [37] saw in their work with Gappy POD using a human face basis to repair a monkey image.

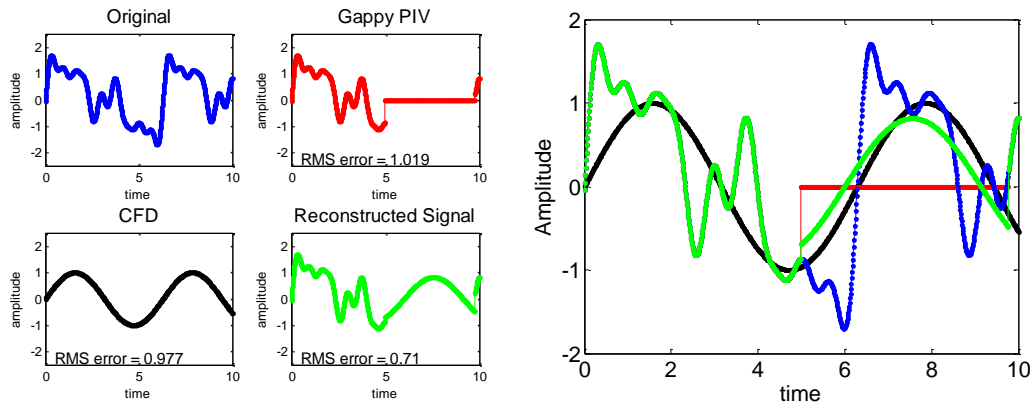


Figure 3.19: *Gappy POD Fusion Applied to a Gappy Signal with a “Bad” CFD-inspired Signal*

The last thing to mention when using Gappy POD is how many modes to use in the fusion process. The RMS error is minimized when the energy in the POD modes is nearly 100%. Looking at figure 3.20, one can see a plot of energy and RMS error as a function of number of modes used in the reconstruction. The RMS error is minimized when five modes are used in the reconstruction, which corresponds to a corner in the energy convergence plot. Any mode higher than this is really noisy and does not add any useful information. In fact, it actually makes the estimate worse.

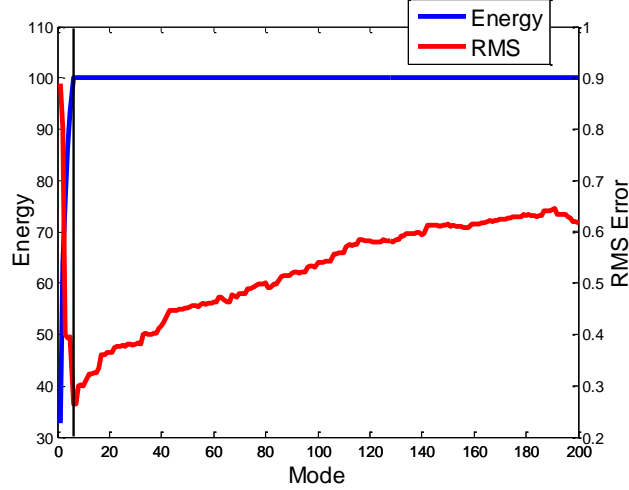


Figure 3.20: Convergence Rate of POD vs. the RMS Error

3.4 LSE Fusion

Wavelet fusion and Gappy POD fusion have been applied to the test cases. In this section, LSE (subsection 1.4.3) is used as a method of fusion. The fusion process works by using the CFD-inspired data to calculate the two point correlation tensor $\overline{(u_j(x)u_i(x'))}$ and the Reynolds stress $\overline{(u_j(x)u_k(x))}$. The two terms are calculated at the points where the PIV-inspired data is also known. The PIV-inspired data can then be used as the conditional signal.

The LSE fusion does not work as well as the other methods. If the first few points of the PIV-inspired signal are used in the estimate (figure 3.21), the fused signal is similar to the original signal at the points used in the estimate, but starts to diverge from the original signal after that region. Despite this, the fused signal and original signal still have similar trends. Using more PIV-inspired data for the estimate increases the RMS error, but the fused signal diverges later than the previous case (figure 3.22). When half of the PIV-inspired data is used in the estimate, the RMS error of the fused signal increases even more (figure 3.23).

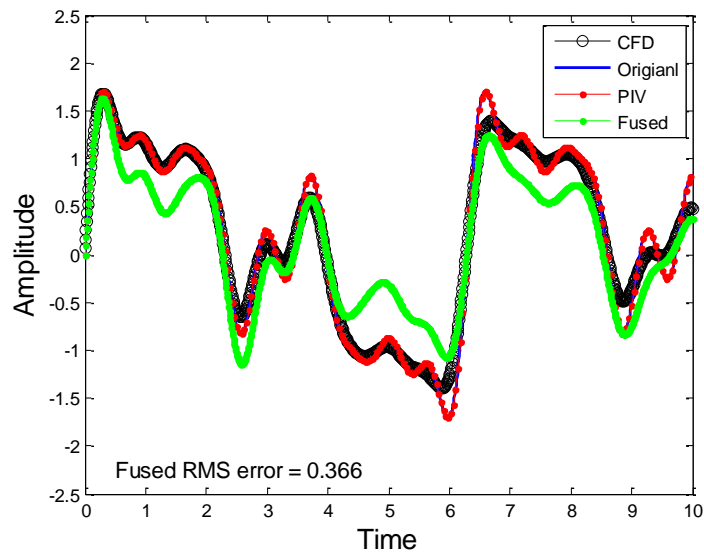


Figure 3.21: *LSE Fusion for a Downsampled Signal Using the First 12% of the PIV-inspired Signal*

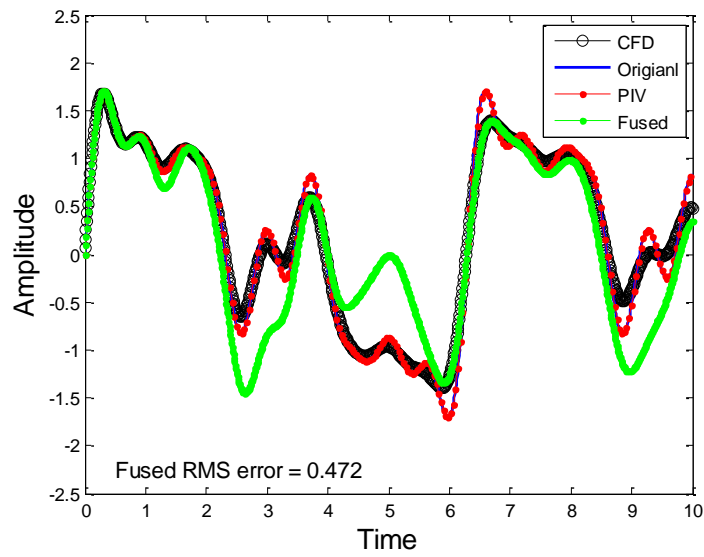


Figure 3.22: *LSE Fusion for a Downsampled Signal Using the First 25% of the PIV-inspired Signal*

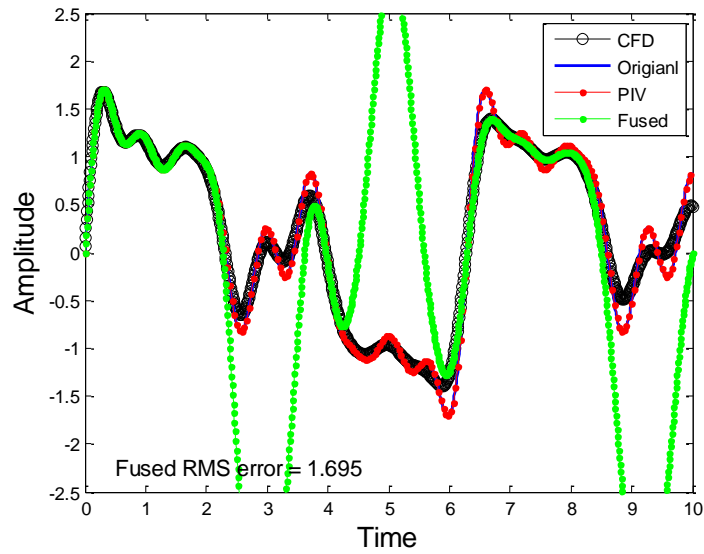


Figure 3.23: *LSE Fusion for a Downsampled Signal Using the First 50% of the PIV-inspired Signal*

One would expect that adding more information would create a better estimate, which is not the case here. The error increases in this case because the correlation between the PIV-inspired signal and CFD-inspired signal decreases with increasing lag as seen in figure 3.24.

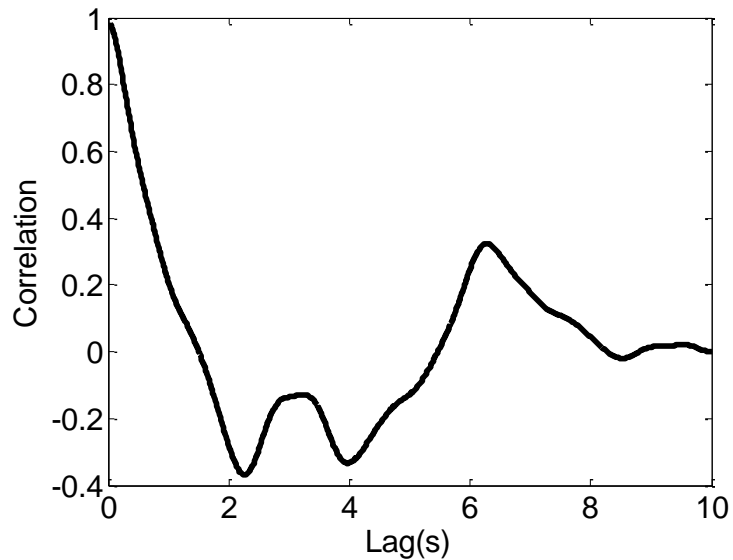


Figure 3.24: *Correlation Between the CFD- and PIV-Inspired data*

The fused signal matches well between one to two seconds, which is where the correlation is strong. The signals match again between six to eight seconds, which is where the correlation is “large” again. This is also the point where the flow is starting another period and should have a strong correlation.

3.5 Fusing Techniques Together for Test Case 4

The three different techniques have been applied to the different data sets and the strengths and weaknesses of each technique has been shown. In the spirit of fusion, the techniques discussed above have been fused together in a way that takes advantage of each of the techniques. In this method, the wavelet technique is used to compress the signals to the same resolution and smooth any discontinuities produced from splicing together the other techniques and reduce any noise that may exist in the experimental data. The Gappy POD fusion is used to fill missing information in the PIV-inspired signal and to extend the PIV domain when necessary. In future work, a modified LSE technique that uses pressure as a conditional signal [24, 57] could be added to give another estimate of the flow field. The fused technique is shown pictorially in figure 3.25. This idea is the basis for the fused techniques to be discussed in subsection 4.6.1 and subsection 5.5.4.

The fused technique is used to fuse the data described in test case 4, which includes; a dispersed signal (CFD-inspired) and, noisy, downsampled, multi-windowed, and gappy signal (PIV-inspired). The fused technique first uses a wavelet decomposition to compress the data to have the same resolution as all of the other signals as is shown by the “Compression 1” step in figure 3.25. A second compression is done on all of the data using the wavelet decomposition as shown in the “Compression 2” step to reduce any noise in the signal. POD modes are then calculated from the compressed CFD data (the black boxes). Those modes are then used to fill the missing information in the other signals using the Gappy POD method. Another compression

is performed on all of the data to remove any gradients caused by the Gappy POD method. The four compressed signals are now combined using a weighted average. The signals are then decompressed using detail information from the CFD-inspired signal using the inverse wavelet decomposition as is shown in the “Reconstruction” steps in figure 3.25.

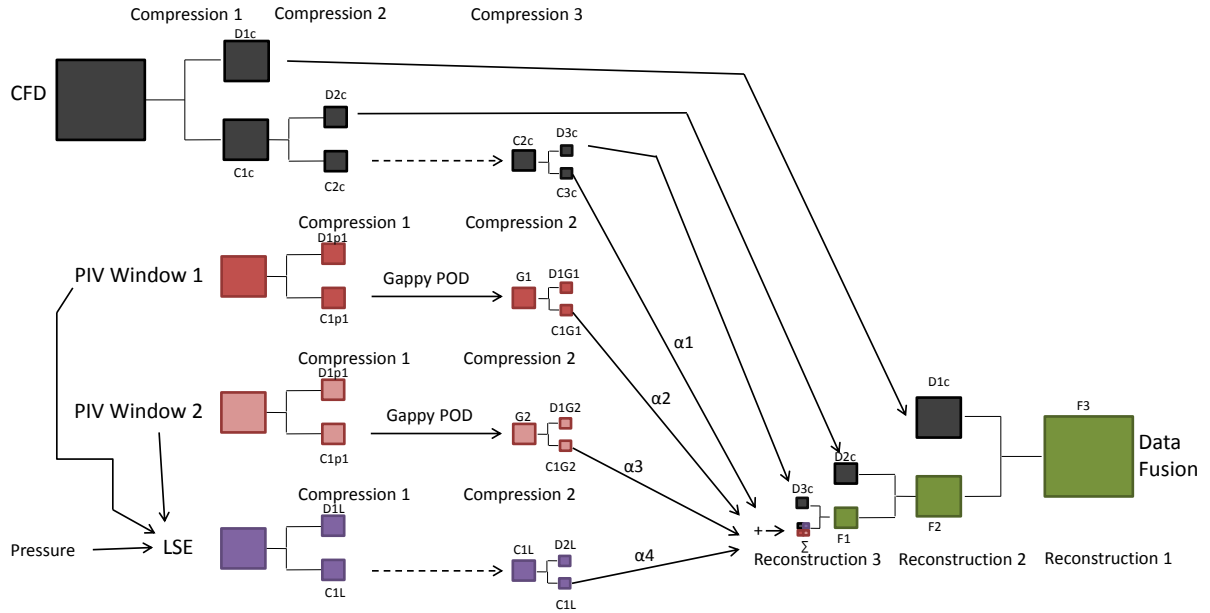


Figure 3.25: *Diagram of a Technique that uses Wavelet Fusion and Gappy POD*

Using this fusion technique creates a fused signal that is better in terms of RMS error than any signal on its own and the fused technique is better than using any one technique on its own (figure 3.26). In addition to reducing the RMS error, the fused signal also fills in gappy regions, reduces measurement noise, and combines two windows for the PIV-inspired data. The fused signal also reduces the dispersion error that is present in the CFD-inspired data.

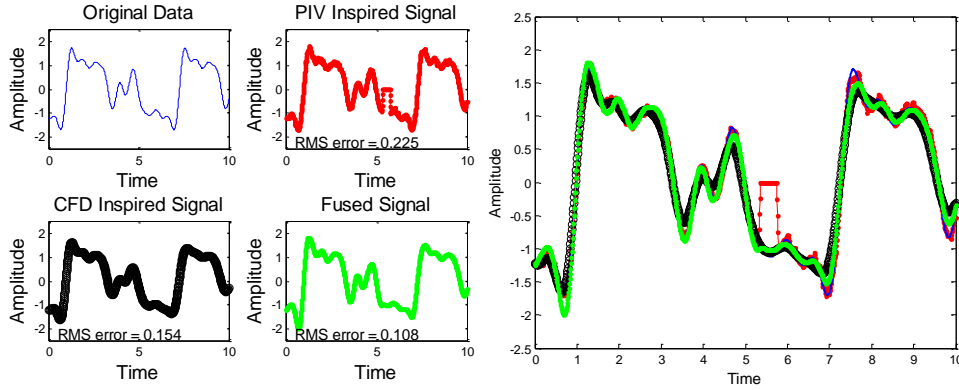


Figure 3.26: *Fusion of Downsampled, Noisy, Multiple Window, Gappy Data and Dispersed Data using the Fused Technique*

3.6 Summary of Fusion Techniques

Different techniques are used to fuse signals inspired by CFD and PIV. Most of the techniques produced a fused signal that is better than the signals used to create the fused signal. The wavelet fusion technique works very well in fusing two signals to increase the resolution of a noisy signal while reducing the noise or smoothing out discontinuities. Of the wavelets studied in this work, the Daubechies10 was found to be the best wavelet to perform the fusion. Each higher order Daubechies wavelet produced better results, but approaches an asymptote and minimal improvement is gained by using higher order Daubechies wavelets.

The Gappy POD fusion method fills in missing data well if the two signals have similar POD modes. If the two signals are too dissimilar the fusion process produces results that have higher RMS errors than both the CFD- and PIV-inspired signals.

The LSE fusion technique did not work very well with the given signals. The correlation matrices of the two signals are very different which creates a fused signal with large RMS errors. LSE might be used to create an estimate of the flow field using pressure sensors as the conditional measurement. This technique will be used later for the airfoil data and shows better results.

3.7 Inspiration for the Application of Fusion on Real Fluid Dynamic Data

The work presented in this chapter was used to determine which tools worked well at handling the different scenarios that could arise in fluid dynamic data obtained experimentally and/or numerically. Most of the techniques used on the synthetic data did not work on the real data because of issues that were not considered. One such issue is the use of wavelet fusion needs two images of the same flow field, which is not the case for the available data. The data was taken at different times because of limitations in the measurement techniques and could not be synchronized in time. Wavy POD, which is not applied to the real data because of time constraints was conceived to account for this by performing wavelet fusion in POD space instead of time which would eliminate the need to synchronize the data in the time domain.

Another issue experienced specifically in the airfoil data [chapter 4](#) is the CFD and PIV having different modes, making Gappy POD fusion produce results with gradients around the holes being estimated with the Gappy POD fusion. A wavelet fusion technique was added to smooth over the gradients caused by differences in the data from the two sources. The effect of this is discussed in [section 5.4](#) and is also discussed in [section 3.5](#) while fusing all of the techniques used on the synthetic data.

One final issue experienced is trying to perform Gappy POD fusion to stitch two PIV windows together that are not synchronized in time. The work in this chapter the data is able to be shifted and is easily synchronized in time due to the periodic nature of the synthetic data. In the real data, the PIV windows are not able to be synchronized in time since data is only correlated with itself for short periods of time. This was solved by using the second half of the data to build a model which could be used to estimate the time in conjunction with Gappy POD which was called DGP ([section 5.5](#)).

The issues listed are why most of the techniques applied to the synthetic data are

not used on the actual data. Despite this, the lessons learned from the synthetic data allowed for the development of the more sophisticated techniques.

Chapter 4

Fusion for Airfoil Data

The results from the synthetic data demonstrated that fusion can be used to improve data that comes from typical fluid dynamic data sources. Holes in the data were repaired using gappy POD. These holes represented seed dropout, reflections, or sensor occlusions that can occur in PIV data. Resolution of data was increased using a wavelet fusion technique and another data set with higher resolution. As stated, the techniques worked with synthetic data, but that data were generated using periodic functions and the data were one-dimensional. With that in mind, it is now important to show how fusion techniques work on real fluid dynamic data.

Ultimately fusion is applied to a high-speed jet flow. However, jet flows are very complicated and trying to apply fusion to the jet problem directly from the may prove to be too large of a jump. This would make it difficult to fully understand how fusion works with fluid dynamic data. In addition, the jet data were still being processed at the time this work started. To account for this, airfoil data was used as an initial attempt to apply fusion to real data, because the flow is less complex, and the data was already available.

In this chapter, the airfoil is introduced, the experimental and computational setup for the data is described, and fusion is applied to the data. It was discovered through this work, that some of the techniques applied to the synthetic data did

not work, or were invalid with this data set. New fusion techniques were developed based on the existing techniques to accomplish fusion because of this issue. One such technique that was developed is Fused POD and is described later in this chapter.

4.1 Definition of an Airfoil

Before fusing the airfoil data, it is important to know a little about them. An airfoil is a common geometrical structure in fluid dynamics and has been studied over the years since these are the means by which lift is produced for an airplane [108]. These were studied extensively in the 1900s by Abbott *et al.* [109], Jacobs *et al.* [110], and Theodorsen [111] of the National Advisory Committee for Aeronautics (NACA) to generate a database of lift and drag characteristics. One classification of airfoil is the NACA 4-series airfoils. Other types exist, but this work focuses on a NACA 4412, which is a 4-series airfoil, so this discussion focuses on this particular series of airfoil. The numbering of the 4-series airfoil correspond to the geometry and to understand this, one must know the basic parts of an airfoil, which are defined in figure 4.1.

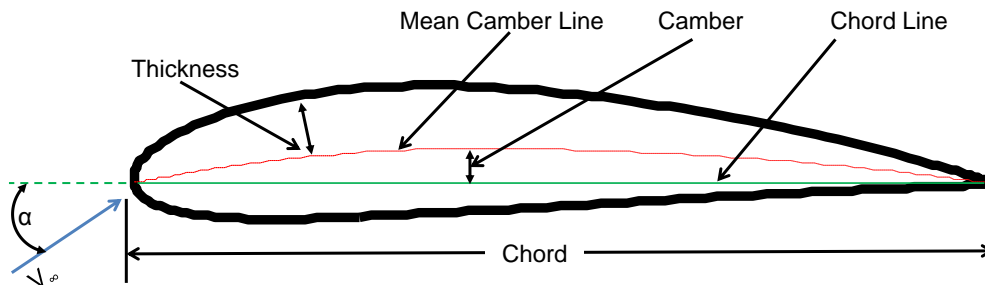


Figure 4.1: *Airfoil Nomenclature*

The camber line is a line equidistance between the top and bottom surface of the airfoil. The chord line is a line that connects the leading edge (front) and the trailing edge (back). This line is horizontal and is not restricted to be contained inside the airfoil. The next things to define is the chamber, which is the distance between the chord line and the mean camber line. The thickness of an airfoil is the perpendicular

distance between the mean chamber line and the top and bottom surface of the airfoil. The last thing to describe is the angle of attack (α), which is the angle the airfoil makes with the freestream velocity.

With that information and the number of the airfoil, the geometry can be described. The first digit represents the maximum camber in percent chord; the second digit represents the location of the maximum chamber multiplied by 10 in percent chord, and the last two digits represent the maximum thickness in percent chord. So the NACA 4412 airfoil has four percent maximum chamber located at 40% of the chord and has a maximum thickness of 12% of the chord. The geometry for a NACA 4-series airfoil can be generated based on the information in the naming convention using

$$y_c = \begin{cases} \frac{m}{p^2}(2p\frac{x}{c} - \frac{x^2}{c}), & \text{if } 0 \leq x \leq p \\ \frac{m}{(1-p)^2}(1 - 2p + 2p\frac{x}{c} - \frac{x^2}{c}), & \text{if } p \leq x \leq c \end{cases} \quad (4.1)$$

where m is the maximum camber, p is the location of the maximum camber, x is the position, y_c is the camber line and c is the chord length. To add thickness to the airfoil

$$x_U = x - y_t \sin(\theta) \quad (4.2)$$

$$y_U = y_c + y_t \cos(\theta) \quad (4.3)$$

$$x_L = x + y_t \sin(\theta) \quad (4.4)$$

$$y_L = y_c - y_t \cos(\theta) \quad (4.5)$$

$$y_t = \frac{t_a}{0.2}(0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (4.6)$$

are used, where x_U is the x-coordinate on the upper surface, x_L is the x-coordinate on the lower surface, y_U is the y-coordinate on the upper surface, y_L is the y-coordinate on the lower surface, y_t is the thickness, t_a is the thickness in percent chord, and

$$\theta = \arctan\left(\frac{dy_c}{dx}\right) \text{ [110].}$$

4.2 Flow Over an Airfoil

Lift is generated by air moving faster on the top surface of the airfoil than on the lower surface. This produces a pressure difference since the faster air has a lower pressure than slower moving air. Suction is created on the top of the airfoil due to the pressure difference, producing lift. The reason the airfoil causes the air to move faster on the top surface is due to the Kutta condition, which prohibits infinite acceleration around the trailing edge of the airfoil due to viscosity [108]. As the angle of attack of the airfoil increases, the lift increases up until a point. At some angle of attack the flow over the airfoil separates, causing a loss of lift. This is known as stall as is shown in figure 4.2. One can see the vortices that form on the surface of the foil. The region where it occurs is the separation region.

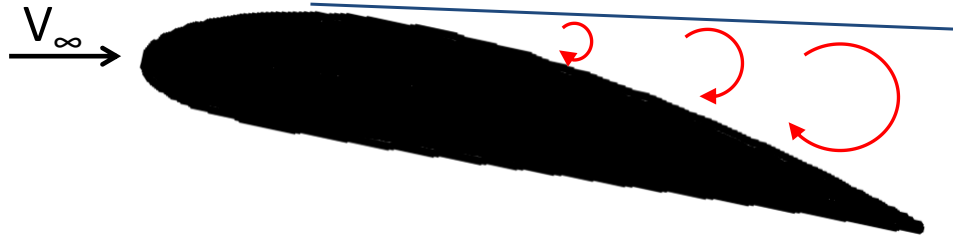


Figure 4.2: *Diagram of Stall on a NACA 4412 Airfoil*

Airfoils have been studied for many years as stated earlier in this chapter and airfoil theory can be found in any aerodynamic textbook such as Anderson [108], Bertin [112], Robinson and Laurmann [113], and Abbott and Von Doenhoff [114]. Since the focus of this work is data fusion and not airfoils, a comprehensive review of airfoil theory is not included in this document. The purpose of this section is to give the reader enough information to understand the basics of an airfoil.

4.3 Experimental Setup

The data used in this chapter was acquired by Pinier *et al.* [24] and Ausseur [67] to reduce the separation on a stalled NACA 4412 airfoil using flow control. Reducing the separation causes an increase in the maximum lift coefficient by allowing an airfoil to operate at higher angles of attack. This could allow shorter takeoff and landing distances, reduce the effect of drag due to pressure difference and create a more efficient airfoil, ultimately decreasing the fuel burned by an aircraft. Ausseur [67] also states that flow control on the airfoil may replace the typical control surfaces (aileron, elevator, and rudder) by changing the flow characteristic to mimic the effect of the controls. The removal of the control surfaces could potentially have weight saving effects due to the removal of mechanical components used to drive the controls.

The experimental work was performed in the Syracuse University subsonic wind tunnel, which has a test section of 2ft x 2ft x 8ft. The airfoil has a span of about 2ft, so it spans the tunnel to neglect three-dimensional effects and has a chord of 0.67ft. The test velocity is 32.8ft/s (10m/s), corresponding to a Reynolds number based on chord of 135,000. Multiple angles of attack ranging from 10° to 18° were tested in the experiment. The flow field was measured using a Dantec Dynamic PIV system which used two charge coupled device (CCD) cameras (1280 x 1024 pixels) and a pulsed New Wave Research 200mJ Neodymium-Doped Yttrium Aluminum Garnet (Nd:YAG) laser as stated by Pinier *et al.* [24]. The PIV system is traversed with the airfoil so the velocity field is always in the frame of the airfoil. The test setup for the PIV system is displayed in figure 4.3, which is figure 2 of Pinier *et al.* [24].

In addition to PIV, surface pressure was measured for the airfoil using 11 unsteady integrated circuit piezoelectric (ICP) pressure sensors from PCB Piezotronics. The sensors were placed on the suction (top) surface of the airfoil at 29% of the chord to 78% of the chord using even spacing as seen in figure 4.4 courtesy of Pinier *et al.* [24].

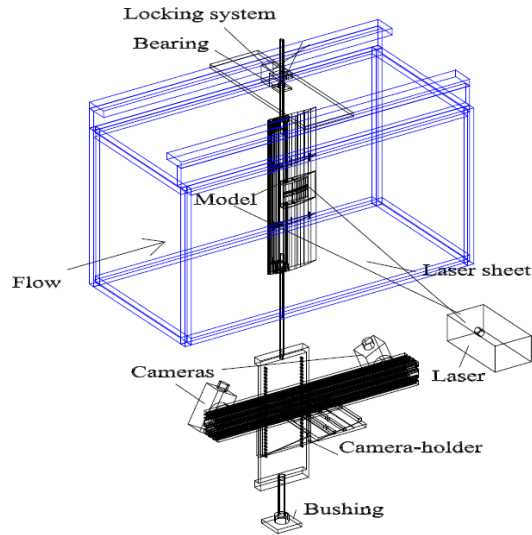


Figure 4.3: *Diagram of the Experimental Setup for the NACA 4412 Test*

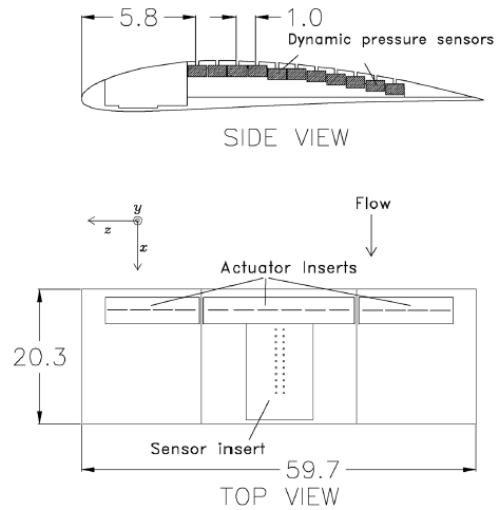


Figure 4.4: *Schematic of the Pressure Sensors on the NACA 4412 Airfoil*

The experiment focuses on understanding the baseline flow as well as applying flow control via piezoelectric disks (operating as synthetic jet actuators). As such, many different data sets were collected for the various angles of attack as well as the different control cases. Data fusion is applied to the baseline case at an angle of

attack of 16° in this chapter.

4.4 Computational Setup

In addition to the experimental work, a simulation of the airfoil at an angle of attack of 16° is performed to fuse with the experimental data. The simulation is performed using the NASA flow solver OVERFLOW [115], which is capable of using OVERSET grids [116, 117].

4.4.1 Grid Generation

The airfoil grid is generated using transfinite interpolation (TFI), which is an algebraic technique for structured grids [118]

$$\begin{aligned} x(i, j) = & (1 - \xi)x(1, j) + \eta x(i, J) + \xi x(I, j) + \\ & (1 - \eta)x(i, 1) - (1 - \eta)(1 - \xi)x(1, 1) - \\ & (1 - \xi)\eta x(1, J) - \xi(1 - \eta)x(I, 1) - \eta\xi x(I, J) \end{aligned} \quad (4.7)$$

where ξ and η are the grid coordinates, x is the physical coordinate, I and J are the largest index, and i and j are the index. The coordinates are described in figure 4.5.

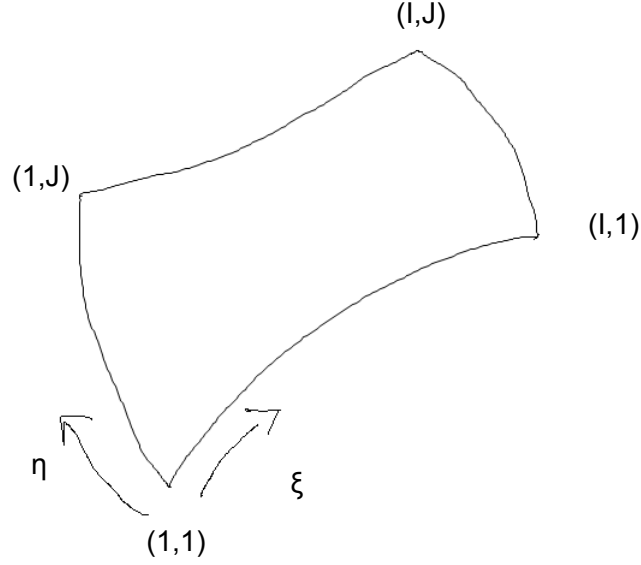


Figure 4.5: *Definition of the Coordinate System for the TFI Algorithm*

The grid generated by the TFI algorithm can then be used as a starting point for an elliptic grid generator.

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0 \quad (4.8)$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0 \quad (4.9)$$

Equations 4.8 and 4.9 need to be transformed into equations that can be solved for x and y .

$$a \frac{\partial^2 x}{\partial \xi^2} - 2b \frac{\partial^2 x}{\partial \xi \partial \eta} + c \frac{\partial^2 x}{\partial \eta^2} = 0 \quad (4.10)$$

$$a \frac{\partial^2 y}{\partial \xi^2} - 2b \frac{\partial^2 y}{\partial \xi \partial \eta} + c \frac{\partial^2 y}{\partial \eta^2} = 0 \quad (4.11)$$

where

$$a = \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 \quad (4.12)$$

$$b = \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} \quad (4.13)$$

$$c = \left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 \quad (4.14)$$

The elliptic grid equations are typically solved using an iterative technique. The elliptic grid algorithm improves the grid quality by creating grids that are more orthogonal than those created by the TFI algorithm.

The grid generated for the airfoil consists of roughly 200,000 nodes, extending 2 chords upstream, 3 chords downstream, and 1.5 chords above and below the airfoil; a subsection is displayed in figure 4.6. The grid displayed in figure 4.6 was down-sampled to show the structure of the grid; the actual grid is more dense, by a factor of 10. The extent of the computational domain is rather small so some errors are expected in the CFD; however, the purpose of the simulation is to fuse it with the experimental data. The small domain is also chosen because of computational costs and the limitations of the computational resources at hand.

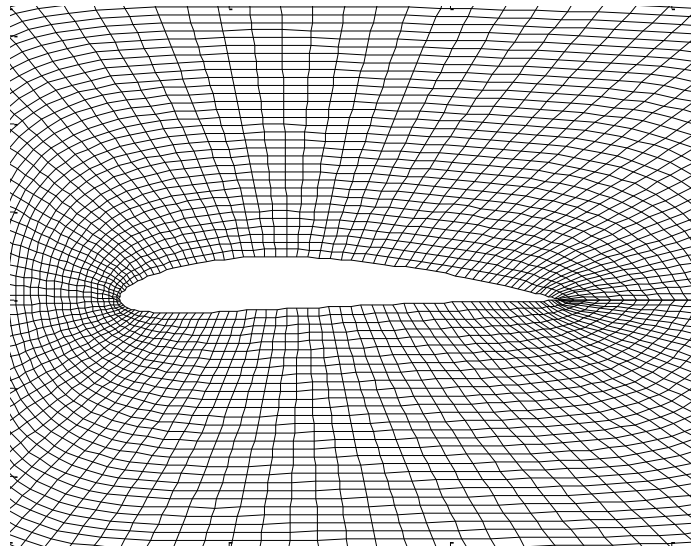


Figure 4.6: *Airfoil Computational Grid*

4.4.2 Computational Setup

With the grid, the flow field can be simulated using CFD by discretizing the Navier-Stokes equations, which consist of mass balance,

$$\frac{\partial \rho}{\partial t} + \frac{\partial u_i}{\partial x_i} = 0 \quad (4.15)$$

momentum balance,

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \quad (4.16)$$

and energy balance

$$\frac{\partial e}{\partial t} + \frac{\partial u_i e}{\partial x_i} = -\frac{\partial u_i p}{\partial x_i} - \frac{\partial q_{x_i}}{\partial x_i} + \frac{\partial u_j \tau_{jk}}{\partial x_i} \quad (4.17)$$

The shear stress (τ_{ij}) can be approximated as the rate of strain or velocity gradient

$$\tau_{ij} = \mu \frac{\partial u_i}{\partial x_j} \quad (4.18)$$

if the fluid is Newtonian. A fluid is Newtonian if the stress is proportional to the rate of strain and the stress for zero strain is zero.

The equations can be discretized and the flow field can be solved. One needs to discretize in such a way that all scales are resolved in both time and space. This becomes a problem for turbulent flows, which have many scales that need to be captured; this can require very fine resolution. To work around this, one usually models the smaller scales with turbulence modeling. A common turbulence modeling technique is eddy viscosity modeling, which models the turbulence as an additional stress term. The eddy viscosity is modeled in many different ways, ranging in complexity from simple algebraic equations to complex transport equations. A few models include the mixing length, Spalart-Allmaras, $k - \epsilon$, $k - \omega$, and the shear stress transport (SST) models (combination of $k - \epsilon$, and $k - \omega$ [119]). These techniques are known as

Reynolds Averaged Navier-Stokes (RANS) turbulence models. These models break the flow into a mean and fluctuating part; this is known as Reynolds decomposition. When performing the Reynolds averaging, a new term is created in the equations. This term is generally thought of as a turbulent stress and is called Reynolds stress ($\frac{\partial \overline{u'_i u'_j}}{\partial x_j}$), which is modeled using eddy viscosity.

$$\frac{\partial u_i}{\partial t} + \frac{\partial \overline{u'_i u'_j}}{\partial x_j} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \quad (4.19)$$

Large Eddy Simulation

A more complex way of handling turbulence is large eddy simulation (LES), which filters the flow equations into turbulence that is modeled and turbulence that is simulated. Unlike the RANS equations which are temporally filtered, the LES equations are spatially filtered. The most common type of filter used in LES is a Top-hat filter [4].

$$G(\mathbf{x}, \mathbf{x}', \Delta) = \begin{cases} 1/\Delta^3 & |\mathbf{x} - \mathbf{x}'| \leq \Delta/2 \\ 0 & |\mathbf{x} - \mathbf{x}'| > \Delta/2 \end{cases} \quad (4.20)$$

G is the filter function, Δ in equation 4.20 is the cutoff width of the filter which is based on the size of the grid. This is usually defined as $\sqrt[3]{\Delta x \Delta y \Delta z}$ since any structures smaller than the grid cannot be resolved. Any turbulence smaller than that is modeled using a sub-grid model. Usually, the large scale turbulence is more specific to the type of flow being simulated and the smaller scale turbulence is usually more general. LES usually produces a more accurate simulation of the flow because of this. However, it takes more computational resources to perform an LES than a RANS simulation due to the increased number of nodes required to resolve the large eddies.

The Navier-Stokes equations must be filtered in order to use the LES technique. The filtering process breaks the flow into two parts (resolved scales and the sub-grid

scales), which is similar to Reynolds averaging used for RANS turbulence modeling.

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial}{\partial x_j} \left((\bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j) + (\overline{u'_i \bar{u}_j} + \overline{\bar{u}_i u'_j}) + \overline{u'_i u'_j} \right) \quad (4.21)$$

The first term in equation 4.21 is the time rate of change of the filtered velocity; the second term is the transport of the filtered velocity by convection; the third term is the filtered pressure gradients; the fourth term is the transport of filtered velocity by diffusion. The last three terms are caused by the sub-grid scale stresses. The fifth term (first term of sub-grid scale stress) is a result of the double filtering and the fact that double filtering is not the same as filtering only once. The sixth term (second sub-grid scale term) is caused by the interactions of the sub-grid stresses and resolved stresses. The last term is similar to the Reynolds stress, and is the interaction between the unresolved scales. The sub-grid stresses can then be modeled using various modeling techniques. A common technique for modeling them is the Smagorinsky model [120]. This models the sub-grid stress by assuming they are proportional to the local rate of strain of the resolved flow [4].

$$\tau_{ij} = -\mu_{SGS} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) + \frac{1}{3} \tau_{ii} \delta_{ij} \quad (4.22)$$

where

$$\mu_{SGS} = \rho (C_{SGS} \Delta)^2 \sqrt{\frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)} \quad (4.23)$$

Detached Eddy Simulation

Another way to handle the turbulence is detached eddy simulation (DES), which is a combination of LES and RANS modeling developed by Spalart [121] in the 1990s to simulate higher Reynolds number flows. A DES is a step between LES and RANS. At the one extreme, RANS can simulate very high Reynolds number flows by modeling all of the turbulence. At the other extreme, LES is limited by computing power since the LES resolves the larger scales of a flow and models the smaller more general scales.

To combine the two, a DES uses a switching function,

$$FDES = \max\left(\frac{L_t}{C_{DES}\Delta}, 1\right) \quad (4.24)$$

where C_{DES} is a constant (order 1) [122], L_t is the length scale, and Δ is the grid size, to determine how much turbulence to resolve. The switching function ($FDES$) serves as a multiplier, which modifies the dissipation term in the kinetic energy equation. If the dissipation term increases, the energy modeled by the kinetic energy equation decreases, which decreases the eddy viscosity term in the momentum equation meaning more turbulence is simulated instead of being modeled. So if the grid is too coarse, RANS is used at that location; otherwise LES is used. Usually RANS is used near an object since the integral length scales are much smaller in that region. This method of using both RANS and LES allows for much higher Reynolds number simulations with the incorporation of much more physics than using either method by itself.

In the simulation of the airfoil, the turbulence is initially handled using the SST RANS model to simulate the initial transient solution. The simulation switches to a DES once the aerodynamic forces stabilize (mean lift and drag do not change more than 5%).

Boundary Conditions

The boundary conditions for the airfoil simulation include setting the velocity at the inlet of the domain to match that of the experimental work, the exit of the domain is modeled as a pressure outlet, the top and bottom of the domain are modeled as walls, and the airfoil surface is modeled as a wall.

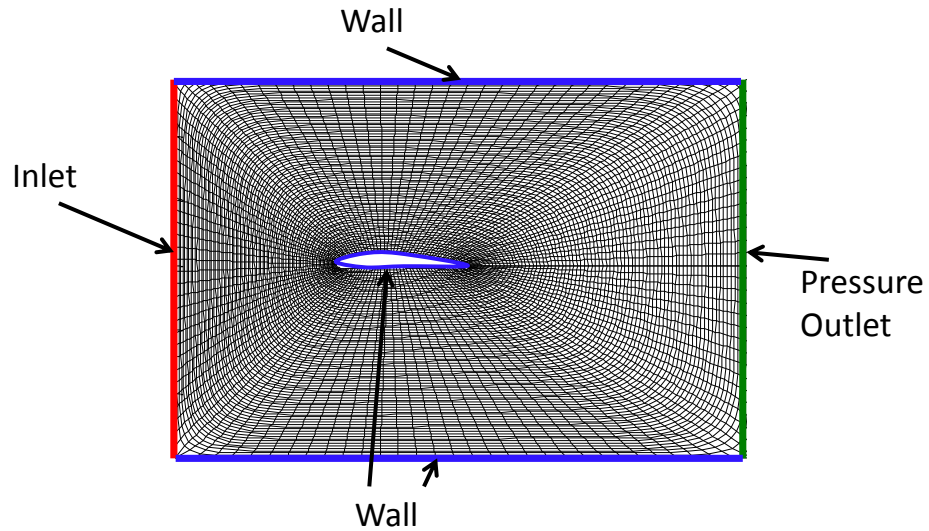
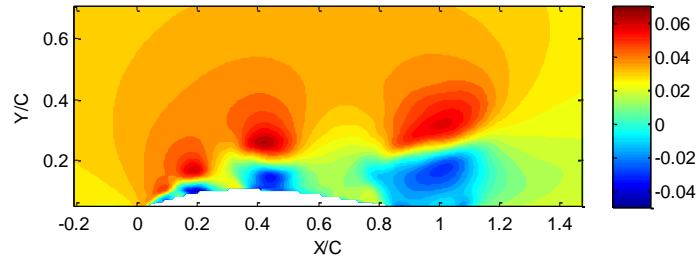


Figure 4.7: *Airfoil Boundary Conditions*

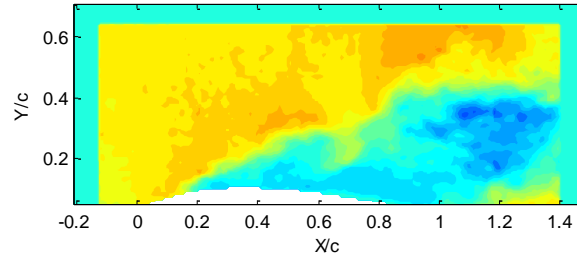
Simulation Results

The results of the simulation are displayed in figure 4.8, which shows an instantaneous snapshot of streamwise velocity for the flow over the NACA 4412 airfoil at an angle of attack of 16° and is compared with an instantaneous snapshot of PIV data.

One should notice the CFD and PIV results are qualitatively similar. Both show a recirculation region near the surface of the airfoil and show structures of similar size being shed from the surface. It should also be stated that the snapshots are not synchronized in time so one should not expect to see the same flow field between the CFD and PIV. One last thing to state is that the CFD domain is larger than that of the PIV and the PIV has a blue color around the edge of the data that represent a region of space where velocity was not measured by PIV but was by the CFD.

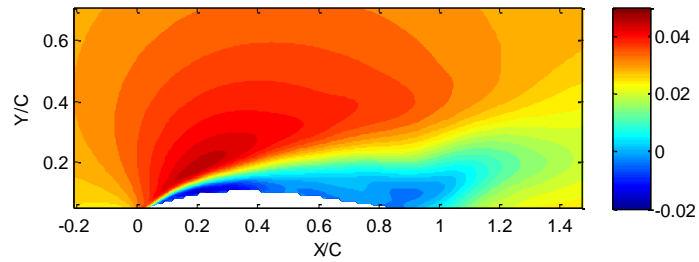


(a) CFD

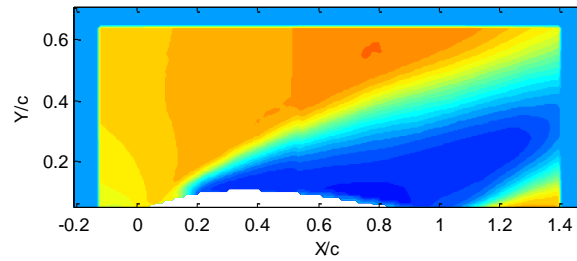


(b) PIV

Figure 4.8: *Instantaneous Streamwise Mach Number for a NACA 4412 Airfoil at 16° AOA Obtained with CFD and PIV*



(a) CFD



(b) PIV

Figure 4.9: *Comparison of Mean Streamwise Mach Number for a NACA 4412 Airfoil at 16° AOA Obtained with CFD and PIV*

The mean velocity is also similar; a separation region is present in both flow fields. However, the separation region on the CFD is smaller than the PIV, which could be an issue with the computational domain being too small, not enough resolution, or poor modeling of the turbulence. Despite the differences, the CFD is still useful and can be used in data fusion, since the two sources of data are showing the same qualitative results.

4.5 Data Fusion Applied to the Airfoil

With the PIV data, CFD data, and surface pressure, a few different techniques are applied to fuse the data together. This chapter focuses on how to fill artificial holes in the PIV, used to represent sensor occlusions, reflections, and dropouts, all of which are seen in the jet data. This tests how well CFD data performs at estimating missing data in PIV data.

The pressure and PIV data are used to determine the time dependence of the flow using sLSE. The time dependence is also estimated by using CFD data to train a stochastic model that can be used with the PIV data. The pressure and the CFD both have time dependence, but the PIV data does not, albeit the PIV does have good spatial resolution. The spatial information from the PIV and the time information from the pressure and CFD are fused using various techniques.

These are shown pictorially in the Venn diagram in figure [4.10](#)

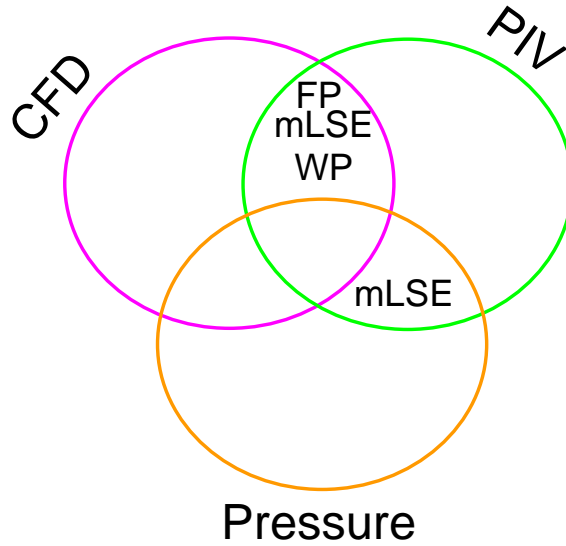


Figure 4.10: *Venn Diagram Showing the Different Fusion Techniques applied to the Airfoil Data*

where FP is Fused POD, which is discussed in the [subsection 4.6.1](#), mLSE is the modified stochastic estimation [subsection 1.4.3](#), and WP is wavy POD which is explained later in the document [section 6.3](#).

4.6 Hole Filling

The first thing to study is how to fill missing data from real fluid dynamic sources. In the synthetic data ([chapter 3](#)), Gappy POD was used. However, Gappy POD is not applicable in some cases. To account for this, a new technique called Fused POD was developed.

4.6.1 Fused POD

Fused POD is a new data processing technique developed to improve fluid dynamic data. It estimates missing experimental data caused by sensor occlusions, reflections, or poor data as shown in [figure 4.11](#). The technique is inspired by the Gappy POD

technique and work done by Wilcox [42], Vendl and Faßbender [96], and Zimmermann [95]. Wilcox used Gappy POD and basis functions from CFD to predict aerodynamic performance on a wing; Zimmermann used a modeling technique based on CFD modes to estimate steady aerodynamics; Vendl used Gappy POD to get the pressure distribution on a wing-body configuration for steady flow.

Gappy POD and Fused POD both use a linear regression to determine the POD coefficients that best fit the data.

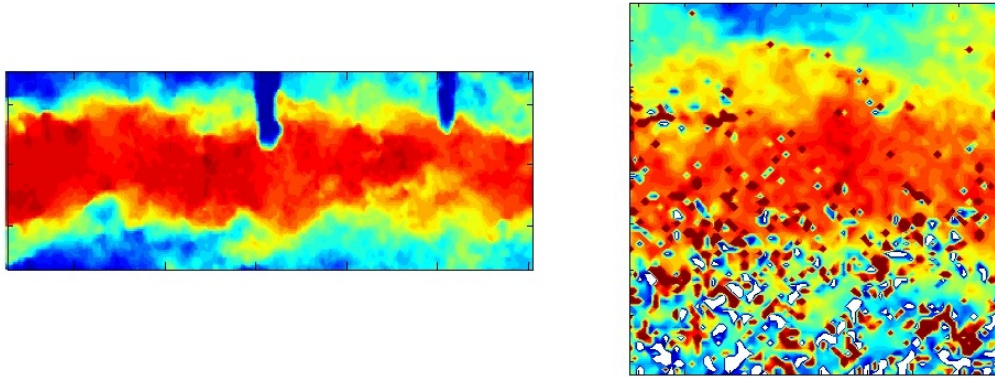


Figure 4.11: *Errors That Can Arise in PIV Data*

However, the application of the techniques are very different. The Gappy POD technique uses only the corrupted data set to estimate the “bad” data. The Gappy POD procedure requires that only a few bad snapshots exist, meaning the statistics of the rest of the data are “good” enough to calculate the POD modes. Then those modes can be used to estimate the data in the “bad” snapshots. Gappy POD can also be used if all the snapshots have missing data randomly distributed in time and space via an iterative process. (For more information on Gappy POD see [subsection 1.4.2.](#))

If corruption exists at the same spatial location in all snapshots, one would not be able to use Gappy POD because the statistics are unknown in that area; this is where Fused POD is needed. In Fused POD, one obtains the POD modes from another source. The other source can be another set of data that is capturing the

same phenomena or from the same data set and some physical knowledge of how the data is related. This gives statistical information about the data in the missing regions allowing estimation in that region.

The other difference between the two techniques is the implementation of a wavelet fusion process to help smooth some of the data. When filling in the data, a discontinuity can form, which is a product of the estimated data not fitting the actual data perfectly, and the fact the estimated data is low-dimensional and missing high energy information. The wavelet fusion helps alleviate this problem (figure 4.13).

A diagram of Fused POD is shown in figure 4.12. The red box (data set “B”) is data that has gaps at the same spatial location in all of the data. These gaps are represented by the white box in the middle of the red box. The black box (data “A”) comes from a source of data that does not have any gaps. The data from “A” should be statistically similar to the data in “B” since data set “A” is used to generate the POD modes (ϕ). With the POD modes from “A”, a linear regression is performed to fit data “B” to the modes of “A”, using the same algorithm as Gappy POD. The regression gives the POD coefficients that are then used with the POD modes of “A” to reconstruct an estimate of the data from “B” (green box). The data from the estimate of “B” is inserted into the holes in data set “B”, which produces the repaired data (red and green box). However, there can be discontinuities in the repaired data, especially if there is much difference in the data. To alleviate this issue, an image fusion technique is applied using wavelet fusion. The estimate and the repaired data are fused together using a wavelet decomposition. The wavelet fusion is accomplished by decomposing the repaired data and estimate into compressed and detail information. Since these are different representations of the same image, the compressed information from repaired data and the detail information from the estimate can be used to reconstruct a fused image. The swapping of the detail information is a primitive method. More sophisticated techniques can be applied in future works. Some of these methods are described in Ramac *et al.* [123]. One might be able to take a

linear combination of the data in wavelet space based on the confidence in the data at various points in space. The fused data no longer has a hole and is smooth. An example of the wavelet fusion is shown in figure 4.13. The repaired data on the top is fused with the estimate on the bottom producing a smooth image.

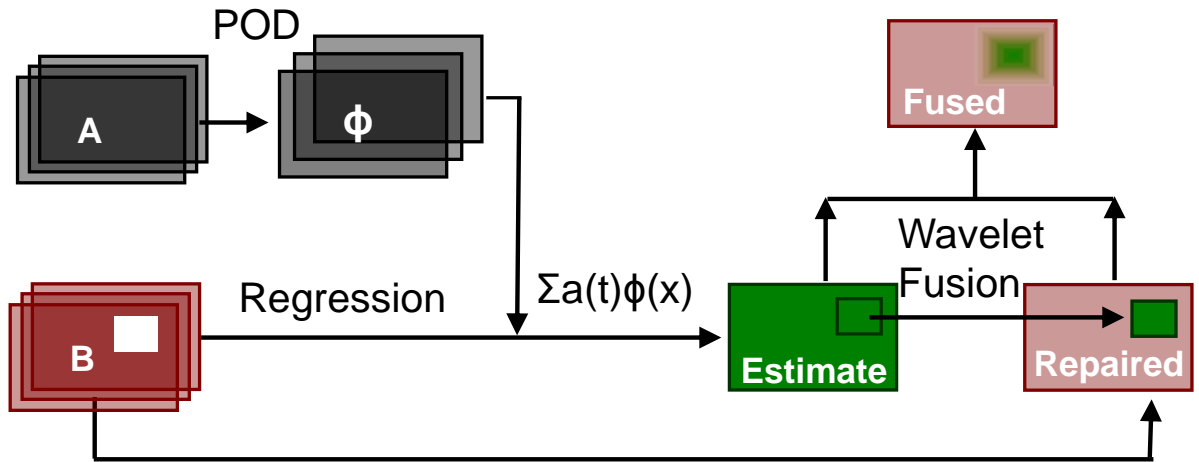


Figure 4.12: Diagram of the Fused POD Technique

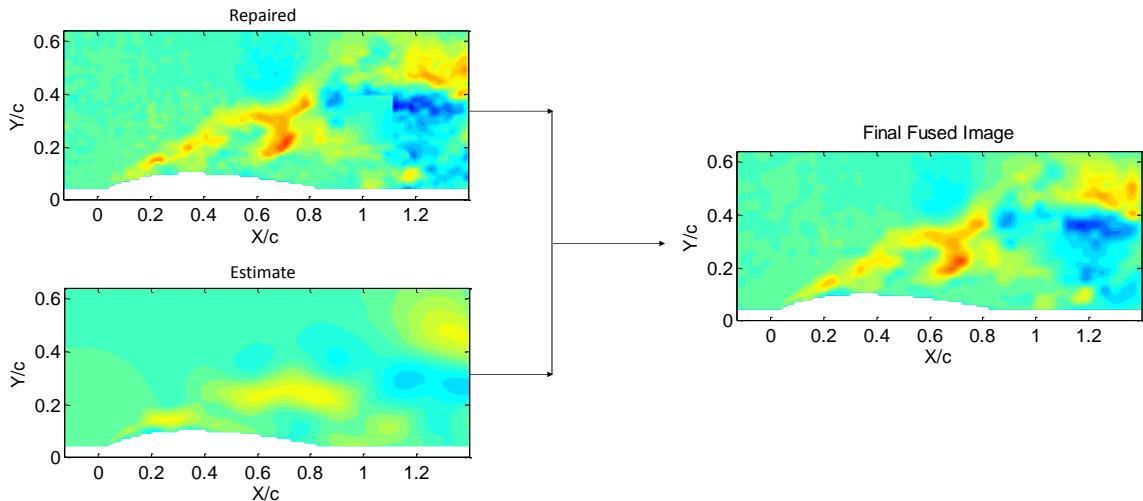


Figure 4.13: Gappy POD and Wavelet Fusion for a NACA 4412 Airfoil

4.6.2 Application to Airfoil Data

Now this technique is applied to real data but, the PIV data for the airfoil does not have any holes due to reflections, sensor occlusions, or dropout. However, the jet data does and to test this technique artificial holes are placed in the airfoil PIV data. This will test how well a CFD solution is able to fill missing data and see if Fused POD works with real fluid dynamic data.

The first thing to discuss is how well the POD modes from the PIV and CFD match. If the POD modes are too different, the technique does not work well (can only estimate the data with the energy in the similar modes). In other words, the modes need to have similar trends for Fused POD to work. A correlation plot between the PIV and CFD, seen in figure 4.14, show the modes are not well correlated. The correlation between the CFD and PIV modes based on the inner product. If the modes are exactly the same, one would see a line at a 45° angle. This would mean mode one from the CFD data would have a correlation of 100% with mode one from the PIV data and would not have any correlation with the other modes. However, looking at figure 4.14, one can see this is not the case. Only the first few modes have a “good” correlation.

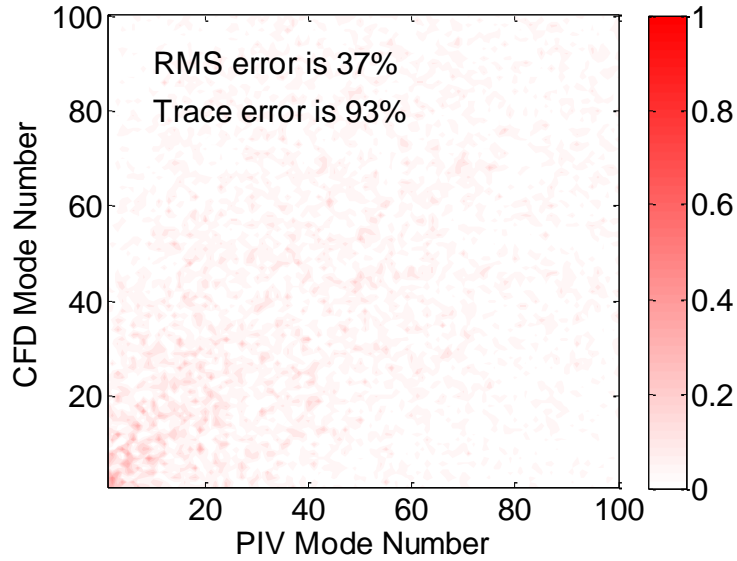


Figure 4.14: *Correlation between CFD and PIV POD Modes*

The first five POD modes have a correlation of about 50%. After mode five the correlation drops off rather rapidly. The low correlation could be due to the lack of time support for the CFD data (only have 0.6s worth of data which is approximately 30 independent snapshots due to the computational limit of the available machines and time constraints) or modeling issues. Despite the poor correlation, the modes are still used to fill the artificial holes since the first five modes still seem to match well. The effect of using better modes is discussed in more detail later in this section and [section 5.4](#).

Since the first five modes match, the next thing to discuss is how to use Fused POD on the airfoil data. The CFD data is in the same spatial location as the PIV data, which is imperative for this technique to work. The two data sets are fused by letting the CFD be “A” in figure 4.12 and the PIV data are “B”. Using the modes from the CFD data and the Fused POD technique, the missing data in the PIV can be estimated.

Now the technique can be applied to determine the effects of the size of the hole. As expected, the larger the hole, the larger the difference between the actual and

estimated data (figure 4.16). If less data is used for the fit, the fused data is less accurate. This is again discussed in greater detail in section 5.4.

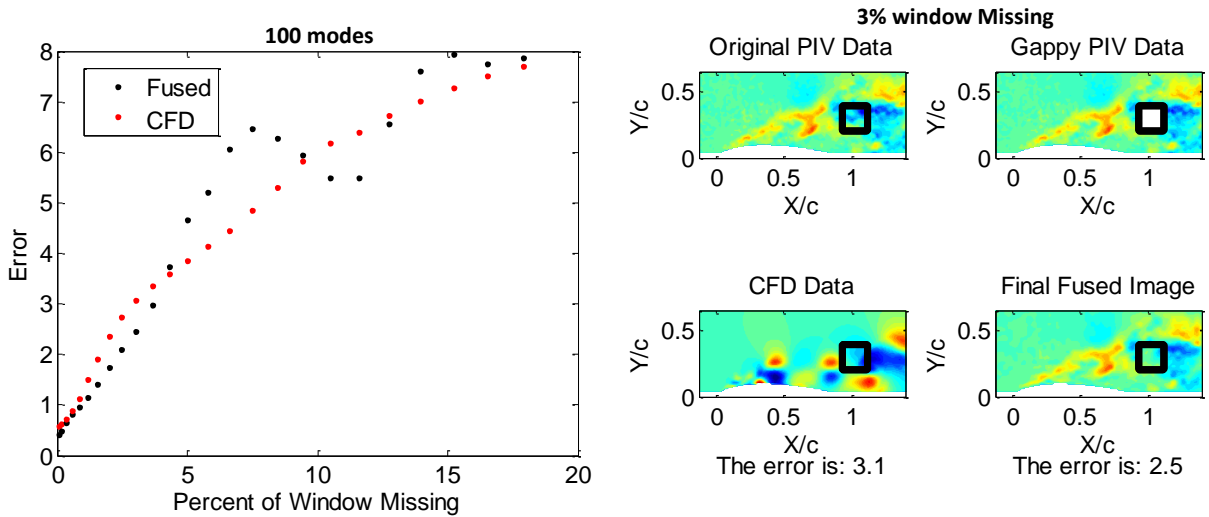


Figure 4.15: *Effect of Hole Size on Fused POD for PIV and CFD of a NACA 4412 Airfoil*

The fused data is better (in terms of RMS error) than the CFD data for windows that are smaller than 5% (~ 1 length scale) of the entire domain when using 100 modes in the reconstruction. If the hole is larger than one length scale, the information surrounding the hole is no longer correlated with what should be in the hole. What happens if more or fewer modes are used in the fusion process? It turns out that 100 is the optimal number of modes for the available airfoil data.

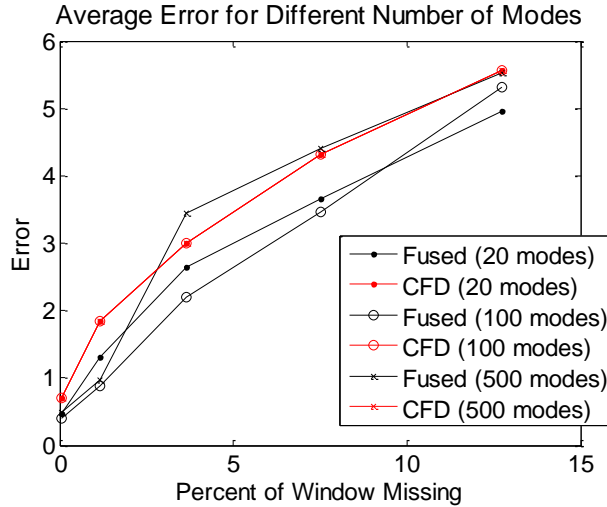


Figure 4.16: *Effect of Number of Modes on the Estimation of Missing Data for a NACA 4412 Airfoil using Fused POD with CFD Modes*

The convergence plot (figure 4.17) is constant after 100 modes, implying the information contained in the higher modes is very high wave number information and/or noise. If more modes than this are used the noise can corrupt the fusion process. If fewer modes are used, not all of the energy is captured which can cause errors. This is seen in figure 4.16, which shows the effect of the number of modes and the effect of window size on the fused data. It makes sense that there is an optimal number of modes to use when using CFD data and PIV data for fusion since the CFD is not going to capture the small scale flow features correctly and the PIV has noise. The small scale structures show up in the higher modes, meaning only the low energy modes of the CFD and PIV match well. A similar issue was shown with the synthetic data (chapter 3).

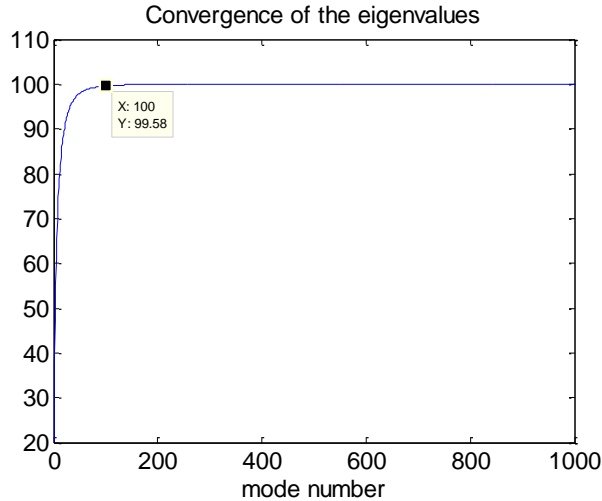
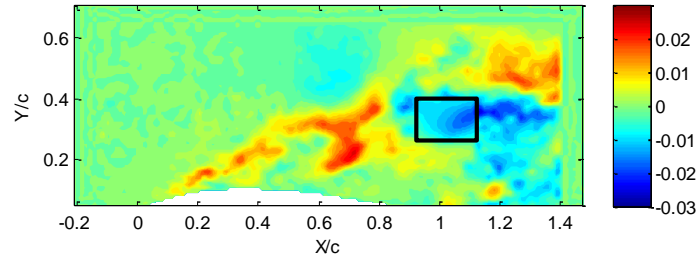
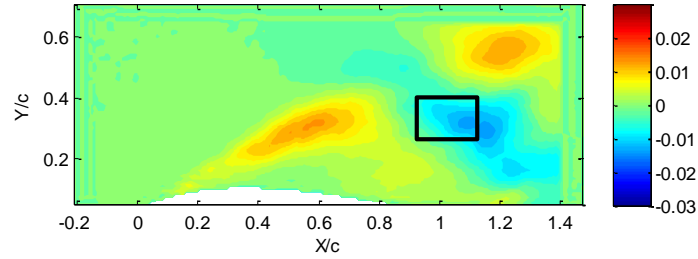


Figure 4.17: *POD Convergence Rate of a CFD simulated NACA 4412 Airfoil*

Looking at the flow field around the airfoil after performing data fusion (figure 4.18), it can be seen that the hole was not filled in perfectly; there is evidence of the hole in the estimated data based on the large gradients in the image. If fewer modes are used, the estimate begins to look correct because the gradients are smaller, meaning the fusion process is obtaining the low-dimensional characteristics of the flow (figure 4.18(b)). A plot of rms error distribution in time and space (only in the hole region) shows the fused data ($\sim 400,000$ points) matches the actual data better than the CFD (~ 800 points) (figure 4.19). The error distribution for the Fused POD results has a standard deviation of 25% while repairing the hole by injecting the CFD data into the hole has a standard deviation of 40%. Therefore, the fused solution is doing better than the CFD solution. It should be noted however, the CFD only had two snapshots that were comparable to the PIV due to the limited time support of the CFD data. This is actually another advantage of using fusion; even though the CFD only had two snapshots that are in the range of the PIV, Fused POD made it possible to repair all of the PIV snapshots, not just the matching ones; using the CFD alone only two snapshots would be repairable.



(a) Full Reconstruction



(b) 10 Mode Reconstruction

Figure 4.18: *Instantaneous Repaired Snapshot for a NACA 4412 Airfoil*

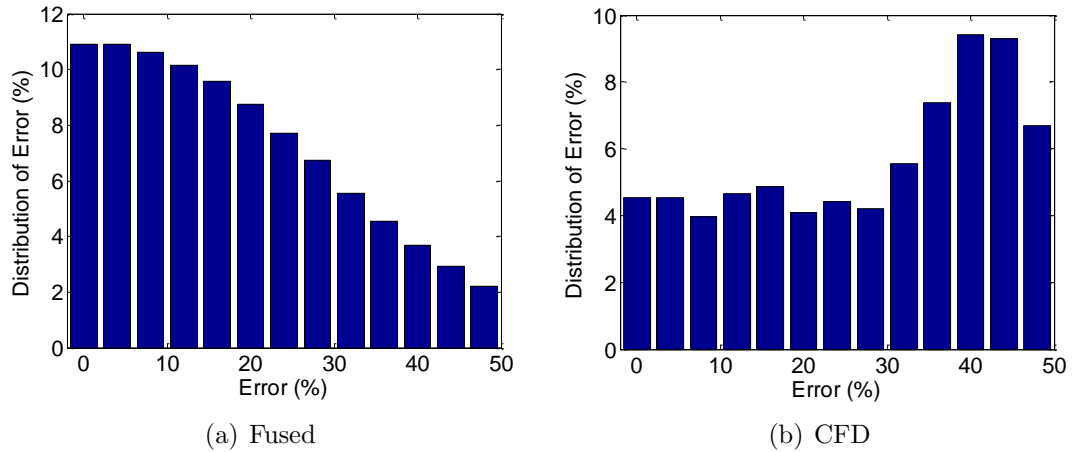


Figure 4.19: *Distribution of Error using Fused POD or CFD to Repair a Snapshot*

To check the convergence of this technique, the POD modes were calculated from the PIV data before the hole was introduced. Using these modes in the Fused POD technique produced the results in figure 4.20. One can see that as the number of modes increases, the Fused POD results converge to the actual data.

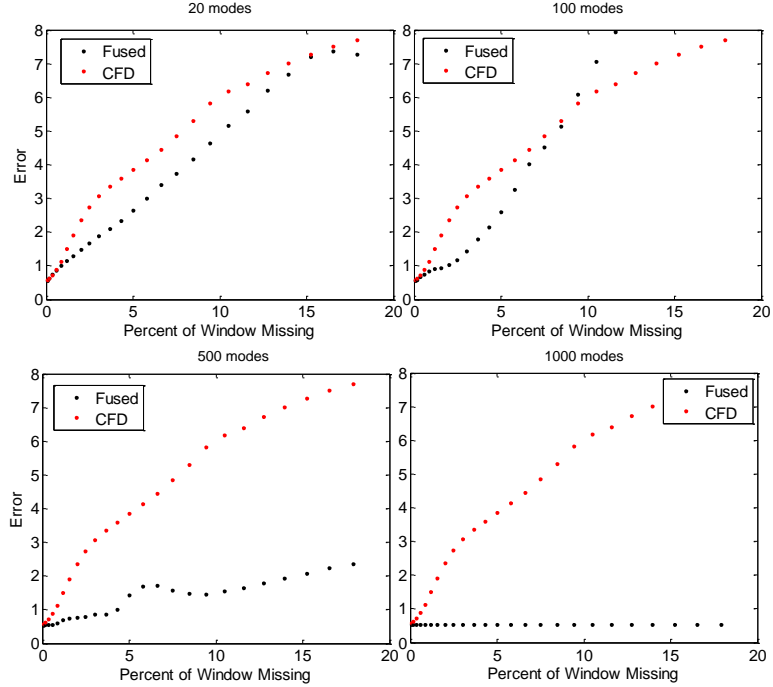


Figure 4.20: *Effect of Number of Modes on the Estimation of Missing Data for a NACA 4412 Airfoil using Fused POD with Exact Modes*

4.7 Temporal Estimation

The holes have been filled using Fused POD. Now the data needs to become time-resolved since the PIV data is sampled using a standard PIV system that samples at 4Hz, meaning there is a quarter of a second time step between each snapshot, which is not fast enough to obtain time-resolved information for the flow. Ultimately, a model based on low-dimensional velocity is desired as was done for this data by Ausseur [67]. For flow control purposes, the model should have time-resolved information to understand how the flow changes with time and as such, the velocity data used to build the model must be time-resolved. To obtain time-resolved velocity from the PIV data, one must use an estimation technique. Ausseur used sLSE to get an estimate of the time-resolved velocity. This is repeated in this work for validation purposes before applying the technique to the jet data. Another attempt to obtain time-

resolved velocity is performed using the CFD, which is time-resolved, to generate a stochastic model. The PIV data is then injected into the model to get a time-resolved estimate of the flow based on the PIV measurement.

4.7.1 Stochastic Estimation

When performing stochastic estimation, it is important to have a high correlation between the estimated value and the conditional signal. In the case of using pressure to estimate velocity, one can use a correlation plot to see the strength of the correlation over the entire spatial domain.

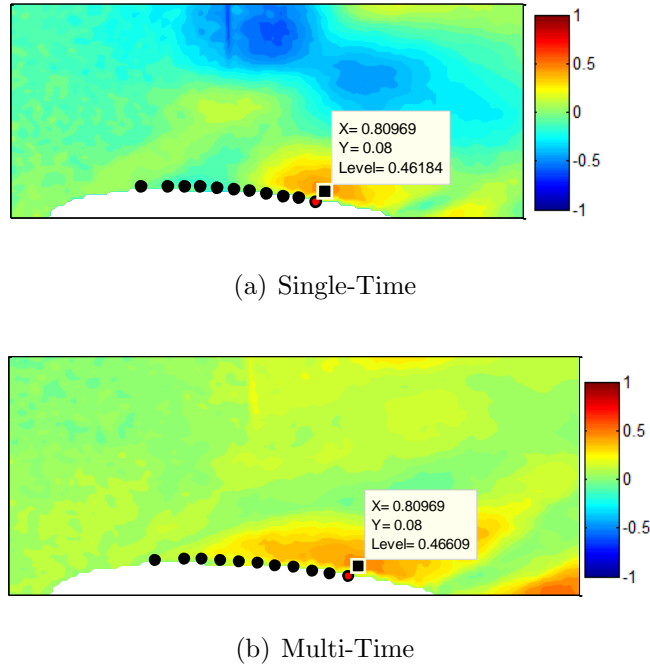


Figure 4.21: *Correlation between the Surface Pressure and the Velocity for a NACA 4412 Airfoil*

In figure 4.21(a) the single time correlation between pressure and velocity is displayed for the airfoil data. The correlation function is

$$\rho_{up_j}(x, \tau) = \frac{\langle u(x, t_o), p_j(t_o - \tau) \rangle}{\sqrt{\langle p_j(t_o)^2 \rangle \langle u(x, t_o)^2 \rangle}} \quad (4.25)$$

where u is the velocity, p is the pressure, τ is the time lag, and j is the pressure sensor number. In figure 4.21(a) the airfoil is shown in white, and the pressure sensors are displayed by the black dots. The single time correlation only compares pressure and velocity at a single lag (set τ to one value, one in this case). This plot shows the velocity is correlated well with the pressure locally. The correlation has a maximum value of 50% at the pressure sensor, and the correlation drops off from there. After about one length scale, the correlation drops to zero and there is no correlation between pressure and velocity. The correlation can be extended by using multiple time lags. This means the maximum correlation is taken instead of only focusing on a single lag (take the maximum of equation 4.25. As one can see in figure 4.21(b), the correlation levels have been increased further away from the pressure sensor.

By finding the optimal lag, which is done in sLSE, the extent of the high correlation region is larger than the high correlation region from the single time correlation, which is used in LSE. This implies sLSE predicts the flow better than LSE since the correlations are higher. This was shown for the airfoil data by Aussuer [67].

Spectral LSE is performed on the airfoil data. All of the sensors were correlated with the POD coefficients of the airfoil data to find the mapping terms for the sLSE. In the sLSE algorithm, the lags were allowed to vary between -128 and 128 time steps. Once the mapping coefficients are calculated, the pressure sensors are used to estimate the POD coefficients. These are then used with the POD modes to reconstruct a time-dependent velocity field.

Even though the sLSE can be performed on the airfoil data, it cannot be validated since the PIV is not time-resolved. However, some qualitative statements about the sLSE results can be made. First, the large scale structures appear to move in a manner that is expected as seen in figure 4.22.

Secondly, by tracking one structure in figure 4.22, we can see it moves 0.06m in 0.012s in the recirculation zone, which is a speed of about 5 m/s and as stated by Aussuer [67] the correct convection speed in the recirculation zone of a stalled airfoil

is $0.6u_\infty$ (6 m/s in this case). Since the large scale structures are propagating as expected, this technique seems to work, at least in a qualitative sense. The results of this estimation could be used in a controller because the dynamics seem correct. The sLSE is discussed in more depth in [subsection 5.5.1](#) for the jet flow.

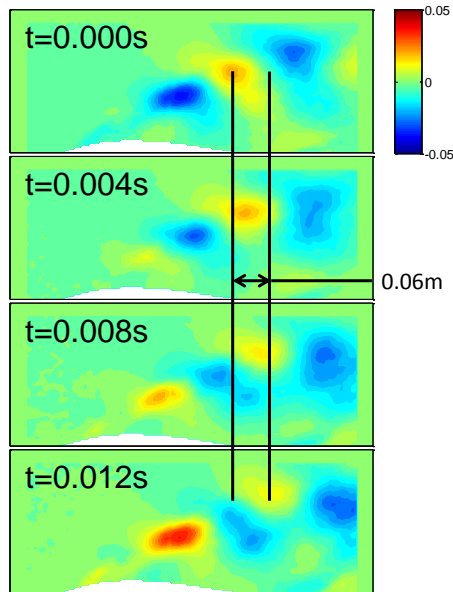


Figure 4.22: *Fluctuating Streamwise Velocity Estimated using sLSE*

4.7.2 Stochastic Modeling

Another method of estimating the time dependence using data fusion is stochastic modeling, where a model is generated from the CFD data and then the PIV data is injected into the model to give an estimate of how the PIV data should evolve in time. This work uses a linear model. A quadratic model was attempted, but it was unstable and diverged to infinity. The CFD data has time resolution, but the results may have modeling issues. The PIV data has the correct structures, but it is not time-resolved. Using the idea of fusion, the CFD time information is to be fused with the PIV spatial information. This works by using the CFD data to generate a stochastic model ([subsection 1.4.4](#)), which shows how the flow will transition between

time steps. POD is performed on the PIV data and the POD coefficients are used as initial conditions for the stochastic model. The idea is to model the flow until another PIV snapshot is available after 0.25s and then re-initialize the model using the next snapshot with a Kalman filter (subsection 1.4.5). The model results are shown in figure 4.23, which shows the estimated POD coefficient and the actual POD coefficient. Looking at the figure, one can see the model does not predict the correct value after 0.25s. The estimated coefficients at times between the two PIV snapshots cannot be compared with real data because it does not exist. Also, the model seems to lose amplitude as time increases, which most likely is not occurring in the actual flow.

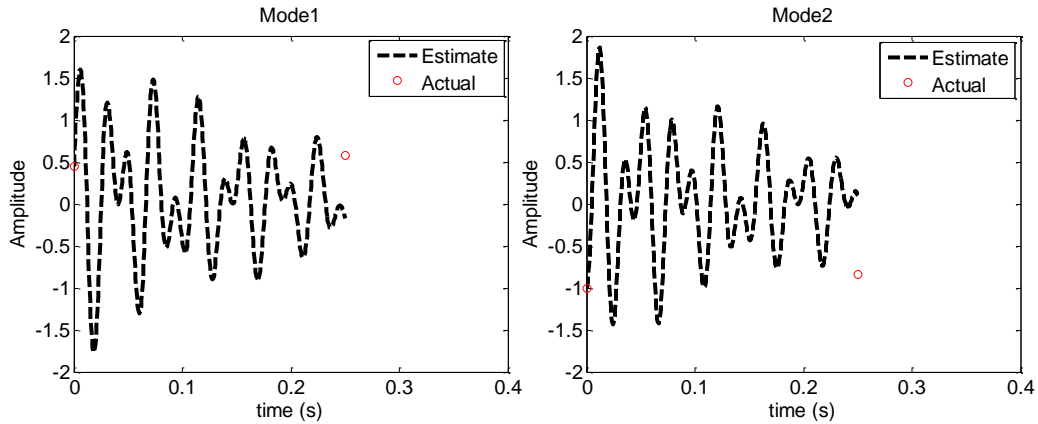


Figure 4.23: Comparison of the First Two POD Modes for the PIV Data and the Stochastic Model Estimate for a NACA 4412 Airfoil

As a check of this technique, the model is applied to the CFD data set and compared with the POD coefficients calculated directly from the CFD. In this case the CFD is still used to generate a stochastic model. However, the POD coefficients from the CFD data are used as an initial condition for the model. The data is then propagated using the stochastic model. Since the CFD data has time information, the estimate can be directly compared to the actual data. This shows the modeled results are matching the actual data for an integral time scale and then diverge from the actual data as seen in figure 4.24. A time scale is a length of time in which the

flow is correlated with itself. After one time scale, the flow is no longer correlated. A plot of the cumulative error between the estimate and the actual POD modes is shown in figure 4.25. This shows that the error increases after one integral time scale. Even though the time trace in figure 4.24 is not correct, the spectral domain of the signal is qualitatively similar.

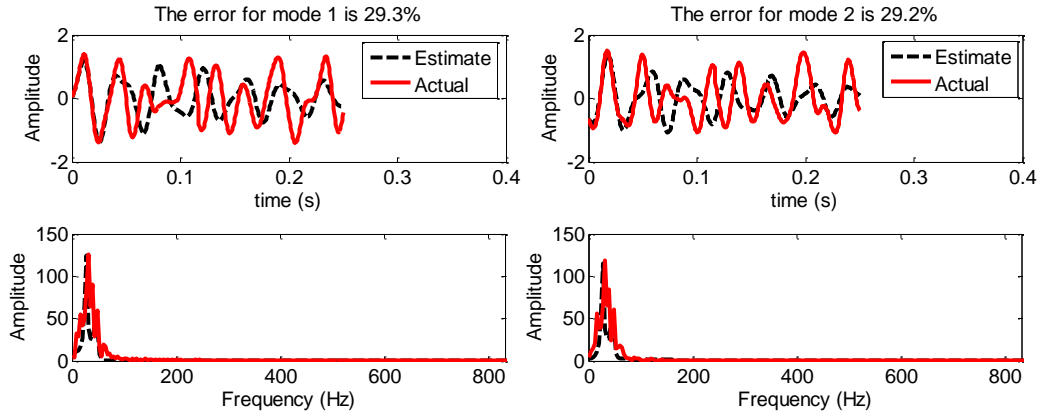


Figure 4.24: Comparison of the First Two POD Modes for the CFD Data and the Stochastic Model Estimate for a NACA 4412 Airfoil

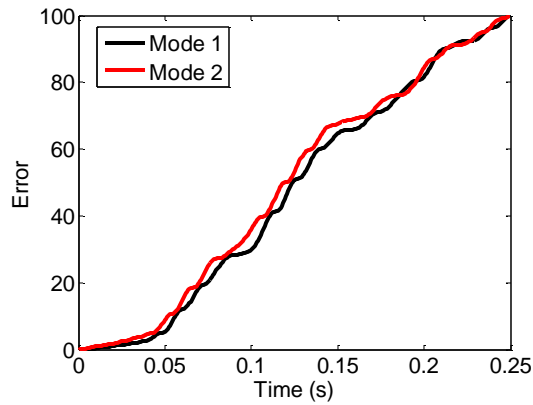


Figure 4.25: Cumulative Error Between the Estimated and Actual POD coefficients for Modes 1 and 2

Since the model seems to be predicting the POD coefficients with reasonable accuracy in the short time, a Kalman filter is applied by feeding the CFD POD

coefficients back every 0.03s which is about one time scale. This improves the long time behavior of the estimate as seen in figure 4.26.

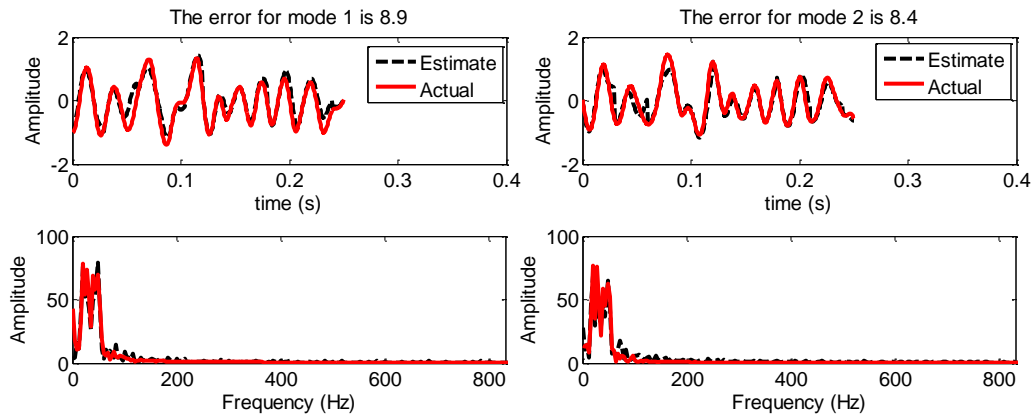


Figure 4.26: Comparison of the First Two POD Modes for the CFD Data and the Stochastic Model Estimate with a Kalman Filter for a NACA 4412 Airfoil

This seems like a viable option of predicting the time dependent velocity field as long as data is fed back into the system every time scale to correct the model. However, in the actual problem velocity is not available for feedback after every time scale, which potentially poses a problem for this technique. This result is the basis for the creation of DGP (to be discussed in subsection 5.5.4), which uses a model for “short” time behavior and actual data to correct the “long” time behavior. The technique is not applied here because it was developed to fuse multiple PIV windows together as will be discussed in the next chapter.

4.8 Summary of Data Fusion on a Flow Over an Airfoil

Data fusion is applied to a NACA 4412 airfoil at an angle of attack of 16° . The conditions were such that the airfoil was stalled and a recirculation region formed on the suction surface of the airfoil. The data used in this test included PIV, CFD,

and surface pressure. Techniques such as Fused POD, sLSE, and stochastic modeling were applied to the data to test the effects of data fusion for fluid dynamic data, test the abilities of the various techniques, and get inspiration for new techniques.

Holes were cut in the PIV data, which were repaired using Fused POD and the CFD data. It was found that the CFD did not match the data as well as expected. Despite that, the holes were repaired well enough to get an idea of how the flow was behaving in the missing region. Using the exact basis, one could repair missing data exactly by using all of the modes. At some point when the modes differ too much, the repaired data does not improve. This shows that one must be careful to use modes that match the data correctly.

An estimate of time-resolved velocity was generated using sLSE and stochastic modeling. The sLSE showed the anticipated behavior based upon the propagation of flow structures, but data was not available for direct comparison. The same is true for the stochastic modeling technique. However the model generated from the CFD data was injected with CFD data and the corresponding estimate compared well for one time scale. The addition of a Kalman filter improved the accuracy of the technique, but this would not be applicable since standard PIV does not have a small enough time step to feedback velocity data after every integral time scale. This result is the inspiration that lead to the development of DGP.

The airfoil work is meant to be another stepping stone and test bed for some of the techniques that are applied to the jet data. This work showed that Fused POD could produce reasonable results when filling in missing data, sLSE produces results that could potentially be correct for the airfoil, and stochastic modeling works well in the short time scales, but more information is needed to make sure the long time is captured correctly.

Chapter 5

Fusion of Mach 0.6 Jet Data

In the previous chapters, fusion, the methods of fusion, and the application of fusion to fluid dynamic data have been discussed. Now that this has been accomplished, fusion can be applied to data obtained from a Mach 0.6 jet flow. However, before fusion is applied to the jet data it is important to understand the jet flow and past work that has been done in this field. One of the largest issues with the jet is the noise it produces. Jet noise has been an issue for decades and is a problem in both military and commercial applications [101]. On the commercial side, airports were originally built outside of residential areas. However, residential areas have begun to encroach upon airports [124] causing an increased concern about jet noise, especially for the people living near airports [125, 126]. To counteract this, the goal is to decrease aircraft noise by 6–9 decibels by the year 2017 according to Viswanathan and Pilon [127].

Jet noise is a military concern as well. The obvious reasons for this are tactical purposes. Military aircraft should be as quiet as possible so they can approach a target without detection. However, there are other reasons which lead to a desired decrease in the sound levels of aircraft. One reason, specific to the Navy, is the large number of aircraft carrier crew members reporting hearing damage [128]. According to Trost and Shaw [128] military personnel serving on an aircraft carrier are 13% more likely to report hearing damage in some capacity than a person not stationed

on an aircraft carrier.

5.1 Jet Flow

The flow field produced by a jet is a phenomena that has been studied by many different researchers over the years both experimentally [30, 60, 99, 101, 129–140] and numerically [141–147]. Jet research has focused on determining the structure of jet flows [30, 131, 132, 134], sound generation [60, 129, 130, 133], sound source identification [99, 101, 142], and flow control to reduce noise [135–139].

5.1.1 Jet Flow Structure

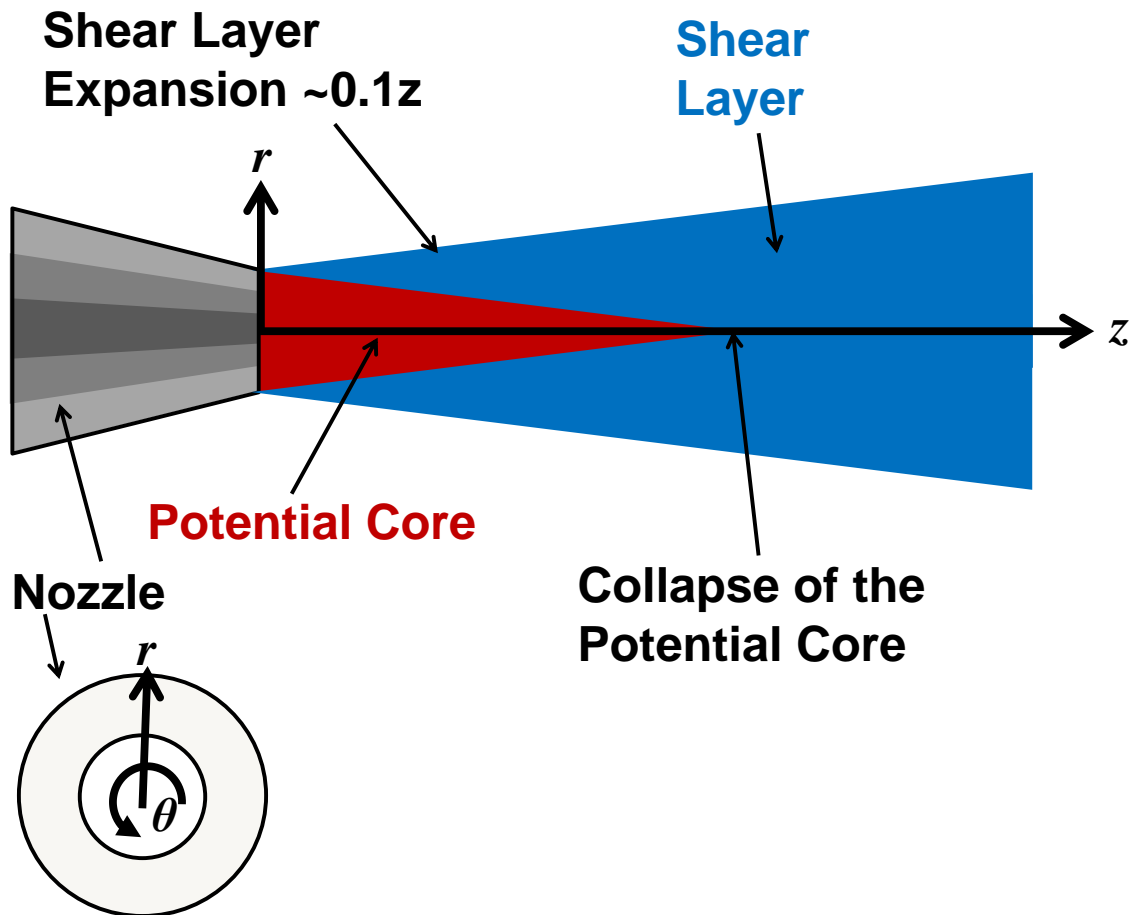


Figure 5.1: Schematic of an Axisymmetric Jet Flow Field

A jet is a complicated flow that includes many features including shear layers, potential flow, and coherent structures (see figure 5.1 for a schematic of a typical jet flow). The potential core is part of the jet that is assumed to be potential flow as the name suggests. This means the flow is irrotational ($\nabla \times \vec{U} = 0$) and the flow remains unmixed. Another feature of the jet is the shear layer, which is caused by the large difference (shear) in velocity between the potential core and the ambient air. The difference in velocities cause mixing and the formation of vortices and coherent structures in the shear layer. As the flow moves downstream the structures grow causing the shear layer to expand. The shear layer grows approximately as $0.1z$ [148]. Eventually, the shear layer becomes large enough that it overtakes the potential core. The point at which this occurs is called the collapse of the potential core.

The standard definition of the coordinate system for an axisymmetric jet is (r, θ, z) where r is the radial direction coming out from the center of the jet. θ is the azimuthal direction, and z is the streamwise direction. This definition is displayed in figure 5.1. Now that the important features of a jet have been described, a discussion of some notable work done to understand these features is presented. Crow and Champagne noted that there had been evidence for coherent structures in a jet flow from previous work, but their work was the first to show these structures [131]. In their work, they were able to see turbulent structures that propagate downstream with a Strouhal number of 0.3, which was independent of Reynolds number. Work done by Brown and Roshko [132] looked at coherent structures in the mixing layer and found structures grow as they propagate downstream. The structures grow by merging with neighboring structures. Further work was done by Glauser *et al.* to identify large-scale structures in a jet shear layer [134]. Using POD (discussed in subsection 1.4.2) Glauser showed most of the energy in the shear layer is contained in the largest structures. This shows that the large structures are the dominant feature in the jet shear layer. This was studied further by Citriniti and George where they showed an azimuthal structure exists near the collapse of the potential

core using a low-dimensional model [149]. In between these structures exists a pair of counter-rotating streamwise vortex structures. It was shown that the azimuthal structure causes fluid to be ejected in the streamwise direction, which is thought to be a source of jet noise. Work done by Tinney *et al.* [30, 57] also showed this explosion phenomenon by creating a 3D time-resolved estimate of the flow field using sLSE (discussed in subsection 1.4.3). Tinney *et al.* also observed that flow structures did not change with Mach number nor Reynolds number for a subsonic high Reynolds number jet.

5.1.2 Aeroacoustics

One of the main reasons to study the jet flow field is to determine how the fluid dynamics create noise. One of the first researchers to try to understand the connection between fluid dynamics and noise generation was Lighthill. Lighthill discusses how sound is generated by aerodynamics [129]. To do this, Lighthill created an acoustic analogy by rearranging the mass and momentum balance equations and combining them in such a way that looks similar to a wave equation.

$$\frac{\partial^2 \rho}{\partial t^2} - c_o^2 \frac{\partial^2 \rho}{\partial x_i^2} = \frac{\partial^2}{\partial x_j \partial x_i} (\rho u_i u_j + P_{ij} - c_o^2 \rho \delta_{ij}) \quad (5.1)$$

where P_{ij} is the physical stress, $u_i u_j$ is the Reynolds stress, and $c_o^2 \rho \delta_{ij}$ is the stress in a stationary uniform acoustic medium.

Further work was done by Ffowcs-Williams and Hawkings in which Lighthill's acoustic analogy was modified for use with rotating machinery [130]. Before this, the acoustics were solved using the homogenous wave equation ($\frac{\partial^2 p}{\partial t^2} - a^2 \nabla^2 p = 0$) and the pressure on the surface of the blade as a boundary condition. Ffowcs-Williams points out that in order to apply the homogenous wave equation, the fluid speed must be significantly less than the speed of sound. Usually, the flow speed is on the same order as the speed of sound. They solved this problem by using Lighthill's equation

above the blade surface with the addition of source terms introduced by the flow surface interaction. The surface terms are monopole terms and dipole terms where the Lighthill source term is a quadrupole term. Using the Ffowcs-Williams-Hawkings method, the volume integral required by Lighthill is no longer required, instead a surface integral can be performed, resulting in less computational work.

Computational Aeroacoustics

Recently, researchers have been attempting to simulate the jet flow field and calculate the noise produced by the flow. There are three main ways to accomplish this, as discussed by Colonius *et al.* [141]. One is to simulate the noise production directly. This would involve solving the Navier-Stokes equations extending to the far-field, which is computationally expensive. A second method is to solve only the near field region and use an acoustic analogy discussed previously to estimate the far-field acoustics. This can cause a problem because any errors in the flow field can act as source terms. These terms can be larger than the actual source terms and cause errors in the sound propagation. A third method is a hybrid method. This simulates the near field and a small portion of the acoustic region, which again can be computationally expensive. Colonius *et al.* showed in their work that the acoustic field produced by a direct numerical simulation (DNS) and directly calculating noise was similar to using an acoustic analogy for the same DNS data.

Bodony and Lele discuss the work done by computational aeroacousticians and the methods used to simulate the flow field [150]. They discuss the various methods to propagate the sound and show researchers have used methods that range in complexity from Ffowcs-Williams-Hawkings to a scaling factor that is dependent on the distance from a source. Bodony and Lele also discuss the use of turbulence modeling and shows that the models used are usually a dynamic or constant Smagorinsky model. However, many researchers have also used a no-model approach in which the sub-grid scales are ignored because modeling can cause more errors than not modeling

the small-scale turbulence. Bodony and Lele also state that the boundary conditions in computational aeroacoustics are a critical factor in producing correct simulation results.

Freund simulated a low Reynolds number, subsonic jet directly and used the Lighthill analogy to calculate the far-field noise. This is compared with the direct method and experimental data [142]. All three showed similar results. Freund also filters out the part of the source terms that are radiated to the far-field and shows the filtered source has peaks at different locations than the unfiltered sources. The filtered noise terms seemed to be similar to the source terms in a laminar mixing flow.

Bodony and Lele also looked at separating the source terms [146] and showed the momentum term was the dominant term in noise propagation in the flow direction of the jet. They also show for a cold jet the sound propagation is dominated by momentum even in shallow angles. However, for a hot jet this was not the case.

One should note, this is not a comprehensive review of jet flow and jet noise. There has been many more studies done on these topics. This section is meant to familiarize the reader with jets and jet noise since jet data is used to test the data fusion techniques. The real focus of this is the application of the data fusion techniques on jet data.

5.2 Experimental Setup

The data used in this study was collected in the Syracuse University anechoic chamber at the Skytop facilities seen in figure 5.2. The dimensions of the chamber are 7.9m x 6.1m x 4.3m. The chamber is treated with fiberglass wedges to absorb any sound that is echoing above 150Hz and allows for the study of the noise being produced by the jet.

The jet is a blow down facility, meaning the jet works by pressurizing a reservoir (not shown in figure 5.2) and then releasing air from it to produce the flow. A valve

is used to control the back pressure (air leaving the tanks) so the Mach number of the jet remains constant despite the decreasing stagnation pressure in the tank. The nozzle is a converging nozzle with an interior profile that uses a matched 5th order polynomial as stated by Tinney *et al.* [151]. The nozzle has a contraction ratio of 3:1 with a two inch diameter at the exit. The jet facility is capable of running between Mach 0.3 and Mach 1.1 and temperatures ranging from approximately room temperature ($\sim 70^\circ\text{F}$) to $\sim 1000^\circ\text{F}$. The current work uses a jet at Mach 0.6 and a temperature of $\sim 70^\circ\text{F}$.

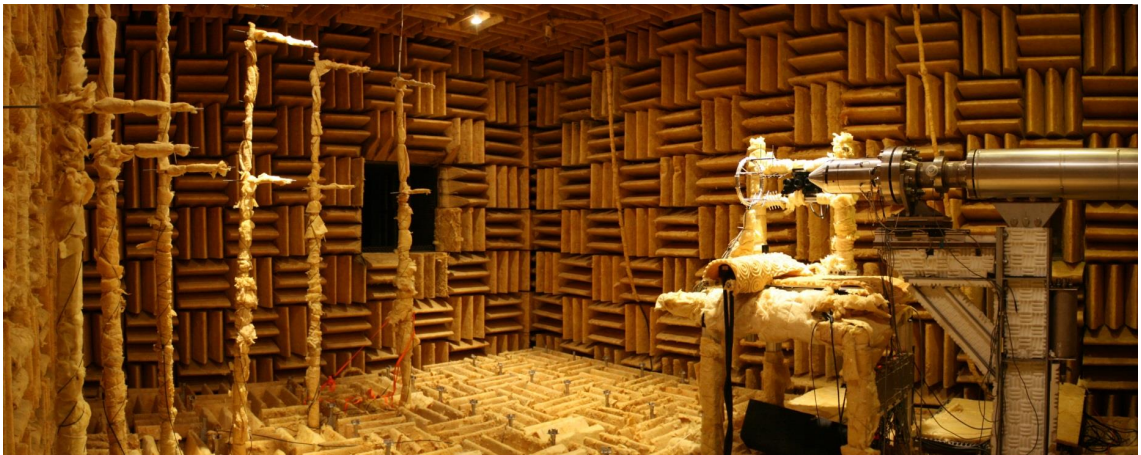


Figure 5.2: *Syracuse University Anechoic Chamber at the Skytop Facility*

Two sets of experiments done in this facility are used for the data fusion work. One set of experiments were performed by Berger *et al.* [107] and involves using far-field microphones, near-field pressure sensors, and large-window PIV (LWPIV). The other set were performed by Low *et al.* [99] and includes far-field microphone, near-field pressure sensors and time-resolved PIV (TRPIV) data.

5.2.1 Large Window PIV

The LWPIV was acquired using three 8-bit, HiSense cameras and a New Wave Gemini Nd:YAG laser. The setup of this experiment is seen in figure 5.3, which shows the jet in the middle of the image, the three cameras on the top left, and the PIV laser on the

bottom right. Three cameras were used in this study to increase the spatial domain without sacrificing spatial resolution, which produced three separate images that are stitched together as stated earlier in [section 1.3](#). The use of multiple cameras and the stitching algorithm [25] allows for a window that has 7 diameters in the streamwise direction and 2 diameters in the radial direction.



Figure 5.3: *2013 LWPIV Experimental Setup for the Jet at the Syracuse University Anechoic Chamber at the Skytop Facility*

In addition to the LWPIV data, near-field pressure was acquired using 13 Kulite transducers as seen in [figure 5.4](#). Five sensors were arranged in an azimuthal array at $z/D = 6$, another five were arranged in an azimuthal array at $z/D = 8$, and five were arranged in a linear array at $z/D = 6, 6.5, 7, 7.5, \text{ and } 8$. The sensors at $z/D = 6$ and $z/D = 8$ were in both the linear array and azimuthal array. This means that even though 15 sensors were mentioned, two are actually duplicates. The sensors are placed outside the shear layer first by using the $0.1z$ expansion and then the placement is fine tuned by ensuring the RMS of the pressure signals are the same, due to the axisymmetric nature of the jet.

Far-field noise was also acquired in this experiment using G.R.A.S. type microphones. The far-field noise is not used in the data fusion work, but the ultimate goal is to reduce the noise levels of the jet. The microphones are used to determine the sound levels and how the flow is generating noise. These are seen in [figure 5.3](#) on the left side of the image.

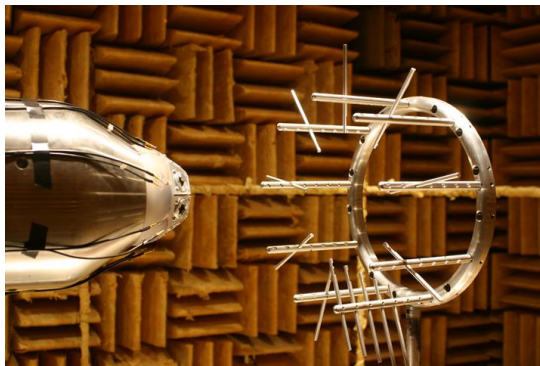


Figure 5.4: *Kulite Array Configuration for the Jet at the Syracuse University Anechoic Chamber at the Skytop Facility*

5.2.2 Time-Resolved PIV

The TRPIV was acquired using a Quantronix Hawk-Duo series PIV laser capable of collecting data with a sampling rate of 10kHz. The camera used to capture the data was a Photron FastCam CCD camera. Only one camera was used in this experiment and as a result a smaller domain (2 diameters x 2 diameters) was captured. The camera was limited to 0.86 seconds (8623 snapshots) worth of velocity data due to the size of the buffer in the camera (data the camera can hold before having to download to a computer). The experimental set up for the TRPIV experiments is seen in figure 5.5.

The availability of the one camera limited the size of the PIV window as previously stated, but a large portion of the jet in the streamwise direction is still desired. To compensate for this, seven windows of data were acquired at different downstream locations. Each window was overlapped with the previous window by 0.5 diameters (~ 1 length scale). Each window was taken at different points in time so each window is time-resolved but not synchronized with one another.

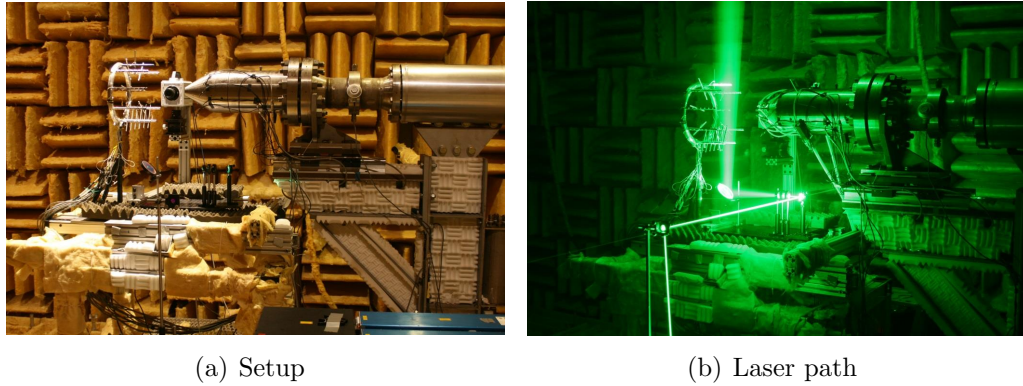


Figure 5.5: *2011 Time-Resolved PIV Experimental Setup at the Syracuse University Anechoic Chamber at the Skytop Facility*

In addition to the PIV measurements, near-field, and far-field pressure were acquired similar to the LWPIV experiment.

In this test, the near-field pressure was acquired using 13 Kulite transducers. However, the placement of the sensors for the TRPIV experiment is different from the placement for the LWPIV experiment. Nine of the sensors were arranged in an azimuthal array at $z/D = 6$ and five were arranged in a linear array at $z/D = 4, 5, 6, 7, 8$. The sensor in the linear array at $z/D = 6$ is also included in the azimuthal array where the two arrays overlap.

The data collected for the jet is not what is ideally desired. Despite the richness of these data sets, there are deficiencies caused by limitations of the measurement techniques. There are holes in the data, large-window data that is not time-resolved, and time-resolved data that is acquired over a “small” spatial domain. Still, much information is obtained from these data sets, which is evident by the works of Low *et al.* [99], Berger *et al.* [100], Kan *et al.* [152], Lewalle *et al.* [153], and Shea *et al.* [25] but using data fusion can allow for an improved data set that can potentially lead to a better understanding of the physics.

5.3 Summary of Data Fusion for the Jet

In the following sections, the techniques used to repair and improve the jet data are discussed. A summary of this is shown in figure 5.6, where FP is Fused POD, sLSE is spectral linear stochastic estimation, GP is Gappy POD, and DGP is Dynamic Gappy POD. The diagram shows how the three different data types were fused together and the techniques used to accomplish the fusion.

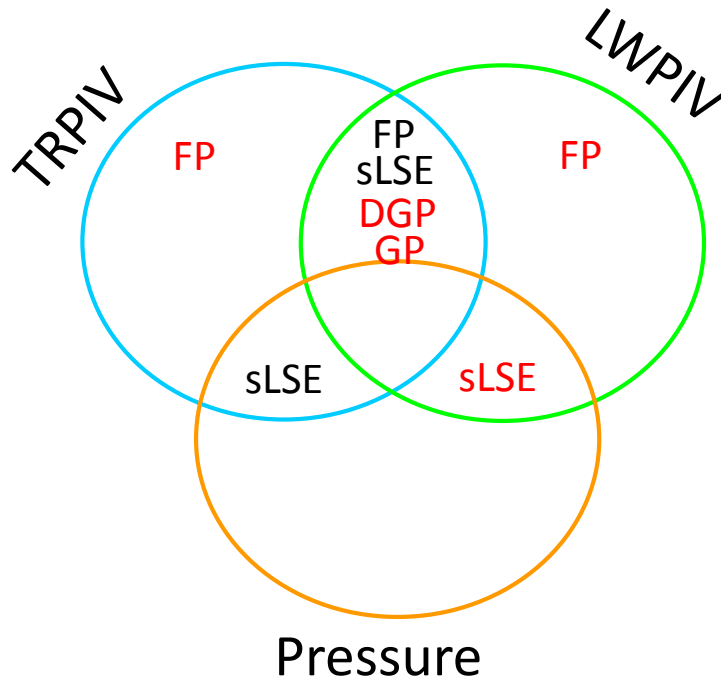


Figure 5.6: *Data Fusion and Technique Fusion used on the Mach 0.6 Jet Data*

The techniques highlighted in red are discussed in the sections to follow. They focus on Fused POD to fill in holes in the LWPIV and the TRPIV. DGP and Gappy POD are used to estimate a large-window time-resolved data set using the LWPIV and TRPIV. Spectral LSE is used to estimate a large-window time-resolved data set with the near field pressure and TRPIV data. The techniques in black were attempted, but the results are omitted because they did not work as well as the ones to be presented in this chapter. These include using Fused POD to fill holes in the

LWPIV based on the modes from the TRPIV data set, using velocity from the time-resolved window to estimate a large-window time-resolved data set, and using sLSE to estimate a large-window time-resolved data set with the near-field pressure and LWPIV data. Most of the tests cases that are omitted from this chapter were unable to be verified and could not show how well the techniques work, since data was not available for comparison.

A few deficiencies are present in the data sets, many of which are discussed in [section 1.3](#). The specific deficiencies in the data from these experiments include: sensor occlusions, reflections, particle drop out, small window size, and “slow” sampling rate.

5.4 Hole Filling

Let us first discuss what caused the sensor occlusions. The sensor occlusions are caused by one of the Kulites being between the camera and the laser sheet. This means the camera takes an image of the Kulite in every frame, effectively producing an occlusion or region of missing data in that area. [Figure 5.7](#) shows an instantaneous image of streamwise velocity captured by the LWPIV system. In the image the sensor occlusions are circled. These spots need to be masked out so the analyses are not corrupted by the incorrect data.

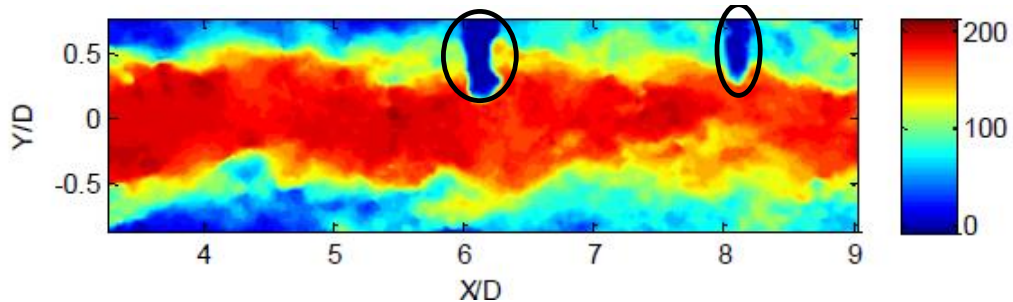


Figure 5.7: *Sensor Occlusions in the LWPIV data of a Mach 0.6 Jet*

Despite the appearance of the data, the sensors are actually not in the flow field.

They are outside the jet at a different azimuthal angle than the PIV, so the flow was not disturbed by the sensors.

Another issue is reflection. This is caused by the laser, as the name suggests, reflecting off an object and creating a bright spot in the PIV images. This error is not always seen in instantaneous PIV snapshots. One sometimes has to compute the mean or higher order statistics to see the reflection. An example of a reflection is seen in figure 5.8. This image shows the mean streamwise velocity for a Mach 0.6 jet. The reflections are located in the circled areas in figure 5.8. Reflections are usually hidden in the instantaneous velocity (figure 5.7). The reflections also need to be masked out to prevent corruption in future analyses.

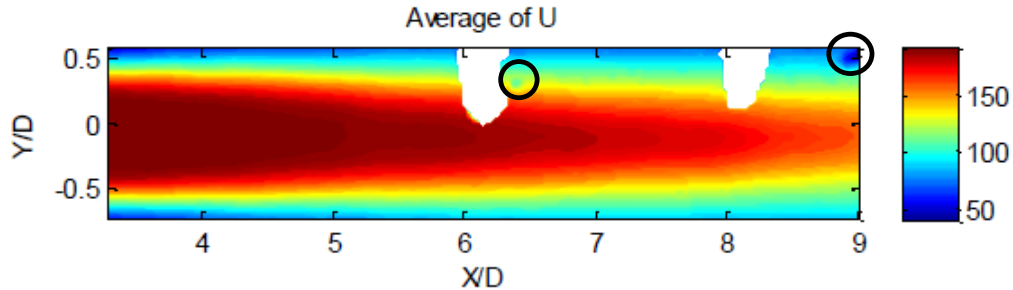


Figure 5.8: *Reflections in the LWPIV Data for a Mach 0.6 Jet*

5.4.1 Sensor Occlusion Estimation

Ideally, one would like to sample pressure and velocity simultaneously for many reasons that include correlating velocity and pressure, using pressure to help understand the time dependence of the flow, creating models for velocity based upon pressure (because pressure is easily measured in real world applications), etc. However, to get pressure and velocity simultaneously for the jet, sensor occlusions and reflections are inevitable without very complex laser and camera placements [154]. These occlusions can cause issues even if they are masked. For example, the POD modes can change if parts of the flow are masked, as shown in figure 5.9, where the mask causes mode 1 and mode 2 to swap as seen in figure 5.9. However, the energies of the two modes

are only different by about a percent, but still something to be accounted for.

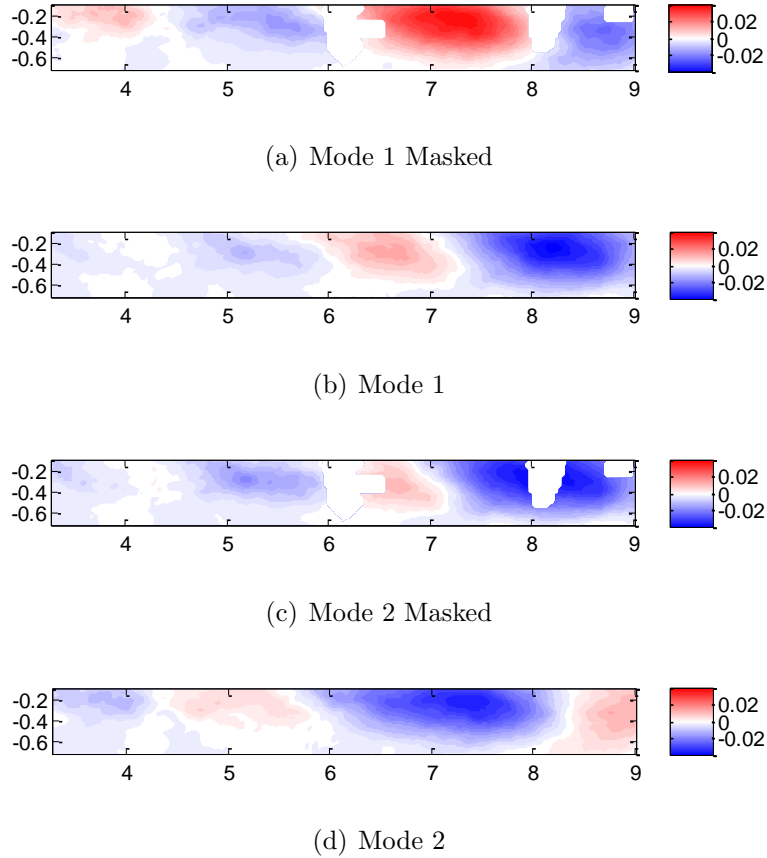


Figure 5.9: *Comparison of POD Modes Calculated With and Without a Mask*

Since the jet is axisymmetric, the top and bottom half of the jet should be mirror images of one another in a statistical sense (see figure 5.11 and the explanation of said figure). In the instantaneous velocity, the flow fields do not look alike, which is why the repair process depends on the POD modes. The fact that the jet is axisymmetric is fused with the PIV data using Fused POD.

The fusion process starts by splitting the jet data in half along the centerline of the jet and flipping the data from the bottom half. The flipping process involves spatially flipping the data and negating the sign of the cross-stream velocity component (radial direction). This should create two data sets with very similar statistics. The flipped bottom half data is then used to compute the POD modes. These modes should also

represent the top half of the data since the jet is symmetric. The modes are then used in the Fused POD algorithm by fitting the top data to the bottom modes. The effects of where to split the data are discussed as well as how sensitive the process is to the location of the split. The number of snapshots needed to converge is also discussed as well as the use of wavelet fusion on the data. The fusion process use 500 modes unless stated otherwise, since 500 modes is where the energy convergence plot levels out. This was shown to be the optimal number of modes to use in the synthetic data set.

Location of the Jet's Axis of Symmetry

The location of the split should be along the centerline of the jet, which is the line of symmetry in the data. Three different cutting methods are discussed. One method cuts the data along the center of the domain. Another technique cuts the data at the center of potential core at the beginning of the domain. The final method cuts the data along the center of the potential core. The center of the potential core in this work was defined by the maximum mean streamwise velocity at different streamwise locations. This method could be effected slightly by noise, a more advanced method of finding the potential core could be used in the future. The different splitting methods are defined in figure [5.10](#).

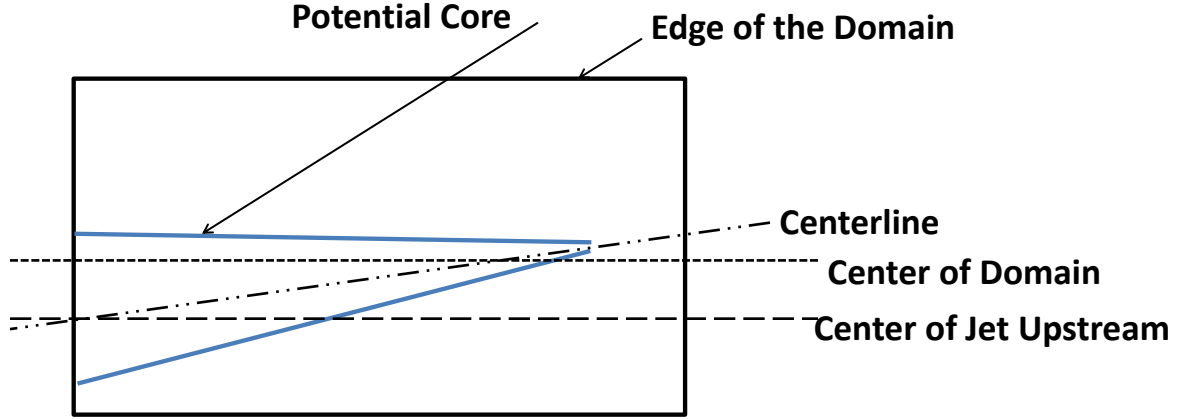


Figure 5.10: *Three Ways of Splitting the Jet Data into Symmetric Top and Bottom Halves*

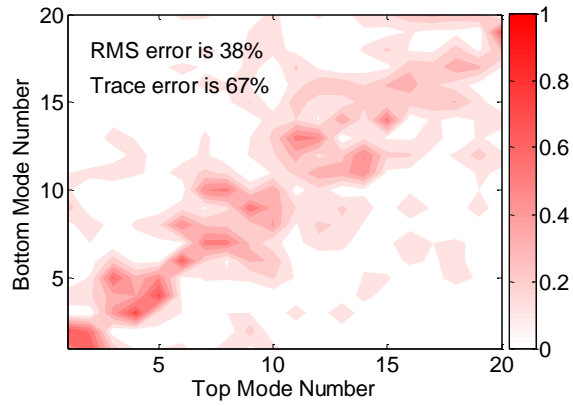
After the data is split and flipped, the POD modes from the top and bottom planes are compared. They are compared by correlating each mode from the top with each mode on the bottom. If the top and bottom are perfectly symmetric the correlation matrix is an identity matrix. The correlation matrices are plotted in figure 5.11. The darker color implies a higher correlation. The correlations in figure 5.11 show that splitting the jet along the centerline is better than splitting at the center, which is better than splitting along the center of the domain. This is quantified using two different methods, one compares the RMS error of the correlation plot versus an identity matrix

$$\sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^n (r_{ij} - d_{ij})^2}{n^2}} * 100\% \quad (5.2)$$

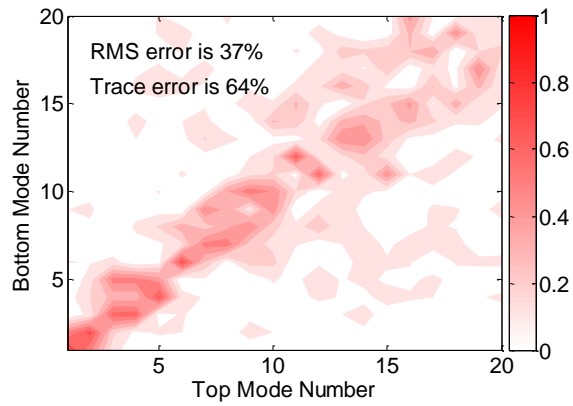
where n is the number of modes being compared, d_{ij} is distance from the diagonal, and r_{ij} is the correlation. The second method compares the trace of the correlation matrix

$$\left(1 - \frac{Tr(r)}{n}\right) * 100\% \quad (5.3)$$

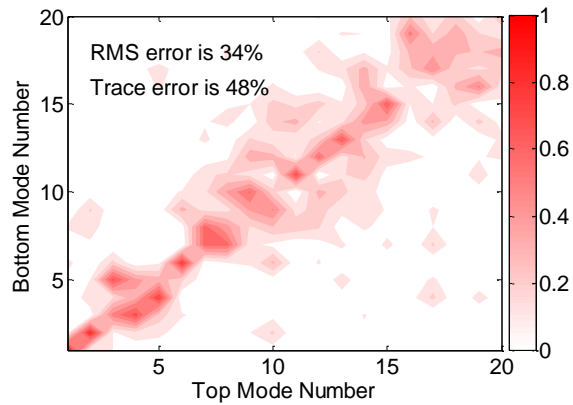
where the trace is the sum of the diagonal elements, and n is the number of modes used in the correlation.



(a) Data Split Along the Center of the domain



(b) Data Split Along the Center of the Jet



(c) Data Split Along the Centerline of the Jet

Figure 5.11: *Correlation Plots Between POD Modes from Top and Bottom of an Axisymmetric Jet Using Different Splitting Methods*

Using these metrics, splitting along the centerline has a trace error of 48% and an RMS error of 34%, splitting along the center has a trace error of 64% and an RMS error of 37% and, splitting along the center of the domain has a trace error of 67% and an RMS error of 38%. This further demonstrates the symmetry of the jet and the importance to split the jet along the actual line of symmetry to get similar modes from the top and bottom of the symmetric jet. If the top and bottom modes do not match, the estimation does not produce results that correctly fill the hole in the higher order statistics.

To validate Fused POD, estimated data was compared with actual data. This was accomplished by introducing an artificial hole with an area that is 1 length scale x 1 length scale, 250 pixels and for 500 snapshots that was treated as if it was another occlusion. The data in the artificial occlusion is estimated using Fused POD and compared with the actual data using equation 5.4

$$e_{ijt} = \frac{(U_{f_{ij}(t)} - U_{o_{ij}(t)})}{(U_{max} - U_{min})} 100\% \quad (5.4)$$

where $U_{f_{ij}(t)}$ is the estimated velocity, $U_{o_{ij}(t)}$ is the actual velocity, U_{max} is the maximum velocity and U_{min} is the minimum velocity.

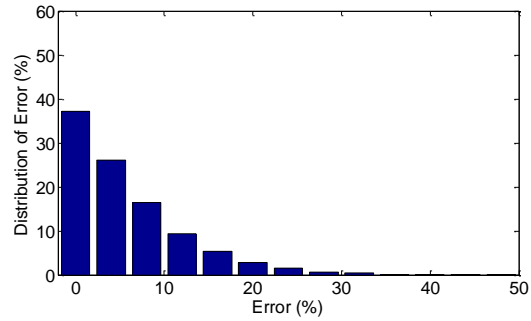
The error at each point in time and space was then placed into bins to create the error distribution chart in figure 5.12. The distribution shows the majority of the error is contained between 0 – 3%. The distribution of error when splitting the jet along the centerline is more concentrated near zero error ($\sigma = 2.5$) and has a maximum error of about 20%. The error distribution for cutting the jet across the center of the jet at the beginning of the domain has a slightly larger distribution ($\sigma = 4.5$) with a maximum error of about 23%. The distribution for cutting along the center of the domain has a even fatter distribution ($\sigma = 5.0$) and a maximum error of about 25%. This implies that splitting the jet along the centerline is a better approach than the other two methods in terms of error distribution. This further supports the findings

from the correlation plot that splitting the jet along the centerline is the best method of finding the modes used to reconstruct the “bad” data.

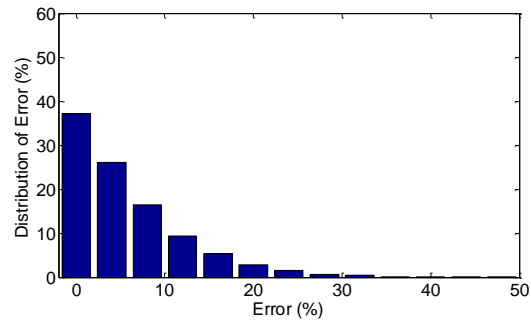
After performing the Fused POD technique on the data, the occlusions from the sensors are no longer seen in the instantaneous snapshots. This is demonstrated in the figure 5.13, which shows an instantaneous snapshot of streamwise velocity for the three different splitting methods. This demonstrated that in the instantaneous sense, the Fused POD is doing a very good job estimating the velocity in the occlusion region. The error is quantified using a distribution of RMS error in figure 5.12 as discussed previously.

Despite the success in the instantaneous sense, Fused POD is not predicting the flow in the statistical sense correctly. This is seen in figure 5.14. In figure 5.14 the RMS of the streamwise velocity is displayed. One can readily see the sensors in these images. The extent at which the sensors stick out is decreased as the splitting method gets more complex. The best splitting method for recovering the data in the occluded region in terms of RMS error is the centerline split as was shown in figure 5.14 and figure 5.12.

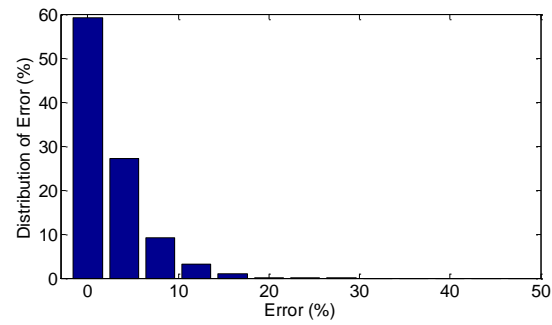
This further supports what is found from the modal correlation plot (figure 5.11) and the distribution of error (figure 5.12); finding the centerline of the jet is important when using this technique. As such, the rest of this work uses the centerline to cut the data.



(a) Split Along the Center of the domain

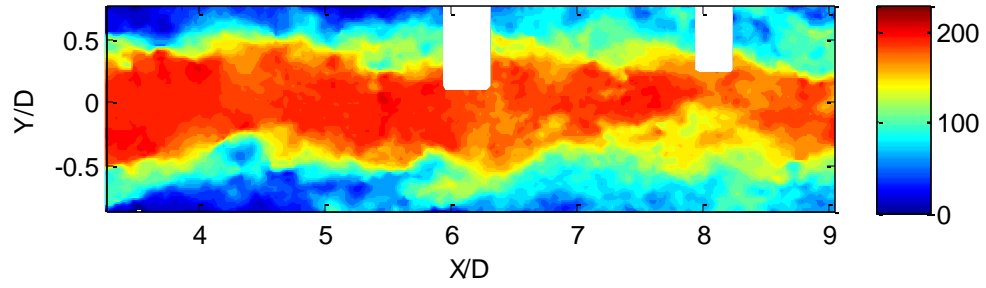


(b) Split Along the Center of the Jet

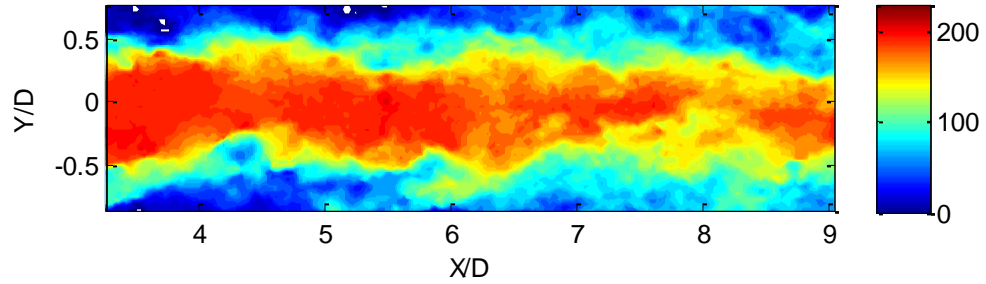


(c) Split Along the Centerline of the Jet

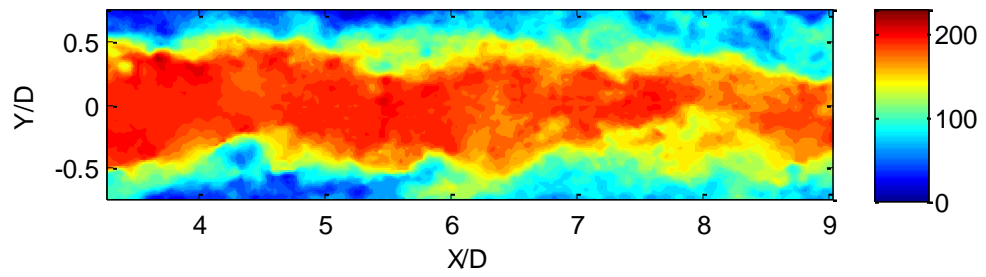
Figure 5.12: *Error Distribution for Fused POD using Different Splitting Methods in an Axisymmetric Jet*



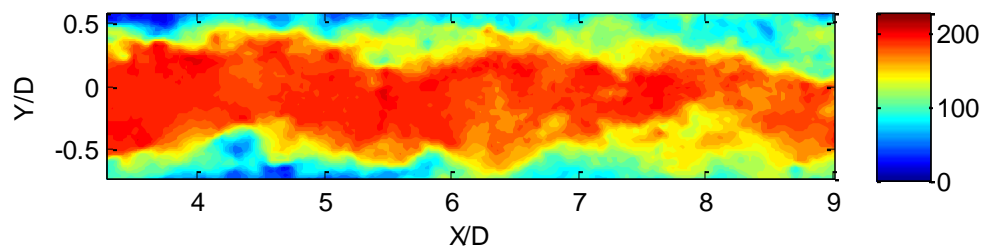
(a) Original Data



(b) Split Along the Center of the Domain

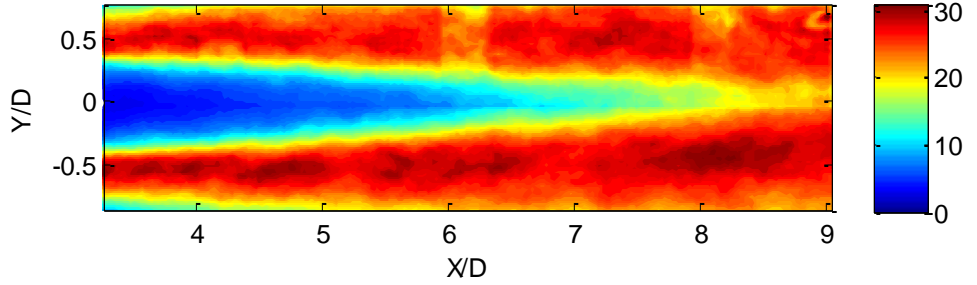


(c) Split Along the Center of the Jet

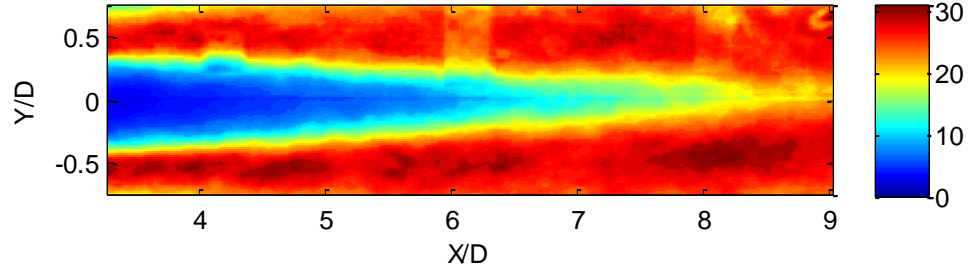


(d) Split Along the Centerline of the Jet

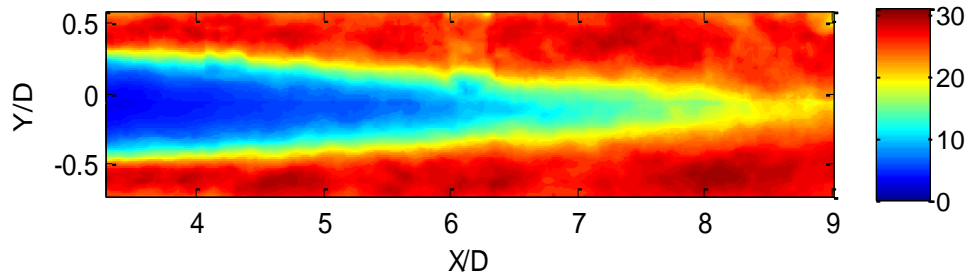
Figure 5.13: *Instantaneous Streamwise Velocity Estimated with Fused POD using Different Splitting Methods on an Axisymmetric Jet*



(a) Split Along the Center of the Domain



(b) Split Along the Center of the Jet



(c) Split Along the Centerline of the Jet

Figure 5.14: *RMS of the Streamwise Velocity Estimated with Fused POD using Different Splitting Methods in an Axisymmetric Jet*

The Effect of Wavelet Fusion

The next thing to discuss is the effect of using a wavelet fusion process on the data. As stated earlier, there is a potential for a discontinuity while using Gappy/Fused POD. The wavelet fusion process reduces the discontinuity as seen in figure 4.13. The wavelet fusion uses information from another source to do the smoothing. This means one is not simply smoothing the image with an arbitrary function. The wavelet fusion was accomplished with the Daubechies10 wavelet and one level of compression.

When filling-in data in the location of the sensor occlusions, the wavelet fusion reduces errors in the higher order statistics. Without using the wavelet fusion process, one can see a more distinct jump in the RMS of the streamwise velocity. The wavelet fusion alleviates this problem. This is shown in figure 5.15. There is slight improvements in the error as seen in figure 5.16, which shows the wavelet fusion increases the percentage of points that have 0 – 3% error from 57% to 59%. The maximum error is reduced slightly from 28.7% to 28.5% and the standard deviation is reduced from 3.7% to 3.5%.

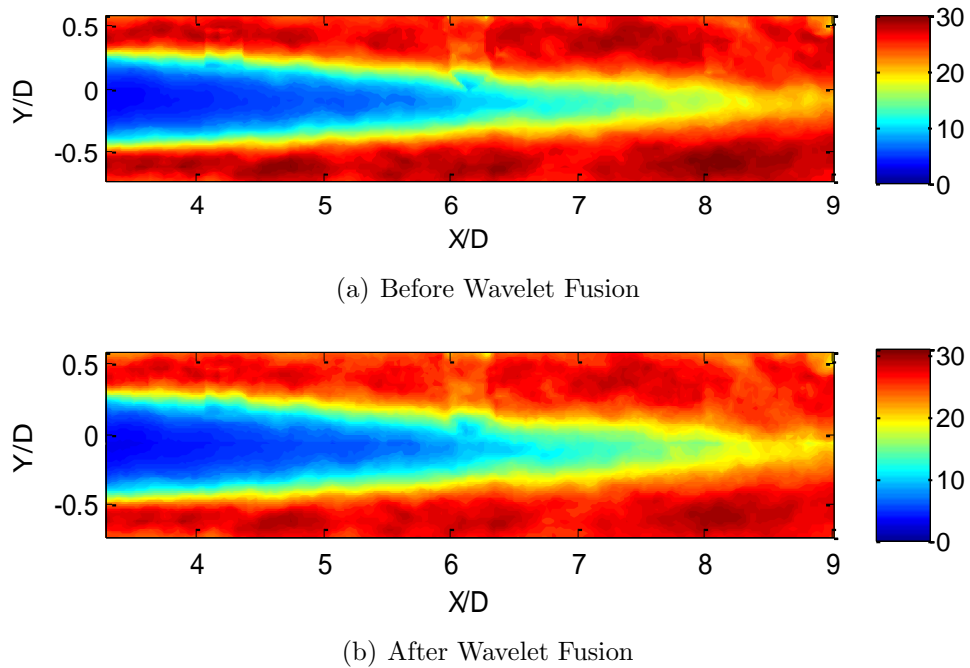


Figure 5.15: *RMS of Streamwise Velocity Estimated with Fused POD Before and After using Wavelet Image Fusion on a Mach 0.6 Jet*

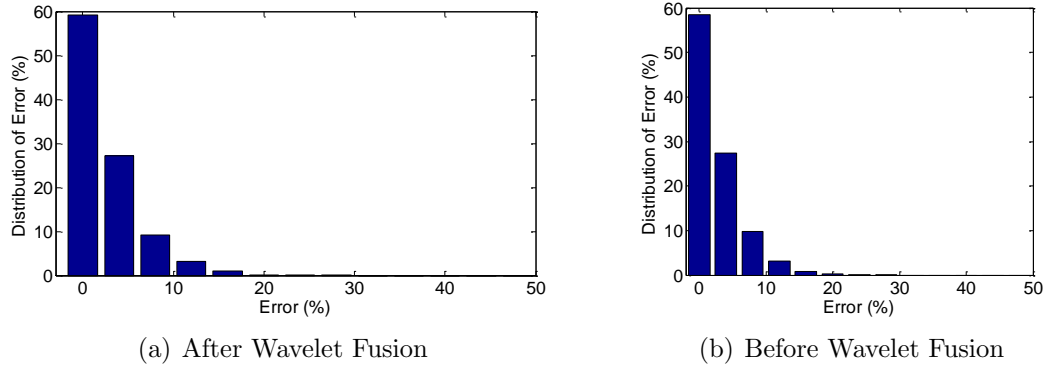


Figure 5.16: *The Effects of Using the Wavelet Fusion in Fused POD*

Effect of the Size of the Hole

Window size is an important factor in how well the data can be estimated, as seen in the work with the airfoil data set (section 4.6). If the missing data is larger, the accuracy of the estimated data decreases. This is studied by varying the size of the artificial occlusion and looking at the error distribution. The size of the artificial window ranges from 0.5 to 2 integral length scales in the streamwise direction and is held constant at 0.5 length scales in the radial direction. The effect of domain size is also studied by splitting the domain in half. The different configurations are shown in figure 5.17, where the hole is represented by a white box. The error distribution plots in figure 5.18 show as the size of the missing area range increases, the error increase. This is shown by the increasing standard deviation in the distribution plots. The plots also show that if domain is smaller, then the error in the estimated data is increased.

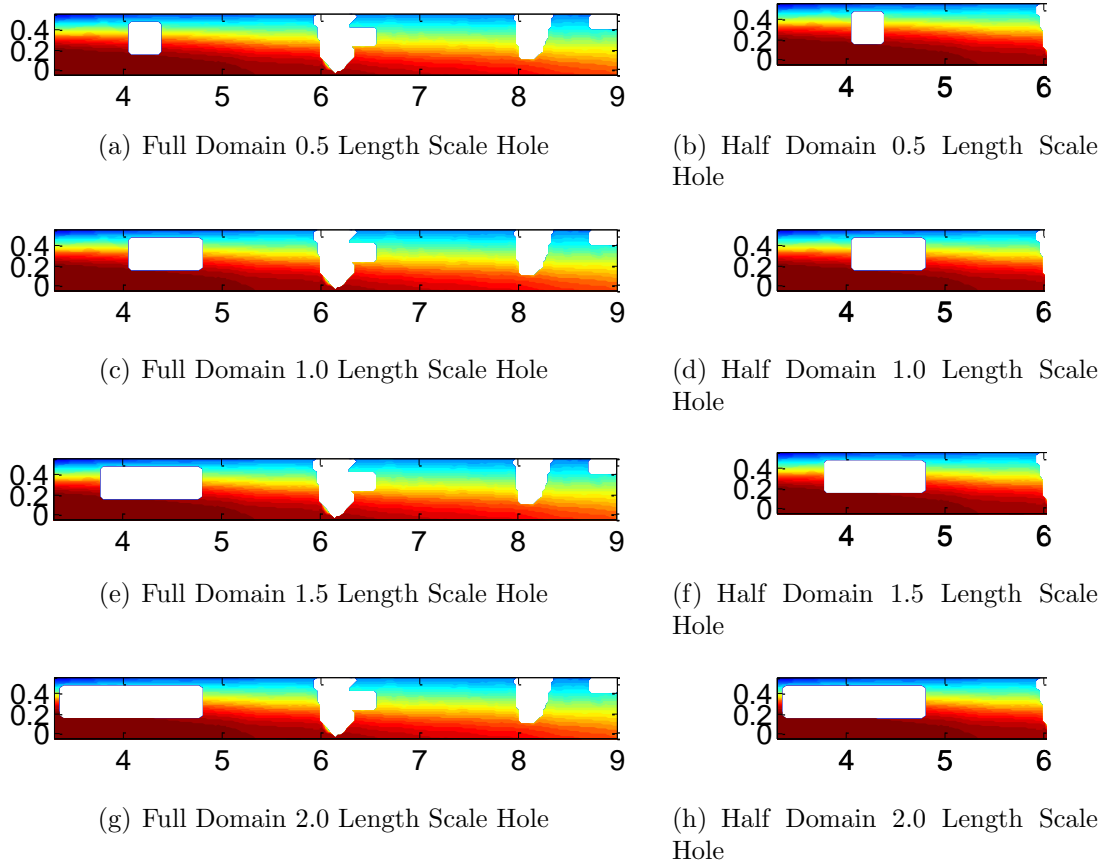
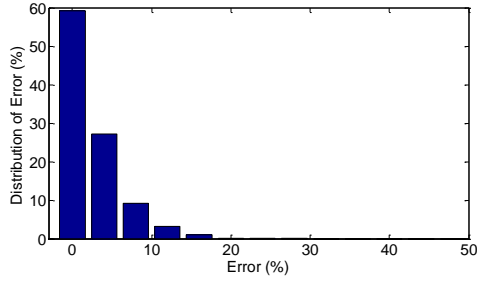
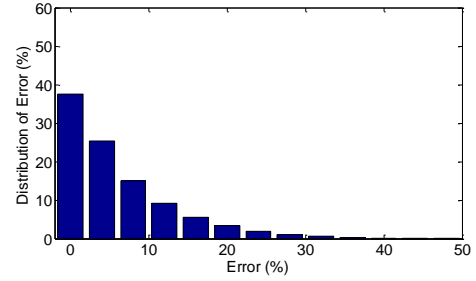


Figure 5.17: *The Different Artificial Hole Configurations Used to Test the Effect of Hole Size*

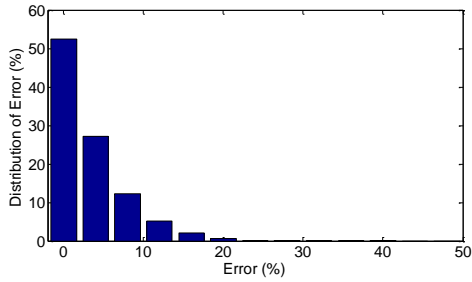
In both the large and small domain, the error (based upon the number of points with error less than 3%) is increased significantly between a 0.5 length scale hole and a 1.0 length scale hole (figure 5.18). Increasing the hole size more increases the error but not as noticeably as the jump in error between 0.5 and 1.0. The flow is correlated within one length scale; this is why the error reaches an asymptote after the hole is greater than 1.0 length scale as seen in figures 5.18 and 5.19.



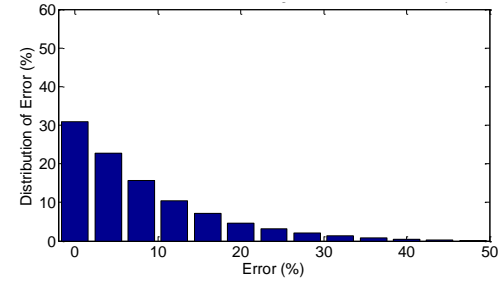
(a) Full Domain 0.5 Length Scale Hole



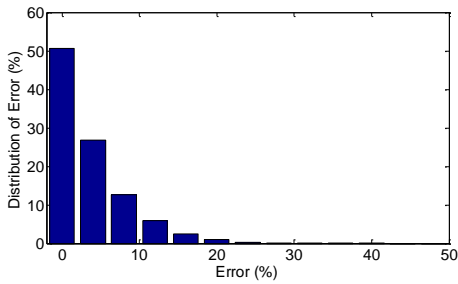
(b) Half Domain 0.5 Length Scale Hole



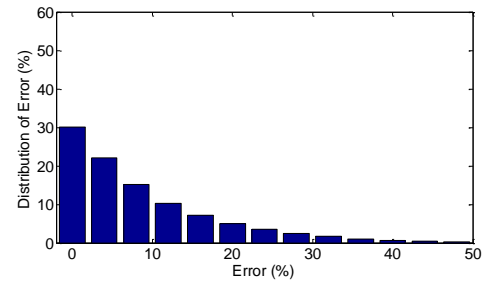
(c) Full Domain 1.0 Length Scale Hole



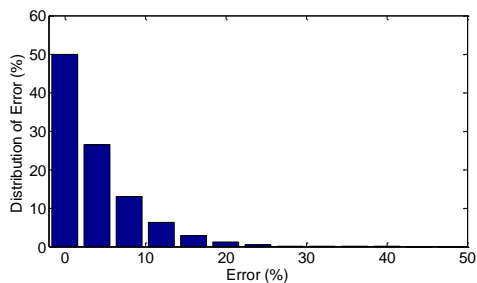
(d) Half Domain 1.0 Length Scale Hole



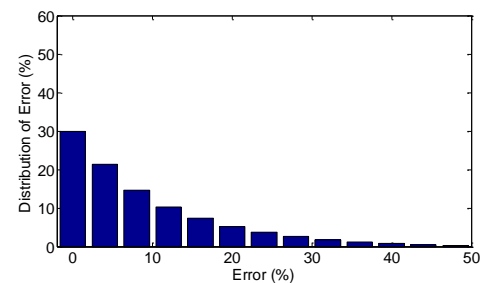
(e) Full Domain 1.5 Length Scale Hole



(f) Half Domain 1.5 Length Scale Hole



(g) Full Domain 2.0 Length Scale Hole



(h) Half Domain 2.0 Length Scale Hole

Figure 5.18: *Distribution of Error for Different Size Holes and Different Domain Sizes*

Figure 5.19 shows that the error between the actual data and the estimated data increases if the hole gets larger, and reaches an asymptote near one length scale. After

one length scale, the flow inside the center of hole begins to become uncorrelated with the data surrounding the hole. This makes it difficult to estimate the data, which is causing larger errors. When comparing the results for the large and small domain, the estimation using the large domain gives less error. So more information about the flow gives a better estimate. The larger domain has more information to fit and should give a more accurate estimation, which it does.

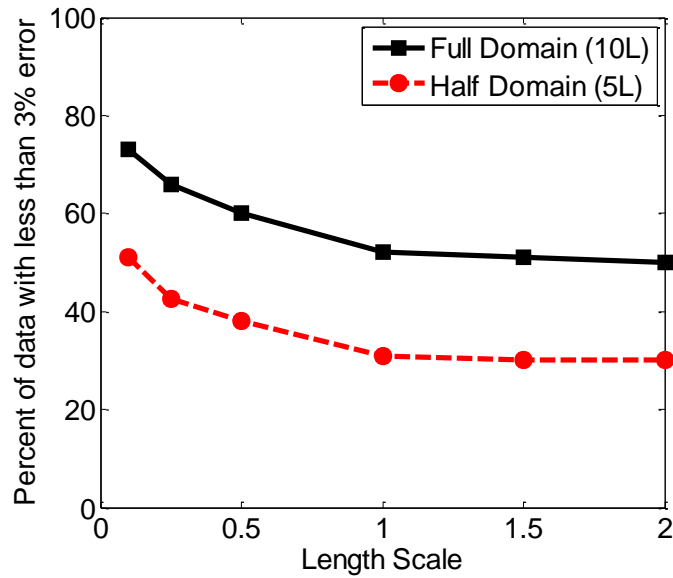


Figure 5.19: *Effect of Hole Size on Fused POD Estimate*

Convergence of the POD Modes

The convergence of the modes is also an important factor in the Fused POD process. The Fused POD assumes the modes are, in the case of the axisymmetric jet, the same top and bottom. If that is not the case, the Fused POD does not produce as accurate estimations. If the modes are not converged, then the modes may not match top and bottom. One of the reasons that the modes might not converge is due to lack of time support, (not enough snapshots). This is observed in figure 5.20, which shows the convergence plots when using a different number of snapshots to calculate the modes, as well as the RMS of the streamwise velocity. The convergence plots show how well

the modes are correlated with the modes calculated using half as many snapshots. If the modes are perfectly correlated, then the modes have converged. However, one would not expect to see a perfect correlation because the higher order modes will always have some noise associated with them and the energies are so small, that it becomes difficult to obtain those modes. Some modes may have very similar energies and may switch position (i.e. mode 5 may become mode 6 if using more snapshots). In general, the correlation plot will become more narrow as the modes converge as is seen in figure 5.20.

Figure 5.20 also displays the distribution of error in the artificial hole using different number of snapshots and the associated trace error, RMS error, and standard deviation are shown in table 5.1. One can see that as more snapshots are used the error decreases. This implies that using modes that are more converged produce better results when using Fused POD, which is consistent with what is qualitatively seen in RMS plots in figure 5.20.

Snapshots	Trace Error (%)	RMS Error (%)	σ
510	49	33	6.5
255	58	36	7.0
128	64	38	7.9
32	71	43	8.7

Table 5.1: *Effect of Modal Convergence on the Accuracy of Fused POD Estimates*

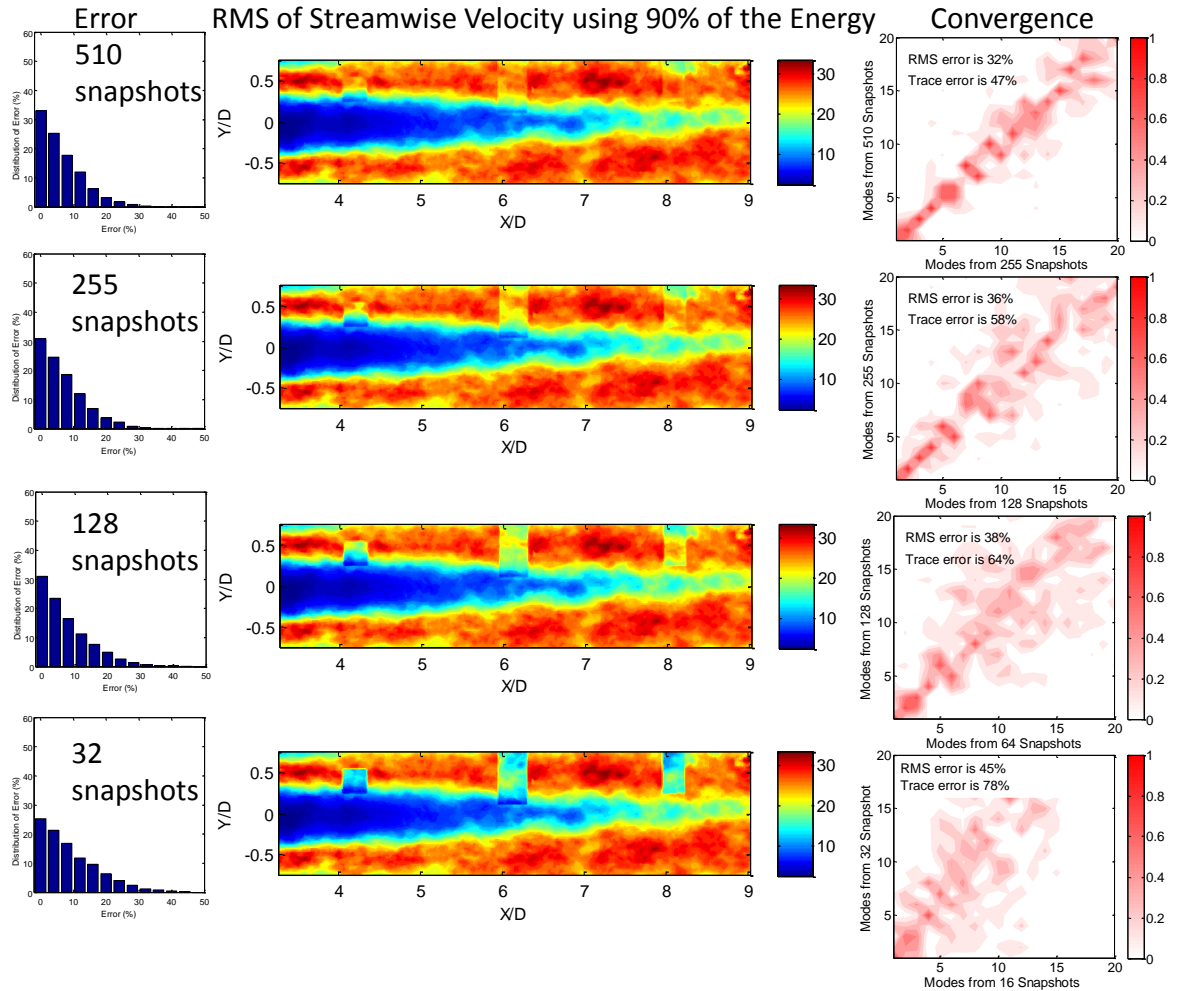
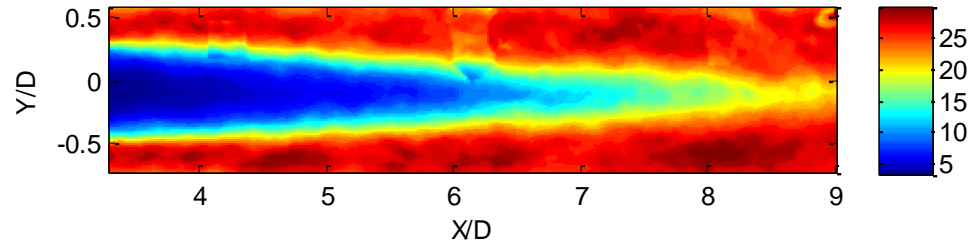


Figure 5.20: *Effect of POD Modal Convergence When using Fused POD to Fill-in Missing Data for an Axisymmetric Jet*

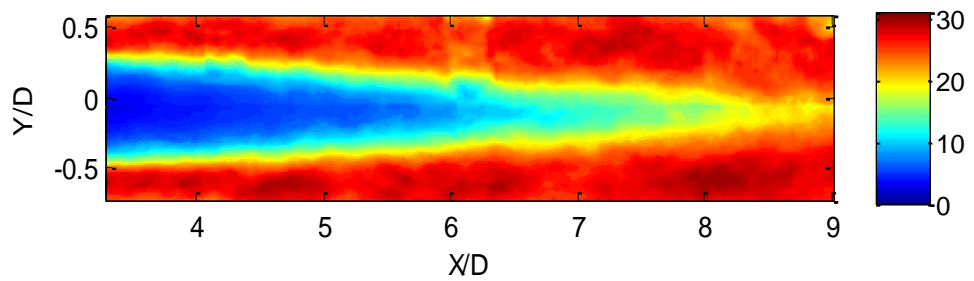
Using the Average in the Fitting Process

Another point to address is the effect of fitting the instantaneous (mean included) or fluctuating (mean subtracted) velocity. When fitting the instantaneous velocity, the average is used as mode 0, which contains most of energy. This makes the other modes less important in the fit and the results have higher error than if the average was subtracted. This is seen in figure 5.21, which shows an image of the streamwise RMS velocity using instantaneous velocity and fluctuating velocity in the Fused POD algorithm. The discontinuity in the RMS is more noticeable when using the instant-

neous velocity. This is quantified in figure 5.22, which shows the distribution of error in the artificial hole.



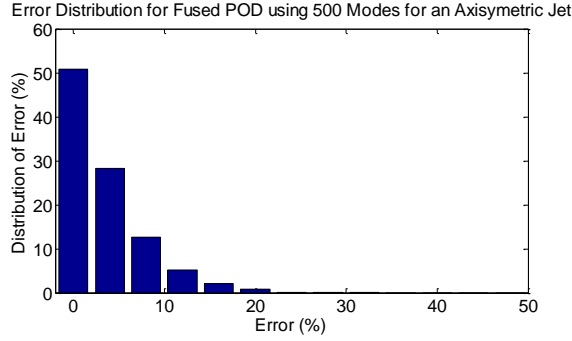
(a) Using the Instantaneous Velocity



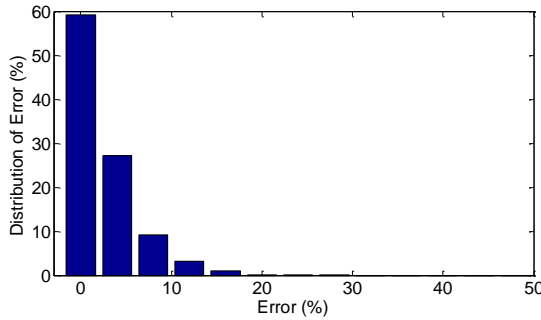
(b) Using the Fluctuating Velocity

Figure 5.21: *The Effects of using the Average in Fused POD*

As a note, sometimes subtracting the mean is not possible if the instantaneous field is required after the fusion process. In the case of the axisymmetric jet, it is possible because the mean flow is symmetric about the top and bottom, meaning information about the relationship between the mean of the two data sets (top and bottom) is known; the mean is the same top and bottom.



(a) Using Instantaneous Velocity



(b) Using Fluctuating Velocity

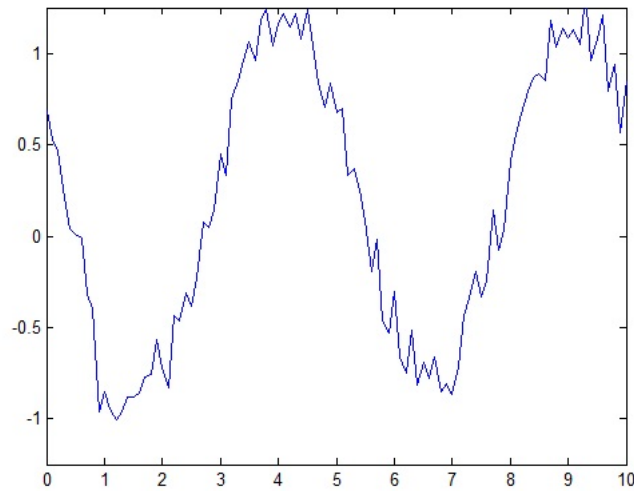
Figure 5.22: *Error Distribution for Fused POD using Instantaneous vs. Fluctuating Velocity*

The error distribution when using Fused POD with the instantaneous velocity is worse than if using the fluctuating velocity. The error distribution for the instantaneous velocity is spread out further ($\sigma = 3.25\%$) than the distribution for the fluctuating velocity ($\sigma = 2.5\%$), has fewer points with low error, and has a higher maximum error (figure 5.22).

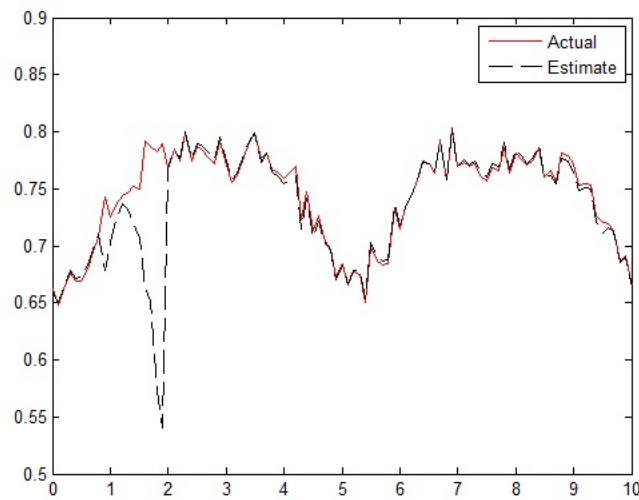
Why Higher Order Statistics Show the Hole

The last point to discuss is why the sensor is seen in the higher order statistics, specifically the RMS, even though it is not seen in the instantaneous. One would expect using more POD modes in the estimate would decrease the error. However, this is not the case unless the modes are exactly correct as shown with the airfoil data in section 4.6. Since the modes for the bottom are not the same as the modes

for the top of the jet, there is no guarantee of convergence. Using the higher modes does not aid the estimation since they are not correct and therefore the higher energy of the flow is cannot be estimated. This causes an under prediction of the RMS and the other higher order statistical moments.



(a) Instantaneous



(b) RMS

Figure 5.23: *Noisy Sine Wave Repaired with Sinusoidal Basis*

One can see a similar effect in the simple case of using sines and cosines as basis

functions to fill in the missing data in a sine wave with random noise. Since the basis functions do not have any connection to the random noise that was added to the sine, they are unable to estimate said noise. However, the noise levels are large enough that the error in the estimate is hidden in the noise. Therefore, the instantaneous image looks correct (figure 5.23a) but the hole is seen in the RMS (figure 5.23b).

With this in mind, the jet data is filtered using POD. In this particular case 20 modes are used in the reconstruction, which corresponds to 40% of the energy. Twenty modes were chosen because this is where the correlation between the top and bottom modes is 50%, which has gives the best representation of the estimated flow field without sacrificing too much of the higher order information. This is shown in figure 5.24, which shows the low-dimensional representation of the instantaneous streamwise velocity after Fused POD is applied. Observing figure 5.25, which shows the RMS of the streamwise velocity using 20 modes, one can see the RMS no longer has evidence of the occlusion from the sensor.

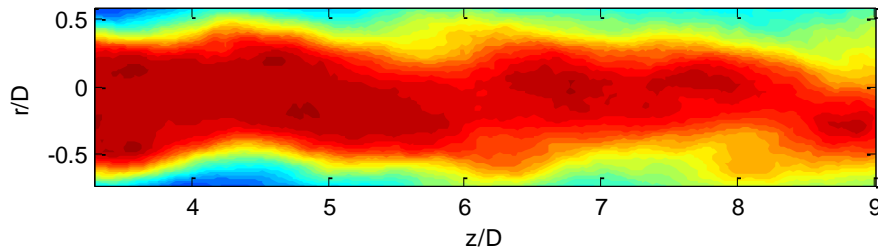


Figure 5.24: *Low-Dimensional Representation of the Streamwise Velocity*

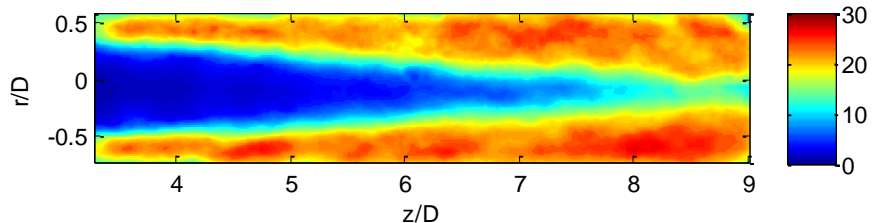


Figure 5.25: *Low-Dimensional Representation of the RMS of the Streamwise Velocity*

5.4.2 Estimation of Spurious Vectors/Particle Dropout

In addition to filling-in missing data where the holes are the same in every snapshot, Fused POD can also be used to fill in data when the missing points are “randomly” distributed in time and space. This can also be done with the iterative version of Gappy POD. However the iterative method can take a long time ($\mathcal{O}(\text{weeks})$) depending on the number of images in the ensemble. The processing time is proportional to approximately the number of snapshots squared because the two-point correlation matrix needs to be computed for every iteration and the least squares has to be solved as well. Both of these calculation require n -squared operations. The iterative analysis took eight hours on 1,000 of the 15,000 snapshots of the time-resolved data. If all of the data was used it could take about 12 weeks. This is not a practical time to wait to repair the data. To work around this, Fused POD is used.

To repair the time-resolved data set, 1,000 snapshots are extracted from the original 15,000. They were extracted by taking every 15th snapshot of the ensemble. The snapshots should be independent since about three times scales are between them in the 1,000 snapshot ensemble. It is assumed that 1,000 snapshots represented the data set well enough statistically. The iterative Gappy POD process is applied to the extracted snapshots, which only takes about eight hours to perform. Once the

snapshots are repaired, they are treated as a second data set in which the statistics are known at all points in space. These snapshots are used to create a basis that all 15,000 snapshots are projected onto; this process takes about two hours.

Using the Fused POD technique instead of the Gappy POD technique, the snapshots were repaired in a much faster time. This makes it feasible to repair the snapshots since the Gappy POD technique would take about 12 weeks to repair a single test case. Also, recollecting the data could be difficult since the TRPIV system was loaned to Syracuse University and the data may not be collected again. Even if the system was acquired again, it would take a few days to setup and a few more days to acquire and process the data. These data can be repaired in a matter of hours using the Fused POD technique.

The TRPIV data had many data that was lost due to seed drop out or laser misalignment. If the lasers are not aligned, the PIV software will compare the motion of the particles on two different planes. This, as expected, can cause some error since the velocity is being calculated incorrectly. If there is drop out in the seeding, the velocity cannot be calculated correctly. This is seen in figure 5.26, which shows an instantaneous snapshot of streamwise velocity without performing any type of post processing to account for the data that does not physically make sense because of large velocity gradients, large velocities, and high fluctuations all in regions where this should not exist. This is seen in figure 5.26, which shows the instantaneous streamwise velocity in a Mach 0.6 jet. One can see many “bad” spots in the image that are shown as the zero vectors in the image.

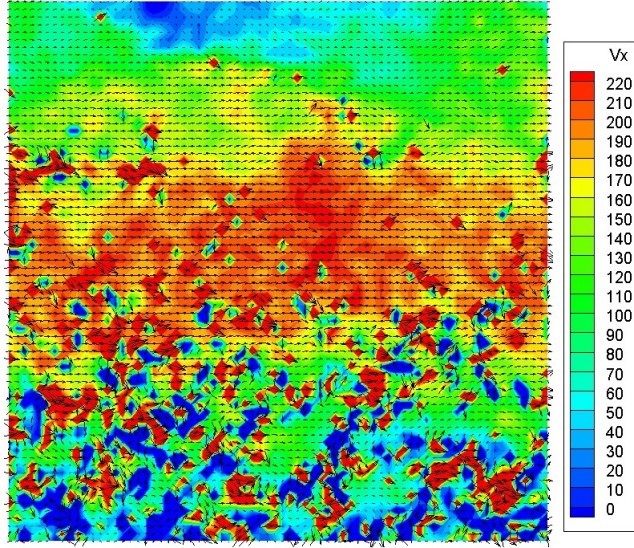


Figure 5.26: *PIV Data for a Mach 0.6 Jet with no Post Processing to Compensate for Errors*

The PIV software can be used to improve the data by performing interpolation or looking for other maxima in the correlation process. The result of this post processing technique is seen in figure 5.27, which shows the same instantaneous snapshots in figure 5.26 after the post processing is performed. This process removes some of the non-physical data, but there are still regions that do not make physical sense. These regions are blanked out with a mask (figure 5.28) defined by any point in the flow with a fluctuating (mean subtracted) velocity larger than 75 m/s. Fluctuating velocities greater than 75 m/s should not theoretically exist in this flow. A better method of finding the “bad” points could be used such as a method suggested by Lewalle *et al.* [155], which involves finding points where the flow changes more than a desired percent of the local flow. This would be a more rigorous method for generating the mask and would possibly find more non-physical points. As one can see in figure 5.28, there are still some bad points that should have been masked out.

Fused POD is applied to the data and improves the spots that were masked in figure 5.28. The repaired image is shown in figure 5.29, which shows the same instantaneous snapshot as figures 5.26, 5.27, and 5.28 after the PIV post processing

and Fused POD were applied. The number of non-physical points have been reduced by about 80% using the PIV post processing and then another 15% using the Fused POD.

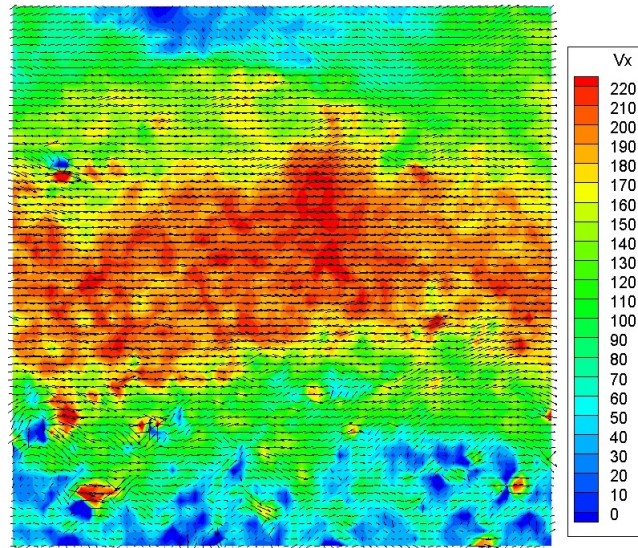


Figure 5.27: *PIV Data for a Mach 0.6 Jet with PIV Post Processing to Compensate for Errors*

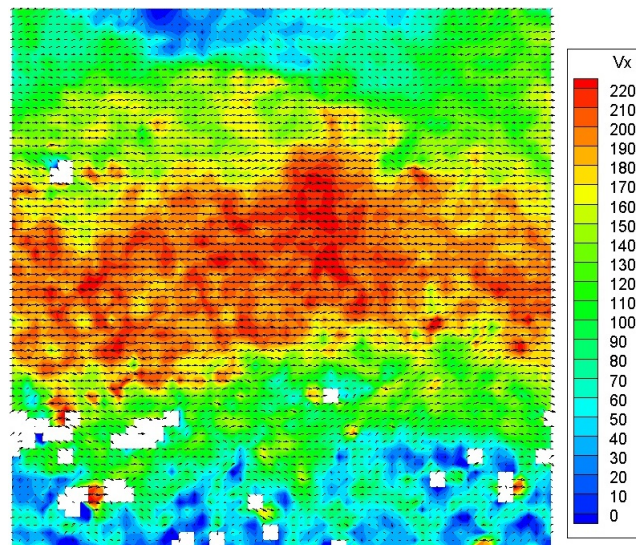


Figure 5.28: *PIV Data with Bad Vectors Masked Out*

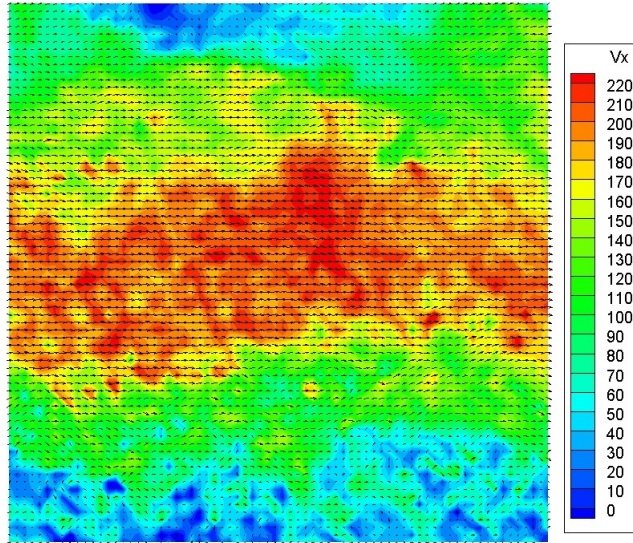


Figure 5.29: *PIV Data with Gappy POD Corrections*

Looking at the statistics shown in figure 5.30, one can see the improvements provided by Fused POD. The mean is very similar for both data sets (RMS difference is 1.5%) therefore the Fused POD did not change anything in the mean sense (figure 5.30). However, the Fused POD makes the RMS of the streamwise velocity more symmetric. This is what is expected since the jet is axisymmetric and should have similar RMS values in the shear layer.

In figure 5.30 one will notice a red streak in the RMS after using Fused/Gappy POD in the lower portion of the figure. This is caused by the fact that the masking method is not finding some of the “bad” points in that area. These points have not been repaired and are causing an increase in the RMS in that area. Using a better method of masking, perhaps the one suggested by Lewalle [155], may decrease the red strip in the RMS.

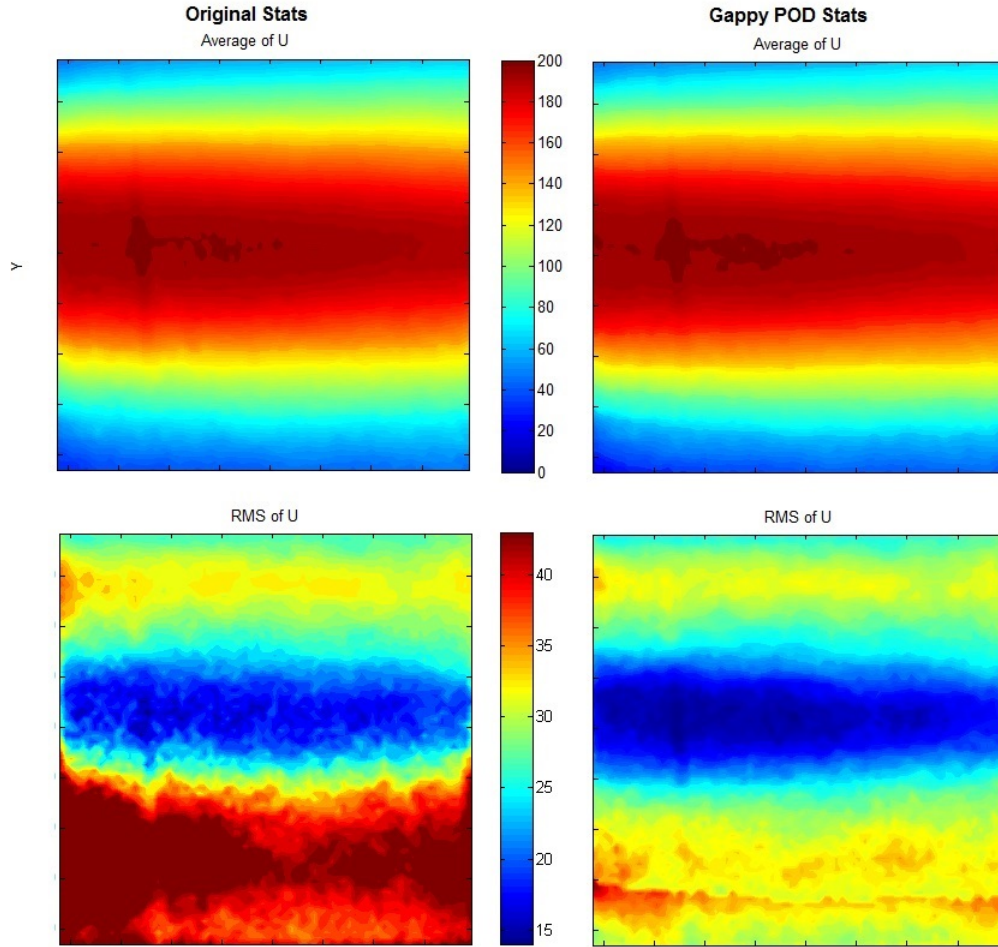


Figure 5.30: *Statistics for the TRPIV Before and After using Fused/Gappy POD for a Mach 0.6 Jet*

One may question the validity of this technique. Iterative Gappy POD has been shown to work by past researchers (see [subsection 1.4.2](#)), but how well does the Fused POD procedure work in the case of repairing data using a subset of the snapshots to generate the modes? The Fused POD technique produces results faster, but the results need to be similar to that of iterative Gappy POD to be a viable substitute. To test this, 1,250 snapshots are extracted from the entire 15,000 snapshot ensemble. Iterative Gappy POD is performed on the 1,250 snapshots and allowed to converge 6 orders of magnitude. This process is accomplished using 50 modes, approximately 35% of the total energy and took about 28 hours. The result of this test is said

to be the “correct” result, the benchmark for comparison. The POD modes are calculated from the iterative Gappy POD case and used to repair the original, un-repaired snapshots. The comparison shows 95% (2σ) (both space and time) of the Fused POD repaired data is within 9% of the Iterative Gappy POD data. The 1,250 snapshot ensemble is cut in half and iterative Gappy POD is used to repair the smaller ensemble. The repaired 625 snapshot ensemble is used to create a POD basis. This basis is then used to repair the original 1,250 snapshots using Fused POD. These results are compared with the 1,250 snapshot ensemble repaired with the iterative Gappy POD. The comparison shows 95% of the Fused POD repaired data is within 10% of the Iterative Gappy POD data. There is a slight increase in the error when using fewer snapshots to generate the POD basis. This process is repeated using 312 and 156 snapshots to create a POD basis. The results of this are shown in figures [5.31](#), [5.32](#), and [5.33](#).

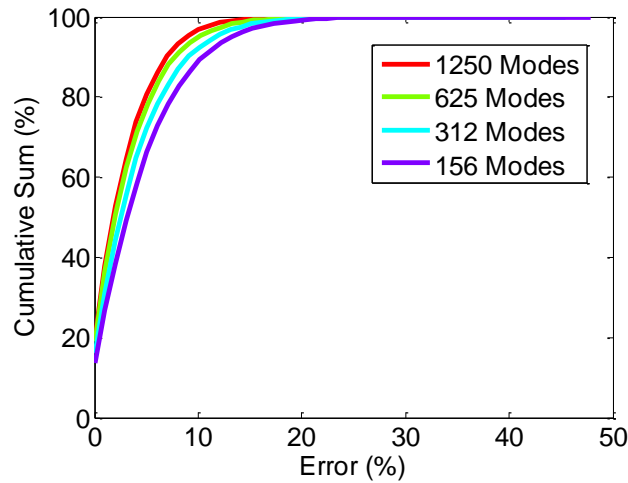


Figure 5.31: *Error Between Fused POD and Iterative Gappy POD at the Cumulative Sum of the Percent of “Bad” Points*

Figure [5.31](#) shows the error between the velocity generated with iterative Gappy POD using 1,250 snapshots and the velocity generated with Fused POD using a varying number of snapshots. The plot shows the percentage of points that have an

error less than a specified error. Each line in figure 5.31 corresponds to the number of snapshots repaired with iterative Gappy POD that were then used to generate the POD modes. For example, the 625 snapshot line used iterative Gappy POD to repair 625 snapshots. Those snapshots are then used to generate a set of modes that are used with Fused POD to repair the 1,250 snapshots originally extracted from the entire ensemble. Using a smaller subset of the data decreases the consistency between the data generated by the two methods. This makes sense because less information about the flow is being used in the repair process. Despite that fact, the error is not increasing much. Repairing an eighth of the snapshots using iterative Gappy POD only increases the error by four percent but decreased the computation time by two orders of magnitude. This test was repeated for 625 snapshots and 312 snapshots. The results are shown in figure 5.32, which show the number of snapshots being reconstructed do not change the error more than a few percent. Figure 5.32 shows a plot of how the number of snapshots repaired using Gappy POD to create a basis affect the accuracy of the snapshots repaired using Fused POD and the aforementioned basis. The different symbols represent a different number of snapshots repaired using Fused POD; the abscissa shows the number of snapshots used to generate the basis. Each point was repaired using approximately 50% of the energy. The plot shows using fewer snapshots to create the basis increases the error but not by a substantial amount.

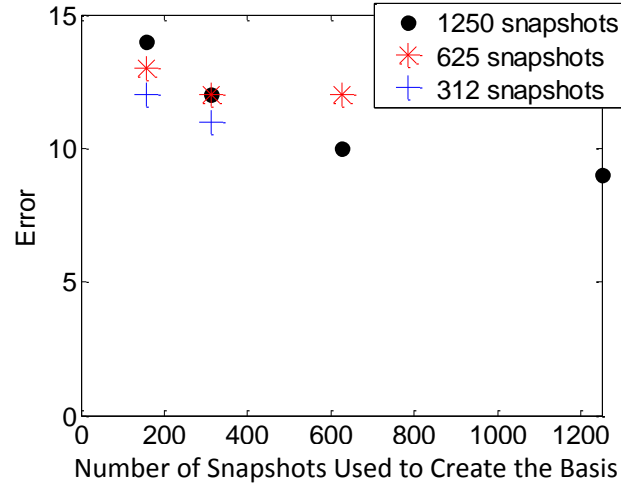


Figure 5.32: *The Error Associated with Using Fused POD Instead of Iterative Gappy POD to Repair PIV Snapshots with Missing Data*

This shows the Fused POD and iterative Gappy POD give slightly different results. However, the length of time that is saved by using Fused POD greatly outweighs the error between the Fused POD and iterative Gappy POD. In fact, it would be impractical to try to repair the entire 15,000 snapshot ensemble using only iterative Gappy POD. This is shown in figure 5.33, which shows the time required to repair varying numbers of snapshots using iterative Gappy POD and Fused POD. The time required to repair the data using Fused POD is nearly constant with increasing number of snapshots, while the time required for Iterative Gappy POD increases as number of snapshots squared as stated earlier.

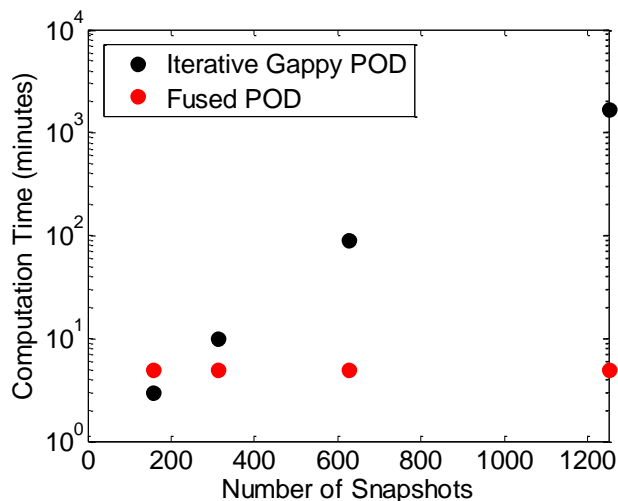


Figure 5.33: *The Computational Time Savings from Using Fused POD Instead of Iterative Gappy POD to Repair PIV Snapshots with Missing Data*

5.5 Domain Extension/Estimation

Another issue with the jet data is the small window size of the TRPIV and the “slow” sampling rate of the LWPIV. One would prefer a large-window, time-resolved data set if possible. Many reasons exist as to why this is desirable, such as better understanding of the flow physics, and the ability to create better low-dimensional models for control purposes.

For modeling, many researchers use low-dimensional models that are POD based (see [subsection 1.4.4](#)). The POD modes are spatially dependent, especially for the r-z plane of the jet flow. This means the POD modes change depending on how large of a window is used. Since the model would be dependent on the POD modes, it is also dependent upon the size of the window. These models are typically used for control applications. Obtaining a large-window time-resolved data set should improve the model and its predictive capabilities.

The ultimate goal is to use the TRPIV, LWPIV, and pressure to obtain a large-window time-resolved data set that could be used to generate a model for flow control.

This work will focus on the techniques that are used to obtain the large-window time-resolved data set.

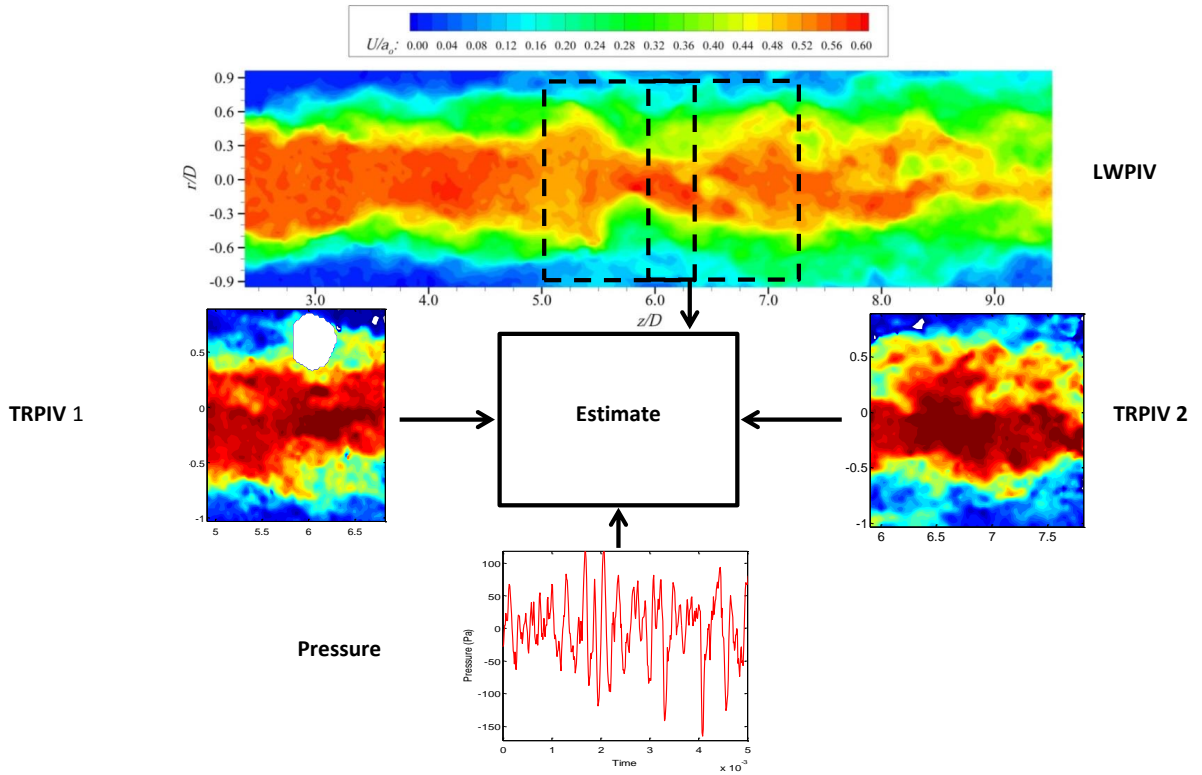


Figure 5.34: *The Ultimate Goal of the Estimation Techniques: Combine Pressure, TRPIV, and LWPIV to Obtain a Large-Window Time-Resolved Data Set*

Figure 5.34 shows an image of the LWPIV data (top). This is a contour of the streamwise velocity and the flow is from left to right. Two snapshots of TRPIV are shown (left and right images), which also depict streamwise velocity. Each TRPIV window fits in the LWPIV. TRPIV1 fits in the left most dotted box on the LWPIV and TRPIV 2 fits in the right most dotted box. It should also be noted that each TRPIV and the LWPIV are at different times, meaning they cannot directly be stitched together. A signal from one of the pressure signals is also shown (bottom). This signal was obtained simultaneously with the LWPIV. Similar pressure data was acquired for the TRPIV.

The techniques described in [section 1.4](#), such as LSE, stochastic modeling, and POD are used to estimate a large-window time-resolved data set. The techniques are compared to determine the best technique to use and why the techniques give the results they are giving. The estimations ultimately will be used to develop control algorithms so the estimates do not have to match exactly, only a basic understanding of the flow structure, time scales, and spectral content are required.

5.5.1 Stochastic Estimation Results

The first technique used is spectral linear stochastic estimation (sLSE). This technique is discussed in [subsection 1.4.3](#). The goal of the stochastic estimation is to combine the time resolution of the pressure signals and the spatial resolution of the LWPIV as is shown in [figure 5.35](#). In this particular case, the TRPIV data is not used in the estimation. However, in future work one may want to use the time-resolved velocity to accomplish the estimation, which may give higher correlations than the pressure. This has been done in the past by Tinney *et al.* [57] using the $r - \theta$ plane of a jet flow, and Ausseur *et al.* on a flow over an airfoil [67]. However, time-resolved velocity data was not available to make a direct comparison. Therefore, to start, the sLSE technique is first verified using the TRPIV data.

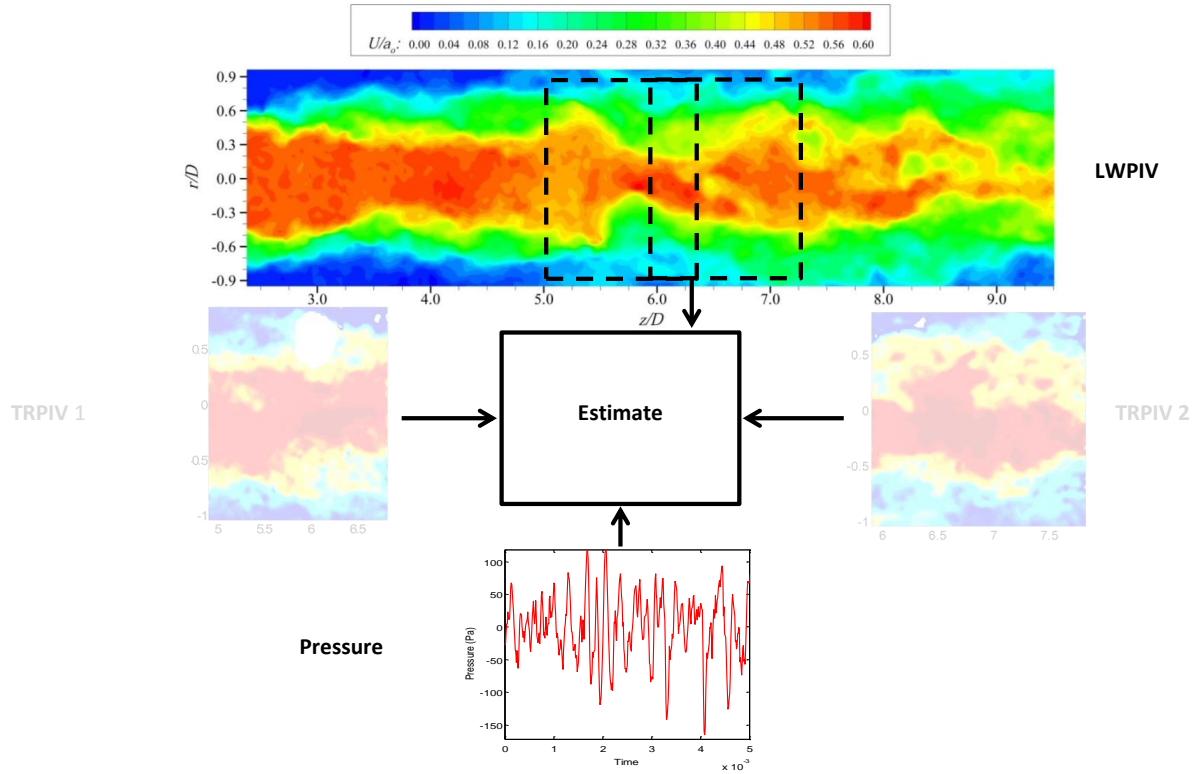


Figure 5.35: *The Ultimate Goal of the sLSE: Combine Pressure and LWPIV to Obtain a Large-Window Time-Resolved Data Set*

To accomplish this, the POD modes of the TRPIV data are computed at a window 6 to 7.5 diameters downstream of the jet nozzle. This gives the spatial eigenfunctions (the shape) and the time-dependent POD coefficients (the magnitude). A mapping is then generated from pressure to POD coefficients using the sLSE technique (equation 1.35). With this mapping, the POD coefficients can be estimated based on the pressure (equation 1.34) and then an estimate of the velocity can be acquired using equation 1.13.

This procedure allows the comparison of the actual POD coefficients (which are time-resolved) with the POD coefficients estimated using sLSE. The first two POD coefficients are seen in figure 5.36. The top half of the figure shows the time trace of the two modes and the bottom half shows the frequency space of the two coefficients. Looking at these plots, it is determined that the sLSE did not do a very “good”

job estimating the POD coefficients. There was an $\mathcal{O}(20\%)$ error between the actual coefficients and the estimated coefficients. The error is defined in the following manner

$$\frac{\sqrt{\frac{\sum_{i=1}^T (a_n(t_i) - \tilde{a}_n(t_i))^2}{T}}}{\max(a_n(t_i), \tilde{a}_n(t_i)) - \min(a_n(t_i), \tilde{a}_n(t_i))} \quad (5.5)$$

where $a_n(t_i)$ is the actual coefficient at time i and $\tilde{a}_n(t_i)$ is the estimated POD coefficient at time i .

It is interesting to note that the POD coefficients appear to have similar behavior in time. However, upon closer observation one can see the estimate is not capturing the signal well. This is shown in the frequency domain. Here it is observed that the estimate is missing the low frequency information and the estimate is picking up large peaks in the frequency domain (figure 5.36).

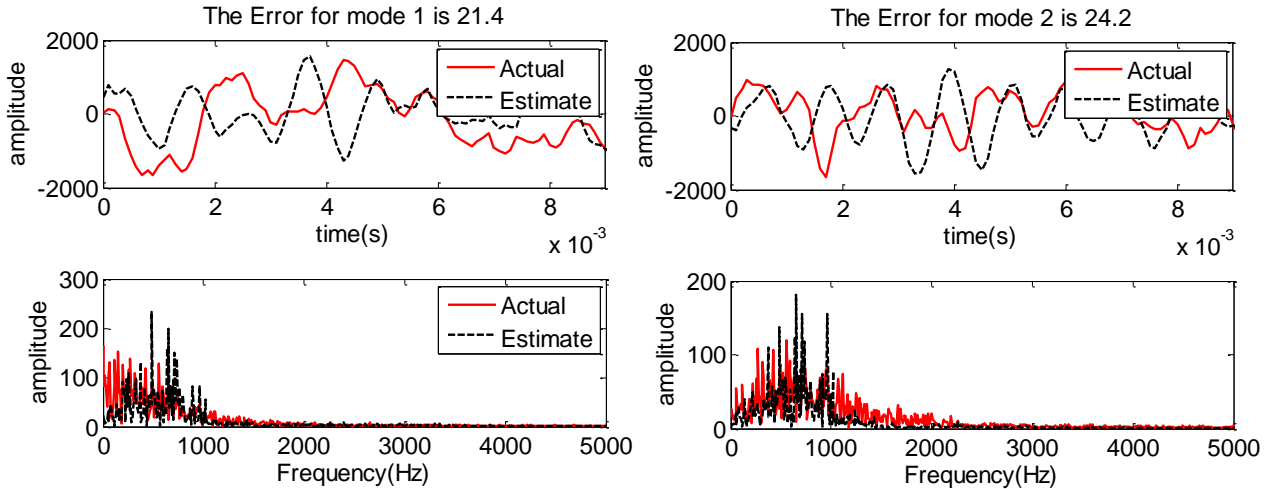


Figure 5.36: *Actual POD Modes 1 and 2 Compared with Estimated Mode 1 and 2 Using the sLSE Approach*

The velocity field is reconstructed using the spatial eigenfunctions and the estimated POD coefficients. In this work, five modes are used in the reconstruction which corresponds to $\mathcal{O}(20\%)$ of the energy. The estimated velocity is then compared to

the actual velocity using the following

$$\frac{\sqrt{\frac{\sum_{i=1}^T (u(\vec{x}, t_i) - \tilde{u}_i(\vec{x}, t_i))^2}{T}}}{\max(u_i(\vec{x}, t_i), \tilde{u}(\vec{x}, t_i)) - \min(U(\vec{x}, t_i), \tilde{U}(\vec{x}, t_i))} \quad (5.6)$$

where $u(\vec{x}, t_i)$ is the actual velocity and $\tilde{u}(\vec{x}, t_i)$ is the estimated velocity.

The RMS error between a 5 mode reconstruction of the actual velocity and the sLSE estimate using 5 modes is shown in figure 5.37. The plot shows the RMS error over space at different instances in time. One can see large fluctuations in the error, implying that sometimes the estimate is really close to the actual data, but other times the estimate is completely wrong. On average, the error is about 80%.

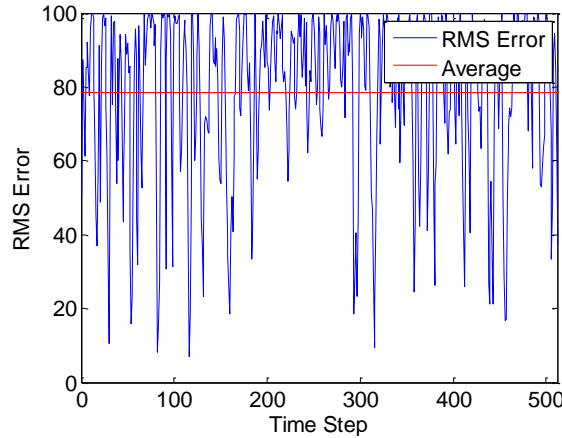


Figure 5.37: *Error Between the Actual Velocity and the Estimated Velocity Using the sLSE Approach with 5 POD Modes*

The reason for using five modes, is because it is the fewest number of modes where the flow structures still propagate in the direction of the flow and using more modes increases the error between the actual velocity and the estimated velocity. The increasing error is caused by the fact that higher order modes are including more information. The estimate is unable to capture that information correctly causing a compounding effect. Similar trends in the error are seen in the cross-stream velocity. This is shown in figure 5.38, which shows the average RMS error between the actual

velocity and the estimated velocity using different number of modes in the estimation.

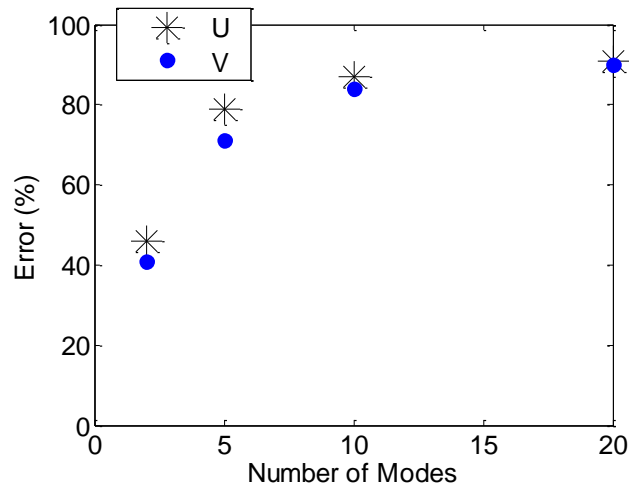


Figure 5.38: *The Average Error in Velocity as a Function of Modes used in the sLSE Estimate*

The effect of using different pressure sensors can also be studied. In this test, the pressure sensors at $z/D = 5$ and $z/D = 6$ are used in the estimation. These sensors had the largest correlation with velocity as one would expect since the velocity field between 6 and 7.5 is being estimated. Using more sensors did not improve the results. This is because the higher correlated sensors are weighted much higher than the sensors with a lower correlation. In other words, the estimation relies more heavily on the strongly correlated signals than the poorly correlated signals. Figure 5.39 shows the average error between the actual velocity and the estimated velocity using five modes and a varying numbers of pressure signals. The average error does not vary much based on the number of sensors used.

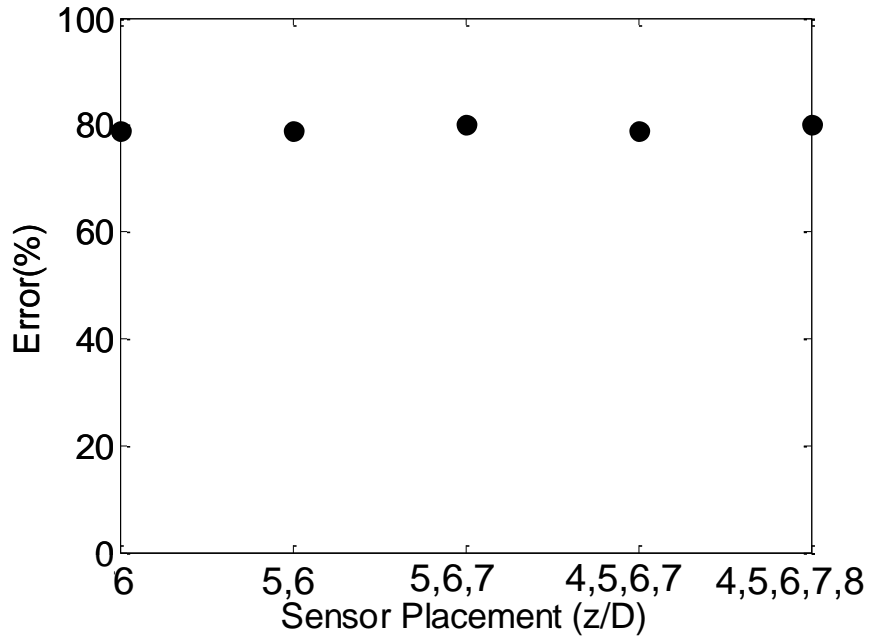


Figure 5.39: *The Effect of Using a Different Number of Pressure Signals in the sLSE Estimate*

Figures 5.40 and 5.41 shows the correlation between the pressure at five and six diameters downstream of the nozzle and velocity using a single- and multi-time approach. The single-time correlation images show a much weaker correlation than the multi-time correlation. This is because the multi-time determines the best lag to use depending on the location in the flow. The correlation plots in figures 5.40 and 5.41 show that the maximum correlation is 30% and the highest levels of correlation are near the lower shear layer. The low levels of correlation and the fact that the highest correlations are localized in the shear layer show using sLSE will not give great results.

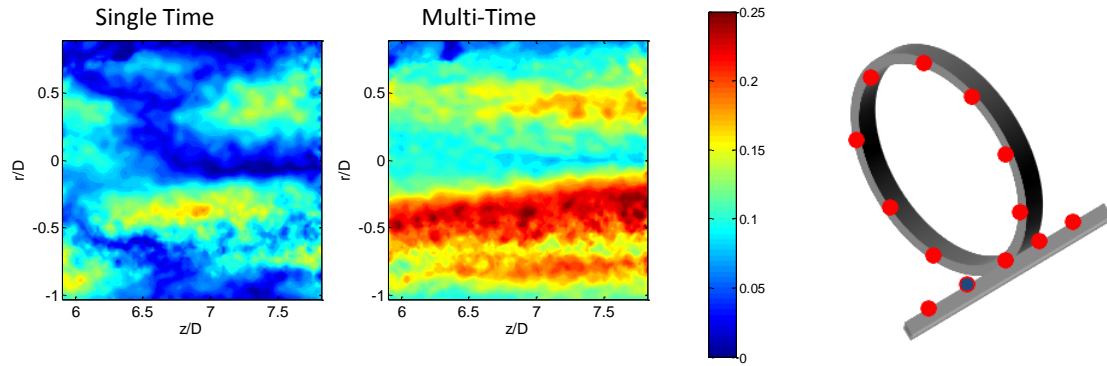


Figure 5.40: *Correlation Between the Pressure Sensor at 5 Diameters Downstream of the Jet Nozzle and the Velocity*

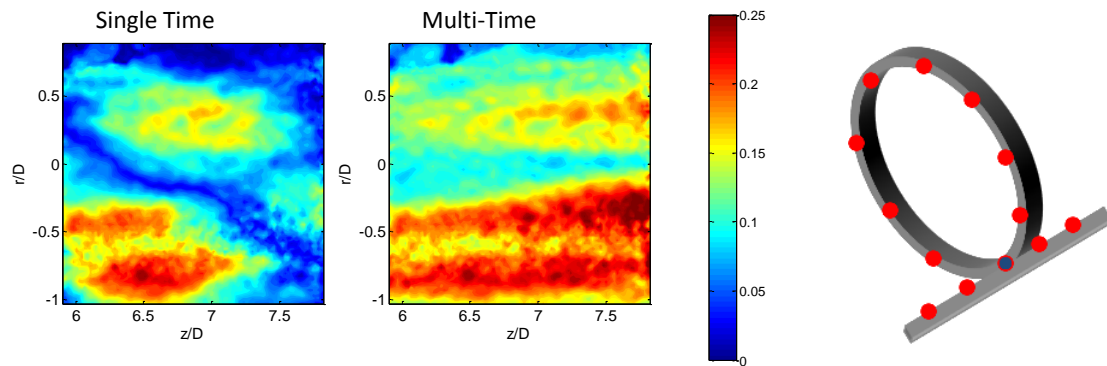


Figure 5.41: *Correlation Between the Pressure Sensor at 6 Diameters Downstream of the Jet Nozzle and the Velocity*

Looking at the correlations shows that the given pressure and velocity data are not correlated well. This means the estimate will not produce a reliable model because the spectral information is incorrect. This is troubling because the work by Tinney *et al.* [57] showed the sLSE worked well at predicting the jet data and was expected to produce “good” results with this data. However, after studying the different cases, there are some big differences that attribute to the conflicting results.

First, the work by Tinney *et al.* [57] was done in the $r - \theta$ plane whereas the data for this work is from the $r - z$ plane. It has been shown that the velocity

modes in the $r - \theta$ plane are Fourier modes. Since this is the case, Tinney *et al.* was able to filter the pressure and velocity to improve the correlations using Fourier modes. The pressure Fourier modes were then used to estimate the velocity Fourier modes. Tinney *et al.* was able to exploit that fact and get a reasonable estimation. In this work, the pressure is still arranged in an azimuthal array. Feeding back Fourier filtered pressure did not improve the results in this study because the velocity modes are not Fourier modes in the $r - z$ plane.

A second issue that could decrease the accuracy of the technique is the location of the linear pressure array. One would assume the linear pressure array would have the highest correlation with the velocity field since the array is in a similar direction. However, the linear array is shifted 20° away from the PIV plane in the azimuthal direction, which reduces the correlation strength as shown by the work done by Ukeiley *et al.* [156]. The shift was done so the laser does not reflect off the sensor array. In hindsight, it would have been better to get the linear array closer to the PIV plane and increase the correlation.

5.5.2 Stochastic Modeling

A large-window time-resolved data set could not be estimated with the given jet data for the reasons discussed above. However, a large-window time-resolved data set is still desired from this data. Spectral LSE is not the only technique available to accomplish this task. Modeling is another candidate technique to accomplish the goal. As shown in figure 5.42, the TRPIV data will now be used as an estimator, instead of using pressure due to the poor pressure-velocity correlations and the poor results from the sLSE approach.

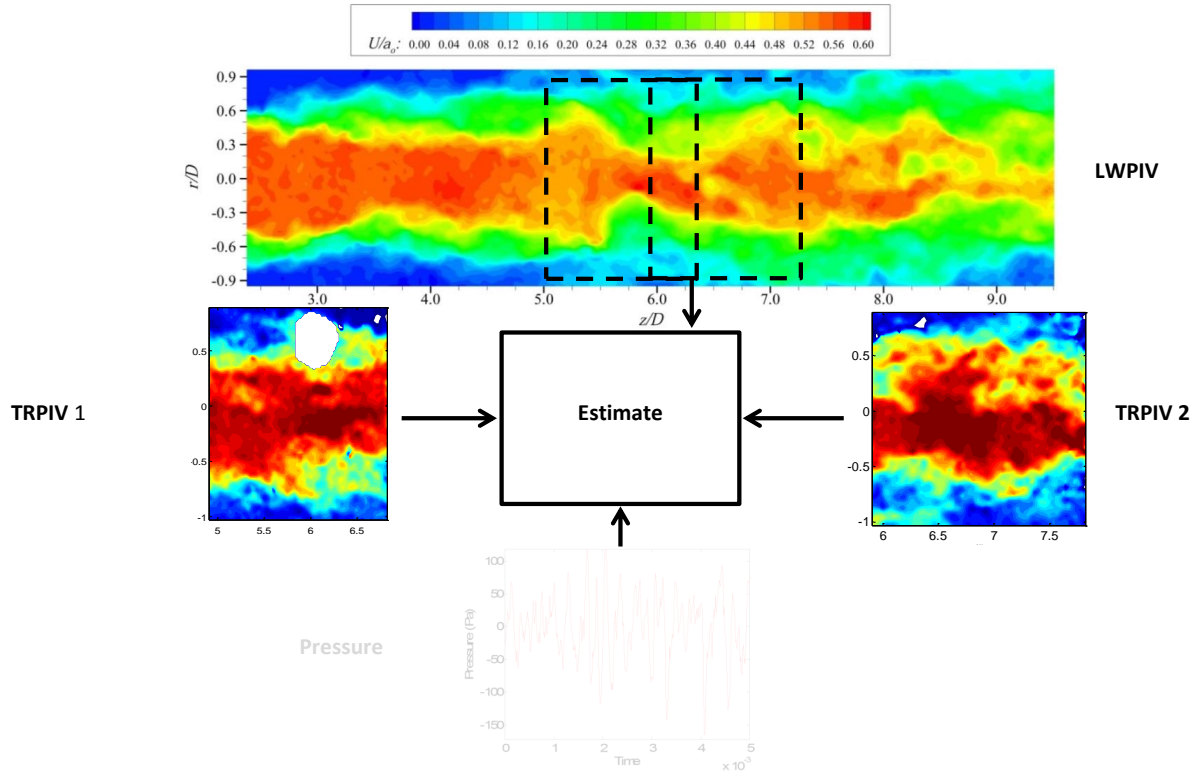


Figure 5.42: *The Ultimate Goal of the Stochastic Modeling, Gappy POD, and Dynamic Gappy POD: Combine Pressure and LWPIV to Obtain a Large-Window Time-Resolved Data Set*

The modeling approach, which is outlined in figure 5.43, uses the TRPIV to produce a model for the jet data. The modeling approach will start with the LWPIV data (pink box) and the data is cut to match the size of the two TRPIV windows that are next to one another. Next the two TRPIV windows TRPIV1 and TRPIV2 (black boxes) are used to create two different stochastic model using equation 1.39. The procedure to create the model is explained in subsection 1.4.4. This produces two sets of modeling coefficients. POD is performed on the two portions of the LWPIV data and the POD coefficients from them are used as initial condition for the models. The modeled POD coefficients can then be used with the spatial POD modes to reconstruct the velocity (green boxes Fused1 and Fused2). The two modeled velocity fields are sampled at the same time and can therefore be stitched together to create

a large-window time-resolved velocity field (green box, Fused).

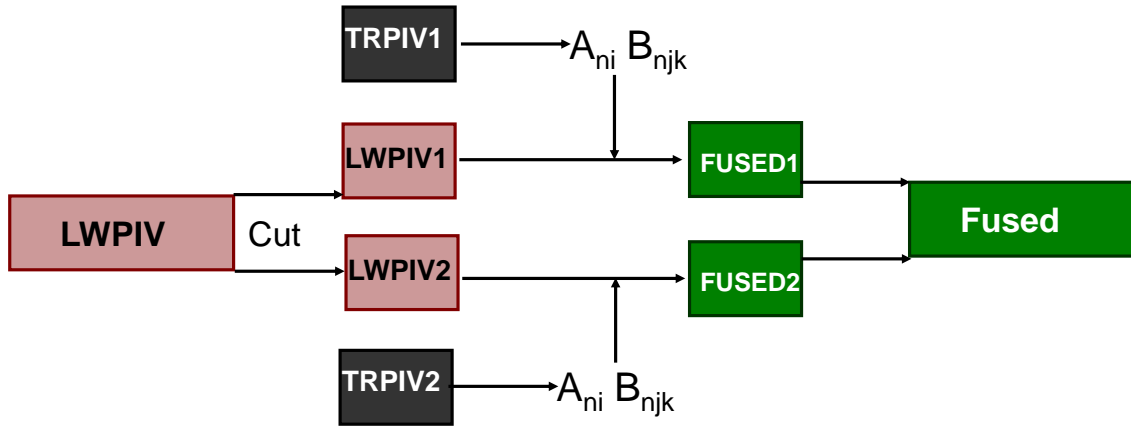


Figure 5.43: Method of Using Stochastic Modeling to Perform Data Fusion using TRPIV and LWPIV

Again before this approach is applied to the LWPIV data set, the technique is validated using the TRPIV data which allows a direct comparison between the actual velocity and the estimated velocity. This is accomplished by using the TRPIV data to generate a stochastic model. The POD coefficients from the TRPIV are then used as initial conditions for the modeling problem. The data is then propagated in time using the model and the estimated coefficients can be directly compared to the actual coefficients.

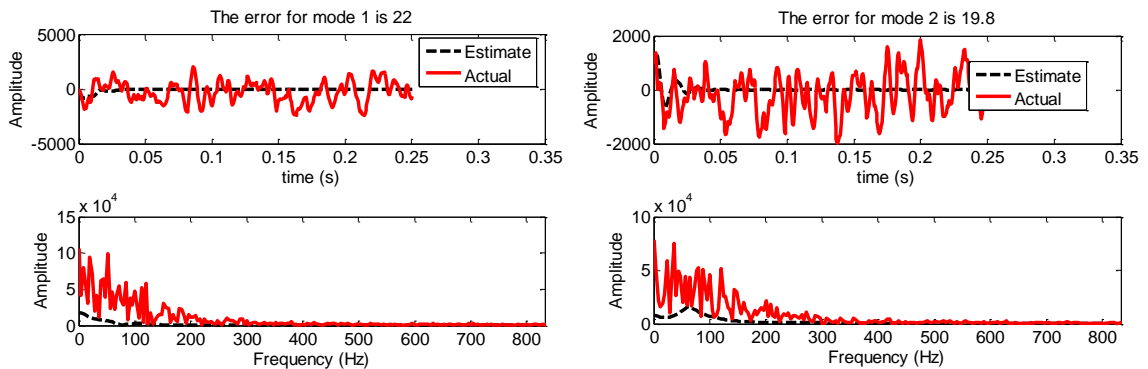


Figure 5.44: Comparison of the Estimated Mode 1 and Mode 2 Using Stochastic Modeling and the Actual Mode 1 and Mode 2

Figure 5.44 shows the actual time dependent POD coefficients and the estimated POD coefficients for mode one and two. One should note that the time of the estimate is 0.25s. This number is not arbitrarily chosen. For the technique to be viable, the estimate must produce reliable results out to 0.25s since the sampling rate of the LWPIV is 4Hz. No velocity information is known until the next snapshot.

Doing this analysis on the TRPIV gives “bad” results. The error in the POD modes are $\mathcal{O}(20\%)$ (The error is calculated using equation 5.5). In the short time (about one to two integral time scales), the estimation is in reasonable agreement with the actual data. After that, the estimation decays to zero. This is not surprising, since many researchers have discovered this fact as discussed in subsection 1.4.4. The important point is the good short time performance of the technique.

5.5.3 Gappy POD

The stochastic modeling method did not work in the long time, so Gappy POD is used as another method of obtaining a large-window time-resolved data set using the LWPIV and TRPIV. The approach to accomplish this is displayed pictorially in figure 5.45. The technique works by using the LWPIV data to obtain the POD modes for a large domain (equivalent to two TRPIV windows in figure 5.45). The TRPIV1 data is placed in a domain that is the same size as the LWPIV with the right half being unknown. A mask vector is created stating that the TRPIV1 data is non-gappy data and the unknown data is said to be a gap, albeit a large gap. This data is used in the Gappy POD procedure outlined in figure 1.6 and the results should give a large window time-resolved velocity field.

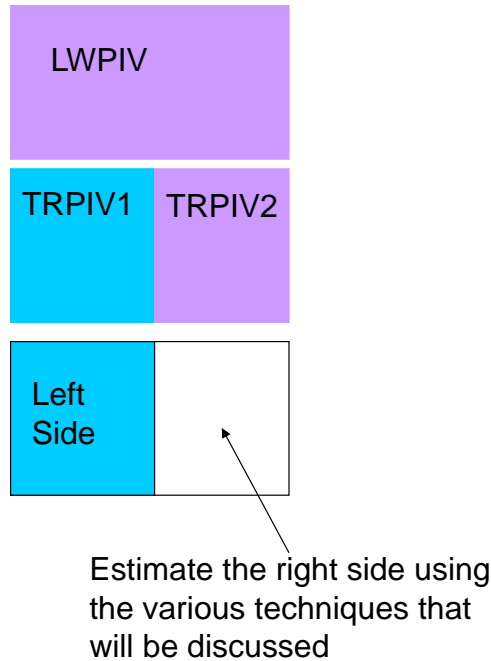


Figure 5.45: *Method of using the TRPIV Data to Verify the Effectiveness of Gappy POD as an Estimation Technique*

This techniques first needs to be verified. To acomplish this, the TRPIV data taken at the 6D-7.5D window is cut in half. This produces three sets of data. The original TRPIV data is similar to the large-window, the left half of the window is similar to the TRPIV1 data, and the right half of the window is similar to the TRPIV2 data. The original, non-cut window is used to calculate the POD modes for the entire window. The left half of the data is then used to estimate the right half of the window using the procedure described previously. The estimated data can now be compared to actual data to see how well the technique works.

Similar to the sLSE and stochastic modeling techniques, the results of the Gappy POD estimation are compared using the POD coefficients. The estimated POD modes in both time and frequency domain have similar shapes to that of the actual coefficients as seen in figure 5.46. This shows the Gappy POD is estimating the largest flow structures. It is found that Gappy POD has reduced the error in the modes from $\mathcal{O}(20\%)$ for the sLSE and stochastic modeling to $\mathcal{O}(15\%)$. So Gappy POD is doing

better than the previous estimation techniques. It should also be noted that the POD coefficients estimated with the Gappy POD match in the frequency domain whereas the sLSE estimates did not capture the low frequency correctly and had many dominate frequency that were not present in the velocity. This is good news especially from a control/modeling standpoint as was stated earlier.

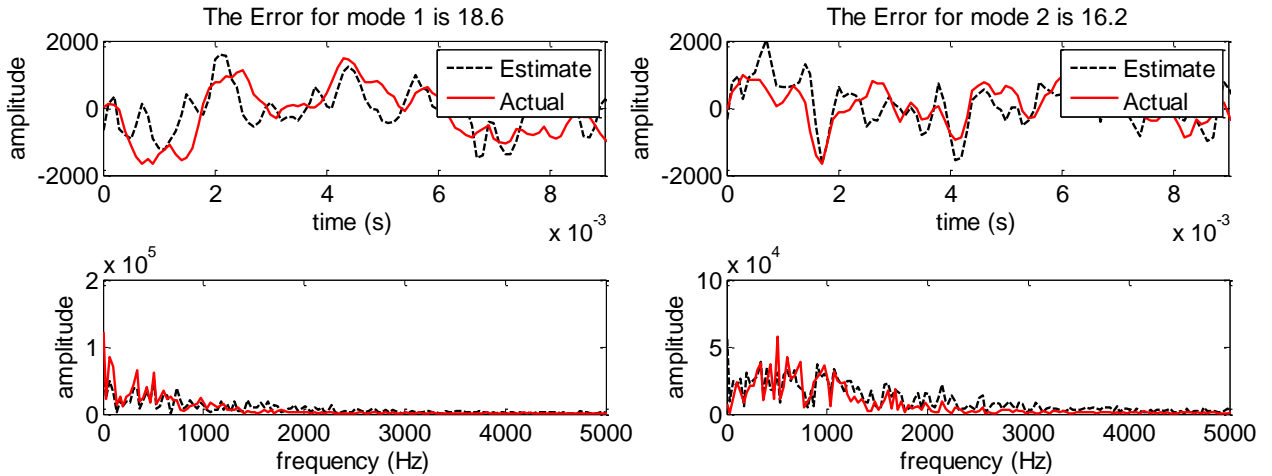


Figure 5.46: Comparison of the Estimated Mode 1 and Mode 2 using Gappy POD and the Actual Mode 1 and Mode 2

Using the POD modes and estimated coefficients, an estimated velocity is generated. The estimated velocity is created using five modes and compared to a five mode reconstruction of the actual velocity.

The error can be quantified in both time and space. The spatial variation in error is shown in figure 5.47 by extracting the estimated velocity and actual velocity along streamwise lines and calculating the RMS error along each line and then averaging the error in time. Looking at the error in space, the error increases further downstream (figure 5.47). This is not surprising; the data being used for the estimation is coming from the upstream side of the estimate. It makes sense that the estimation gets worse further from the data. Another thing to notice is the error is higher near the region of the potential core because the high energy POD modes do not represent the potential

core as well as the shear layer because the shear layer has much more energy than the potential core. This is shown in figure 5.48, which shows the first five POD modes and mode 20. The POD modes do not have much structure near the centerline of the jet until the higher modes are observed. Since the estimation is based on POD modes, it is difficult to estimate the flow where much information is not present. The error in the potential core is similar to the error in other regions of the flow (figure 5.49) if 20 modes are used to estimate the flow. In figure 5.49 one can see a jump in error from $z/D = 6.75$ to $z/D = 7.25$ and then the error levels off. This is caused by the length scale.

One should not expect to estimate the flow with great accuracy after one integral length scale, because after one integral length scale the flow is no longer correlated. Inside one length scale, it is known that the flow is correlated and has statistical knowledge of the points surrounding it. This is why the Gappy POD technique works better with data that is within one length scale than data further away.

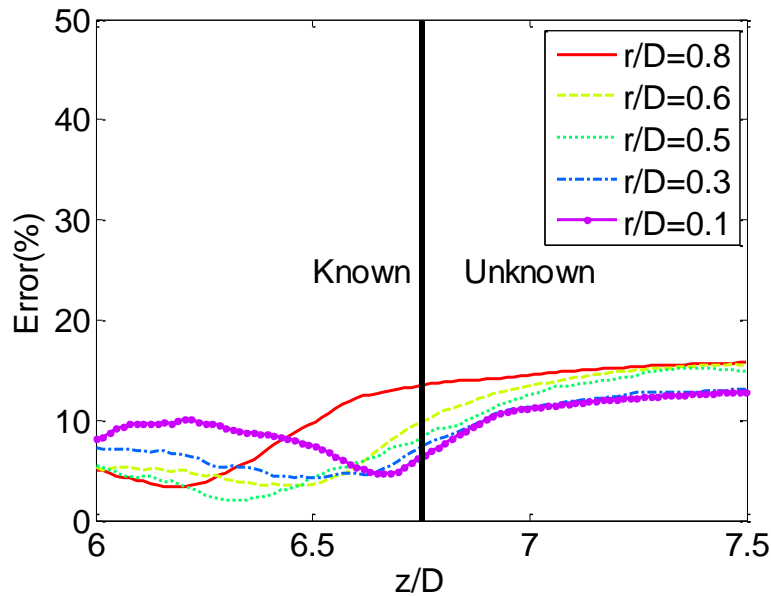


Figure 5.47: *Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach with 5 POD Modes in Space*

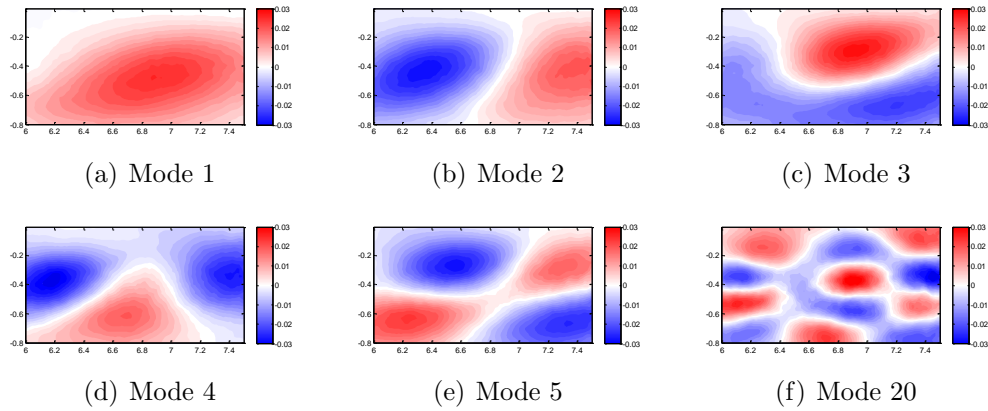


Figure 5.48: *The First 5 POD Modes and Mode 20 for a Mach 0.6 Jet*

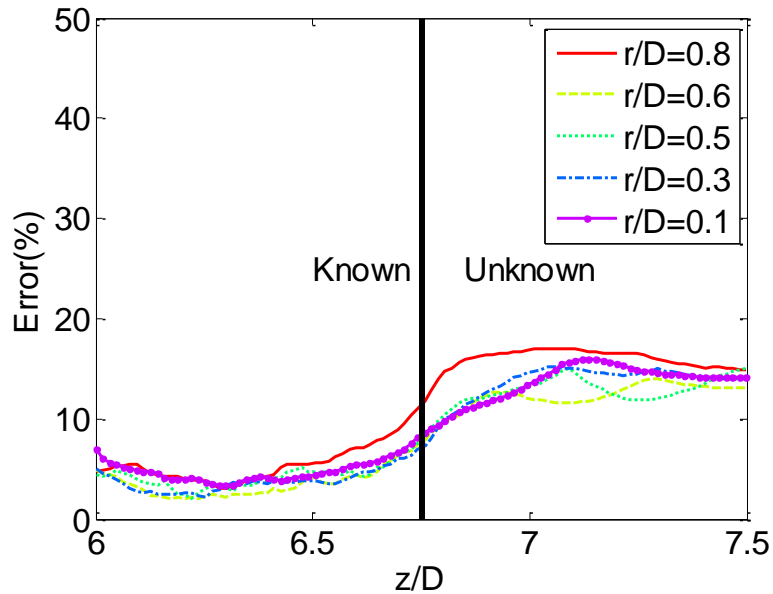


Figure 5.49: *Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach with 20 POD Modes in Space*

In addition, one may notice a slight increase in error in the downstream data when using the 20 mode reconstruction as opposed to the 5 mode reconstruction. This has to do with the spatial POD modes. Looking at the modes (figure 5.48) one can see there are zeros scattered throughout space. Since there are zeros in the spatial modes, any error in the POD coefficient is not accounted for in the velocity reconstruction at

that point in space because it would be multiplied by a zero or a comparatively small number. The error increases slightly when more modes are added because there are modes with large magnitudes in all spatial areas. This is why the error in certain regions is smaller using fewer modes. This is clearly seen in figure 5.47 by looking at the error at $r/D = 0.1, z/D = 6.6$, where a decreased error exists. Looking at the spatial modes (figure 5.48), the modes at that point are small relative to other points in space.

In time, the comparison shows the average error in the velocity is $\mathcal{O}(20\%)$ as seen in figure 5.50. This figure shows the RMS error over space as a function of time. This shows the results from Gappy POD are a huge improvement over the sLSE result. (The better results might be due to using velocity to estimate with Gappy POD instead of using pressure with the sLSE.) However, one should note the error is fluctuating quite a bit. There are 15 fluctuations larger than five percent error per every 100 snapshots. Five percent was chosen as a threshold because that is when the velocity showed a noticeable discontinuity in time. Using more modes, the number of fluctuations decrease as is seen in figure 5.59, which compares the number of fluctuations in the RMS error as the number of modes used in the estimation and reconstruction increases.

Despite the short time errors (fluctuations), the result has “good” long time behavior because data is always injected into the estimate (similar to a Kalman filtering approach) unlike the stochastic modeling approach. This produces an accurate estimate of velocity that can be used for long time since the estimate does not degrade after a few time scales.

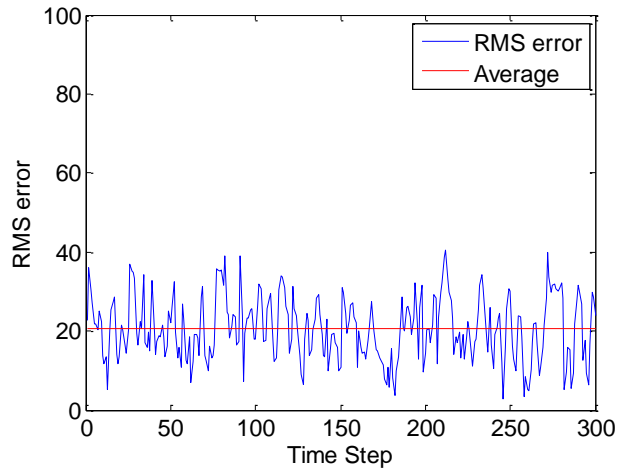


Figure 5.50: *Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach with 5 POD Modes in Time*

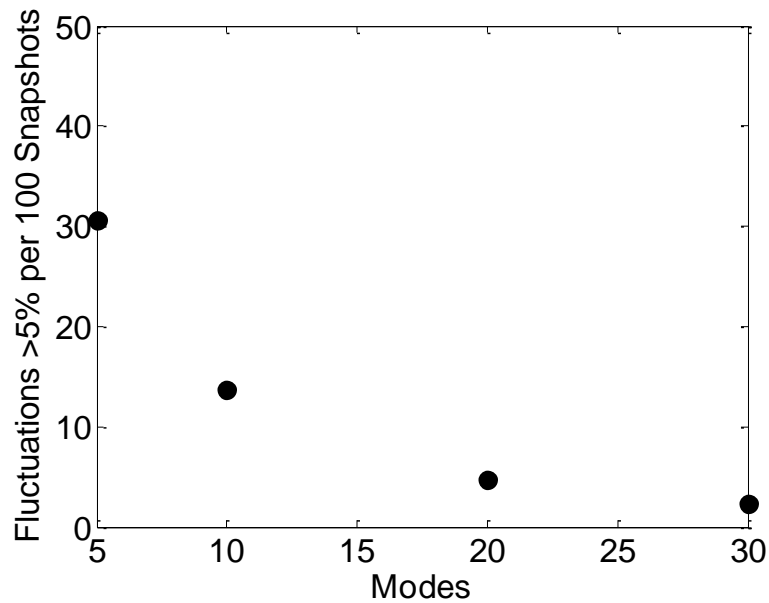


Figure 5.51: *Fluctuations Larger than 5% per 100 Snapshots Using the Gappy POD Approach*

The Gappy POD is producing results with some errors; one would expected Gappy POD to give the correct results, given a correct basis. If all of the POD modes are used, the flow should be reconstructed exactly. In this case, the basis is coming

directly from the same data that is being estimated, which suggests the repaired data should be the same as the original data, within the rounding of error of the computer. If more modes are used the estimation error decreases. In fact if all of the modes are used the estimation gives the original velocity. Due to computational constraints, this cannot be tested for the entire data set. To test this, 50 snapshots are extracted from the entire ensemble and the Gappy POD repair process was repeated. It was found that if all 50 modes were used in the estimation process the estimation produced the actual data (figure 5.52).

With this in mind, the estimation should be done with all of the modes. However, two issues prevent this. One is practicality and the other is imperfections in the data collection process. Using all of the modes is impractical because of the computational resources required. For the 50 snapshot case all modes could be used and the estimation process takes about 5 minutes. However, the time to complete the process is proportional to the number of modes used squared. In the case of the entire ensemble (8623 snapshots), the process would take about 6 months to repair 50 snapshots based on extrapolation of figure 5.53, which shows how much time is required to do the Gappy POD procedure for different number of modes. This is impractical.

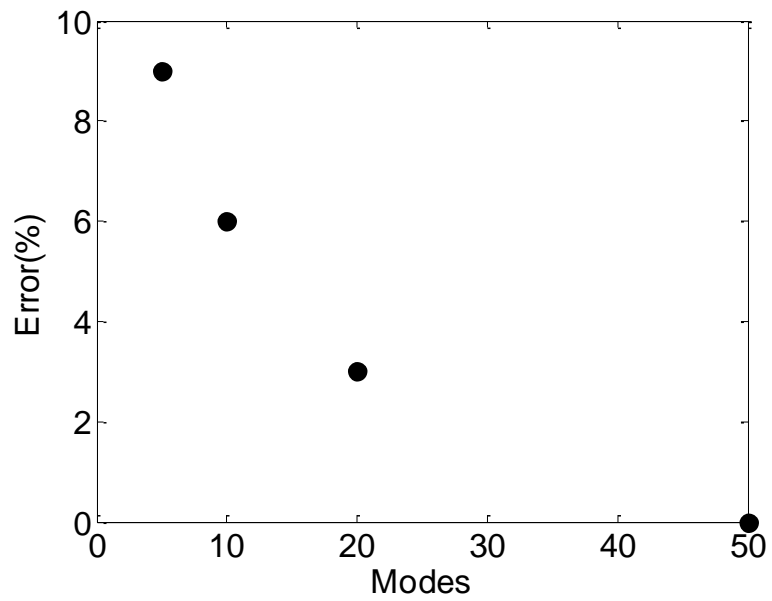


Figure 5.52: *Convergence of the Gappy POD Estimation Technique for the Jet Data using Only 50 Snapshots*

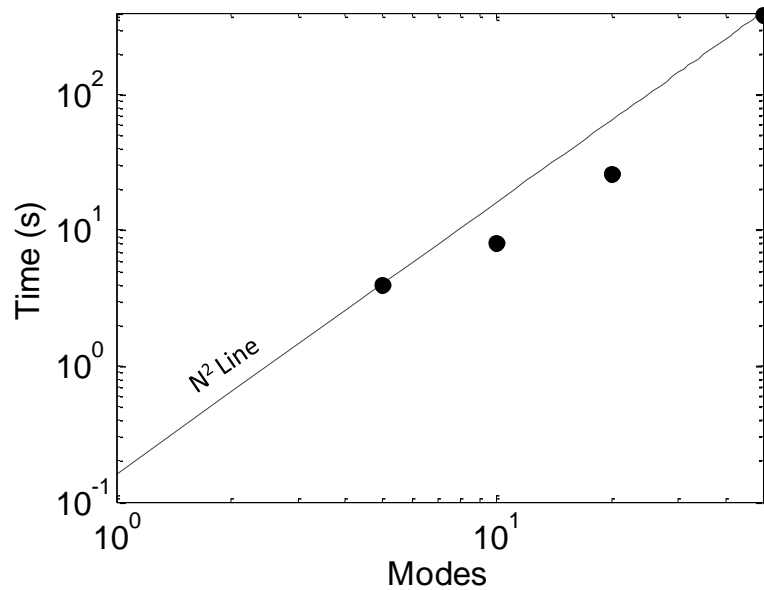


Figure 5.53: *Computational Time to Run the Gappy POD Estimation Technique for the Jet Data using Only 50 Snapshots*

In addition, the LWPIV and the TRPIV may have some inconsistencies between

them. The techniques are similar but data was taken at different times. The placement of the cameras and laser sheet may be a little different, the jet speed may be slightly different, and other differences may be present. This means the modes from the LWPIV and TRPIV can be different, especially the higher order modes. In general, the lower modes are similar because they represent the large scale structures which are less susceptible to errors.

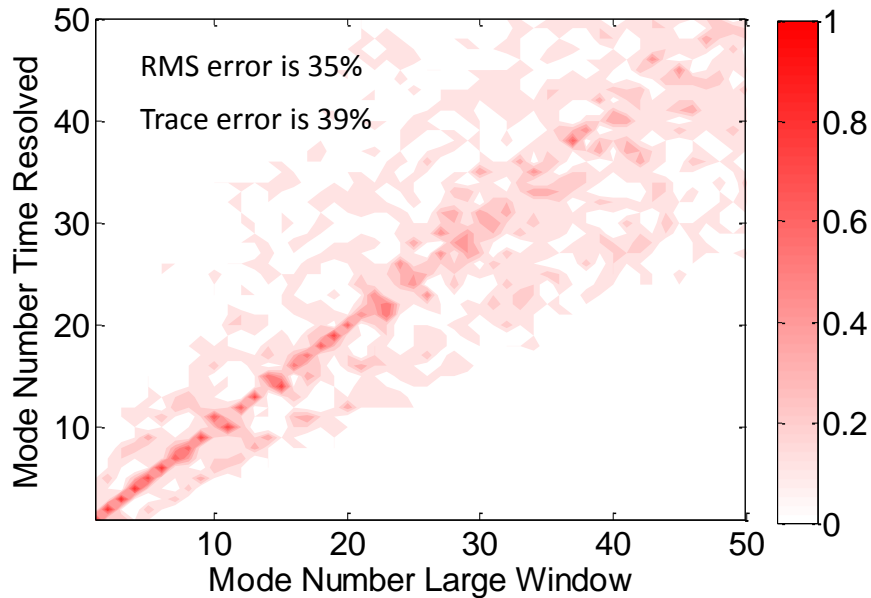


Figure 5.54: *Correlation Between the POD Modes Calculated from the Large-Window PIV and the Time-Resolved PIV*

Figure 5.54 shows the correlation between the modes of the TRPIV and the LW-PIV. The data for the LWPIV was extracted from the entire window such that it was the same size spatially as the TRPIV. To mitigate some error, the LWPIV was shifted in space so it would match the TRPIV data because the experiments were performed at different times. The optimal shift is actually zero, meaning the TRPIV and LWPIV data were taken in the same spatial location. The modes are in “good” agreement ($\sim 90\%$ correlation) for the first 10 modes, which account for about 60% of the energy. Modes higher than 10 began to lose correlation and after mode 25 the

correlation was minimal. This shows that the two different experiments match well for the large scale physics, but there are difference in the smaller scales. This could be due to coarser resolution in the LWPIV system, slight errors in the location of the windows, or convergence issues due to the number of snapshots used and different ambient conditions.

If the modes are not the same, then some of the information in the LWPIV modes will not match with the TRPIV data, which means the snapshots cannot be written as a linear combination of the modes. This implies that the error will not improve if more modes are used in the estimation.

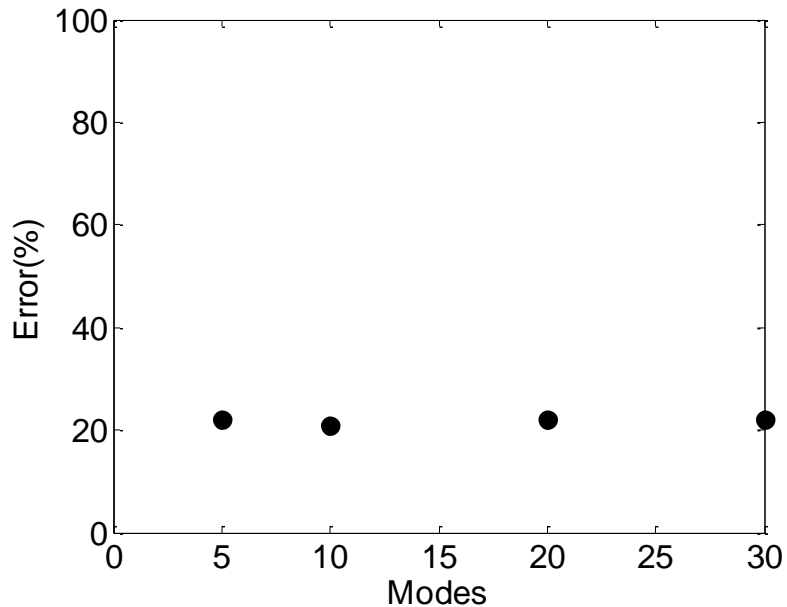


Figure 5.55: *Error Between the Estimated Velocity using LWPIV Modes and the Actual Velocity using Differing Number of Modes*

In figure 5.55 the error does not improve by using more modes. This is because the LWPIV modes are different from the TRPIV modes as seen in figure 5.54. Adding more modes gives information about the finer structures in the flow, but the modes being used are not the right modes. Eventually using more modes should cause an increase in error since the correlation between the LWPIV modes and TRPIV modes

is decreasing. This is not shown here because the computational resources did not allow for the estimation with more than 30 modes.

5.5.4 Dynamic Gappy POD

The previous two sections showed that stochastic modeling worked well in the short time and Gappy POD worked well in the long time. Using the idea of feeding back information into the model the Dynamic Gappy POD (DGP) technique was conceived. The Gappy POD technique is used for the long time (keep the model from deviating) and the stochastic modeling is used for the short time (correct for the fluctuations in Gappy POD). The two techniques are fused together simply by taking a linear combination of the two techniques. Basically DGP seeks to minimize the error between the results from the stochastic model and the results from Gappy POD as shown below:

$$\min_{\hat{a}_n} \left[(g - \tilde{g}^2) + \sum_{n=1}^N (\hat{a}'_n - a_n^{LSE})^2 \right] \quad (5.7)$$

such that

$$\tilde{g} = \sum_{n=1}^N \hat{a}_n \phi_n \quad (5.8)$$

$$\hat{a}'_n = \sum_{i=1}^N \hat{a}_i \phi_i^h \phi'_n \quad (5.9)$$

where ϕ_i^h are the eigenfunctions for the large window evaluated in the region where the velocity is unknown and ϕ'_n are the eigenfunctions from the small window in the region where the velocity is unknown. These two different eigenfunctions are needed due to the windowing effects. In other words, if the modes are calculated from the large window data and then compared with the small window they will be different. A schematic of how DGP work is shown below in figure 5.56

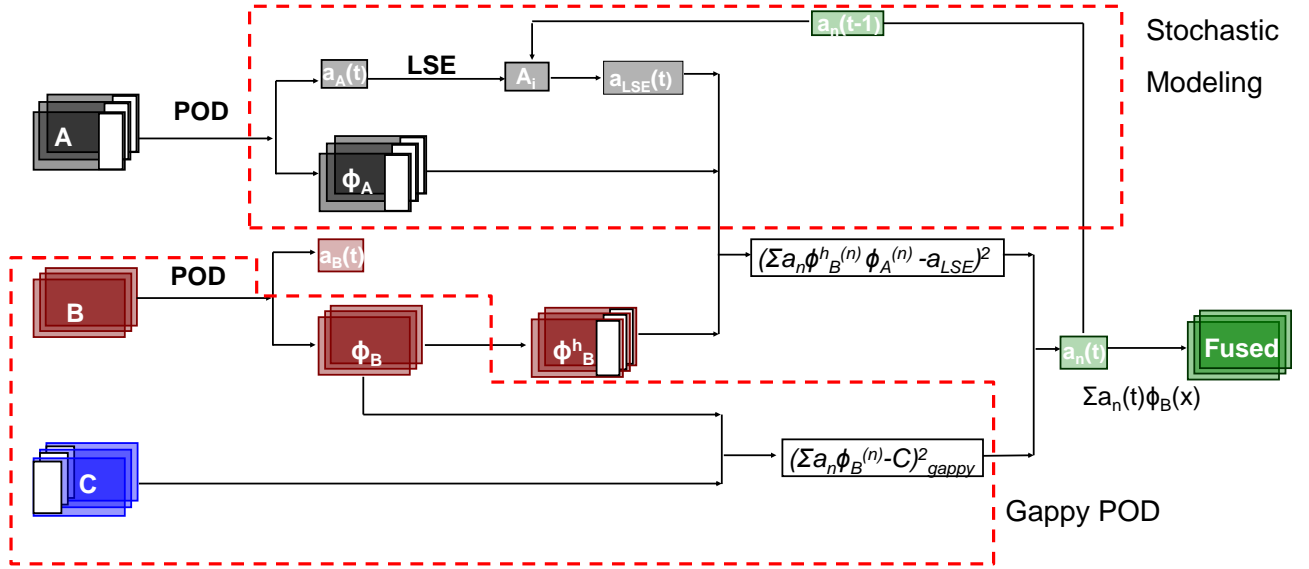


Figure 5.56: *Diagram for the Dynamic Gappy POD Technique*

Dynamic Gappy POD works first by calculating the POD modes of “A” (the black data), which gives “ $a_A(t)$ ” and “ Φ_a ”. The POD coefficients “ $a_a(t)$ ” are then used to generate a stochastic model for the right side of the data. POD is then done on “B” (the red data). This gives the POD modes for the entire domain (“ Φ_B ”). The right half of the mode (“ Φ_B^h ”) is extracted and will be used to compare the data on the left from the right half of the window. The estimated POD coefficient (“ $a_n(t)$ ”) can be estimated two ways. One way is to use data “C” (the blue box) and Gappy POD to minimize the least square error between the known locations of velocity (“C”) and the estimated velocity (“ $\sum a_n(t)\Phi_B(x)$ ”), which gives the value for “ a_n ”. The other way is to project the right half of the estimated velocity onto “ Φ_A ”. This is done using equation 5.9. The projected value is then compared to the modeled value of the POD coefficient (“ a_{LSE} ”) by minimizing the least square error. These two methods are fused together solving the minimization problem simultaneously as shown in equation 5.7.

The DGP technique will be applied to the TRPIV and LWPIV data set in a similar manner that the Gappy POD is applied. The one difference is the TRPIV

data is used to generate a stochastic model. The technique is applied to one TRPIV window that is cut in half to validate the technique. The left side of the window is used in the Gappy POD part of the algorithm and the right side is used to generate a stochastic model.

The DGP technique decreased the error in the POD modes to $\mathcal{O}(10\%)$. This is a decrease from the $\mathcal{O}(15\%)$ error for Gappy POD and $\mathcal{O}(20\%)$ for sLSE and stochastic modeling. The DGP is also capturing the correct frequency just as the Gappy POD did. Again the time and frequency traces have the correct shapes and trends.

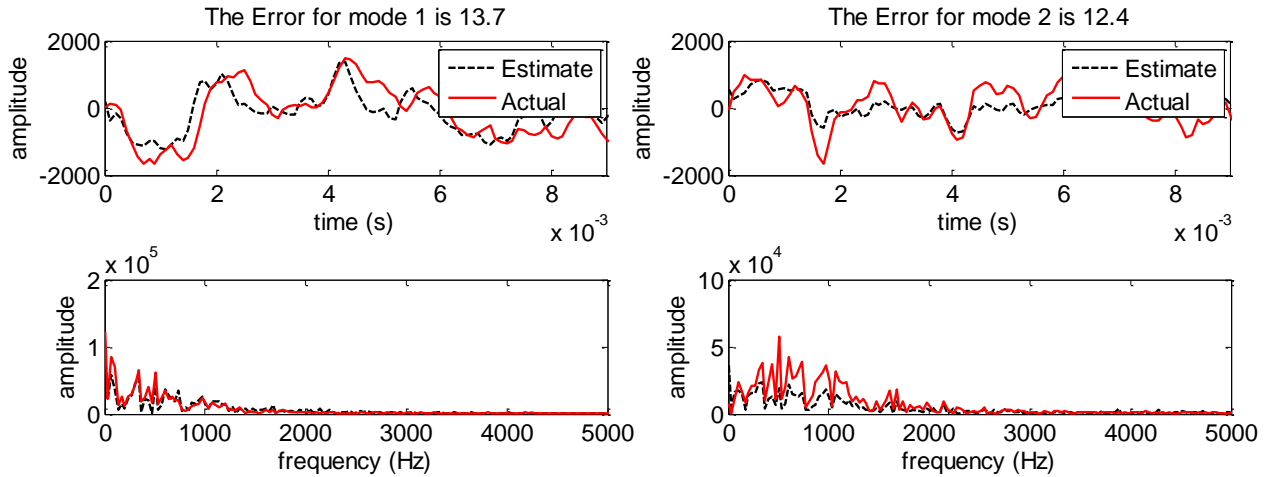


Figure 5.57: Comparison of the Estimated Mode 1 and Mode 2 using Dynamic Gappy POD and the Actual Mode 1 and Mode 2

The velocity is reconstructed using the estimated POD coefficients and the first five modes. The actual low-dimensional velocity is compared with the estimated velocity and a time trace of the error is generated (figure 5.58). The average error is $\mathcal{O}(20\%)$. This is not a significant improvement over Gappy POD in terms of the mean error. However, the advantage of the DGP technique is the reduction of the short time fluctuations in the velocity. In this estimation there are 10 fluctuations greater than five percent error per 100 snapshots. This is a 33% improvement over the Gappy POD technique.

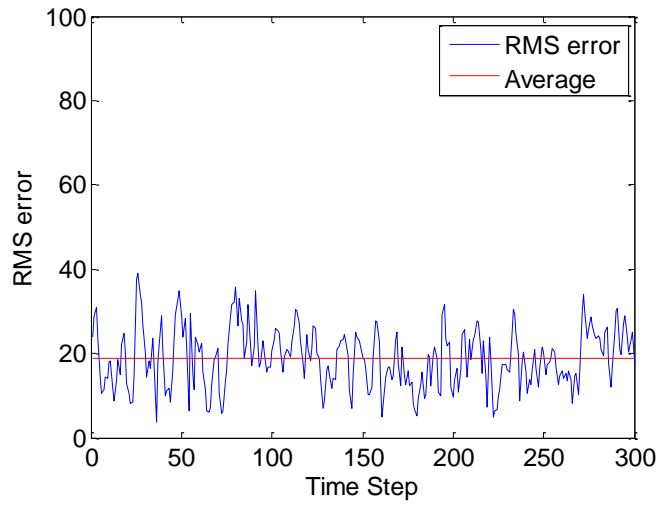


Figure 5.58: *Error Between the Actual Velocity and the Estimated Velocity using the Dynamic Gappy POD Approach with 5 POD Modes*

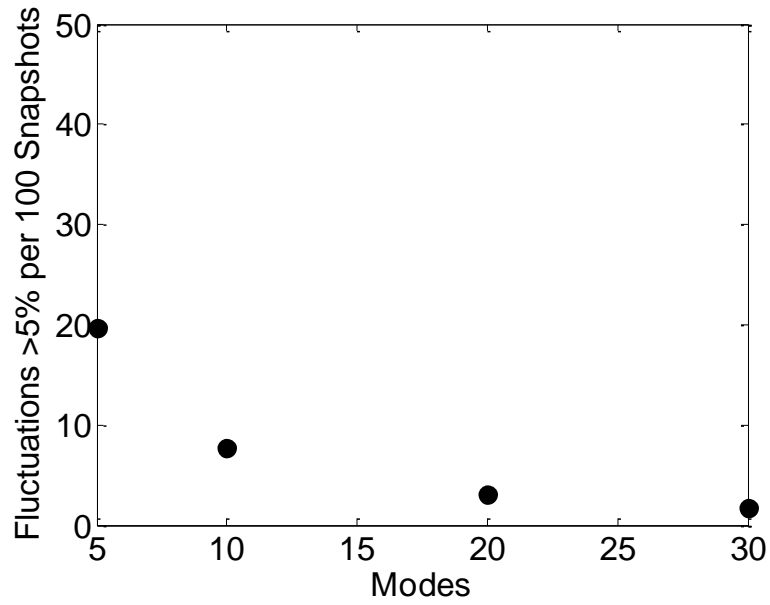


Figure 5.59: *Fluctuations Larger than 5% per 100 Snapshots using the Dynamic Gappy POD Approach*

5.5.5 Comparison of Results

The results of the different fusion techniques applied to the TRPIV jet data are compared and show the DGP is the best candidate to obtain a large-window time-resolved data set given the available data. The sLSE technique resulted in high errors due to low correlations between the pressure and the velocity. The low correlation may be due to the set up of the experiment and the PIV and pressure being sampled in different planes. Moving the pressure sensors closer to the PIV laser sheet may improve the correlations and the results of the sLSE estimation. In addition, perhaps velocity data from the TRPIV data could be used to do this estimation in the future.

The stochastic modeling technique was able to capture the short time behavior of the flow. After a few integral time scales the model did not give correct results as is typical with models of this type. To improve the technique data is usually injected into the model, typically by using a Kalman Filter. However, data is only available every 0.25 seconds due to the sampling rate of the LWPIV, which is much larger than an integral time scale.

Gappy POD is used to estimate the missing portion of a data set using the POD modes (statistics). This technique provides data at every time step and in the ideal world gives the exact answer. However, doing so requires using all of the POD modes and performing Gappy POD using all of the modes is computationally expensive. The modes between the LWPIV and TRPIV also have inconsistencies that lead to incorrect estimations as well.

DGP is a fused technique that takes advantage of the long time behavior of Gappy POD and the short time behavior of the stochastic modeling. The technique modestly reduces the errors but reduces the short time fluctuations in the flow significantly as shown in figure 5.60. This makes the DGP technique attractive for fusing the TRPIV and LWPIV data sets. In figure 5.60 only the error of Gappy POD and Dynamic Gappy POD are shown. Spectral LSE is not compared here because the comparison may be unfair to the sLSE method since it used pressure data and the other techniques

used velocity data to do the estimation.

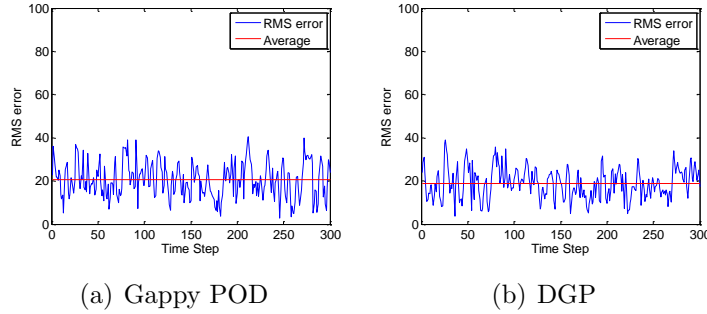


Figure 5.60: *Error for Different Estimation Techniques for a Mach 0.6 Jet Flow*

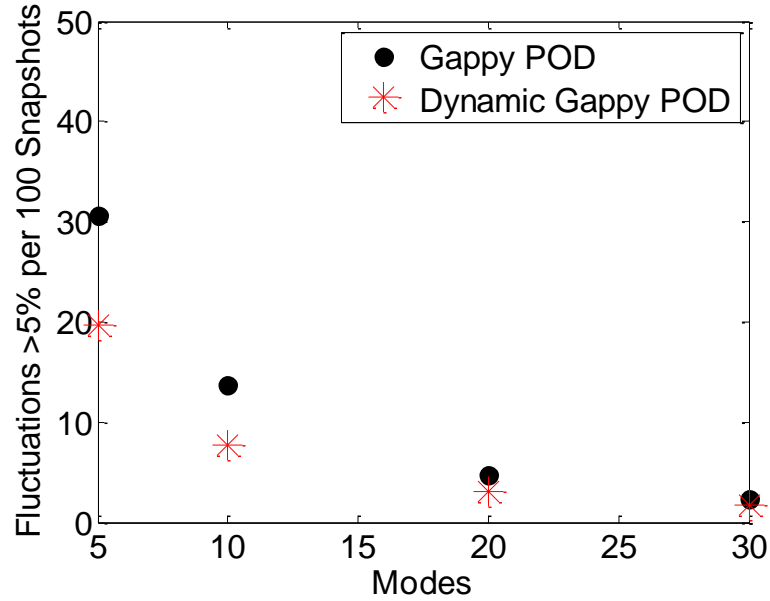


Figure 5.61: *Comparison of Fluctuations Larger than 5% per 100 Snapshots using the Gappy POD Approach and the Dynamic Gappy POD Approach*

One can see the fluctuations reduce as the number of modes are increased (figure 5.61). However as the number of modes increase, smaller fluctuations in the error correspond to discontinuities in the velocity field because the higher energy modes have finer structures. The finer structures are more sensitive and thus discontinuities in the finer structures lead to smaller fluctuations in the error. This means that even

though the Gappy POD and DGP converge towards each with increasing number of modes, the criterion to define the fluctuation should change if more modes are used.

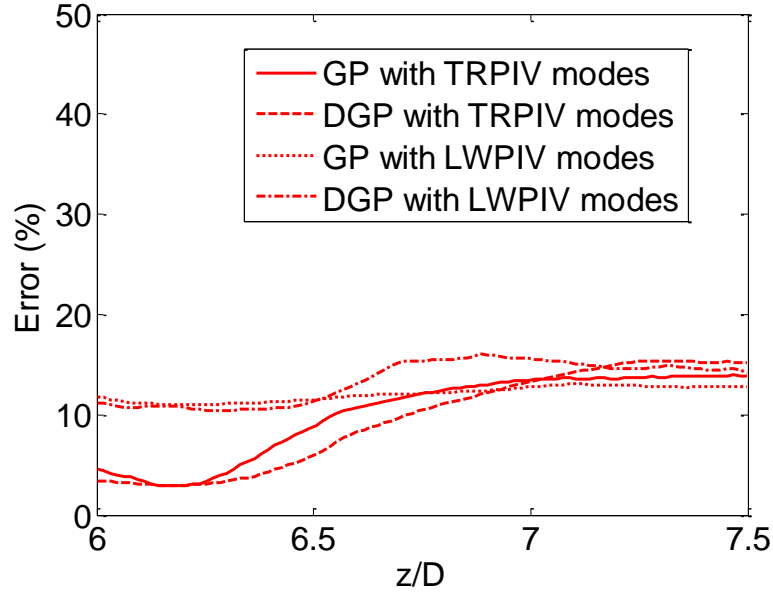


Figure 5.62: Comparison of Error Between the Actual Velocity and the Estimated Velocity using the Gappy POD Approach and the Dynamic Gappy POD Approach with 5 POD Modes in Space at $r/D=0.8$

Figure 5.62 shows a plot of the spatial dependence of the RMS error between the actual data and the estimated data for the Dynamic Gappy POD and Gappy POD using POD modes from the LWPIV and the TRPIV. This shows a few interesting results. The first of which is using the actual modes give better results, which makes sense because the modes are coming directly from the data. However, in the real problem, it is not possible to calculate the real modes. Using the modes from the LWPIV, gives results with slightly more error especially in the known portion of the estimate. This is because the modes are different and the flow is not reconstructed as efficiently using the different modes.

The biggest thing to note is the Dynamic Gappy POD technique is actually doing worse than Gappy POD away from the actual data. This is because the model corrupts the Gappy POD process. Gappy POD produces the correct data if all of the

correct modes are used, but Dynamic Gappy POD does not. The model is relied on more heavily far from the known data because that is where more fluctuations are occurring. Figure 5.62 is also an average of the error in time and does not account for the large fluctuations that are present in the Gappy POD techniques. Looking at the fluctuations, (figure 5.61) one can see the Gappy POD technique is prone to large small-time fluctuations despite the lower average error.

5.6 Summary of Data Fusion on a High Speed Jet Flow

A new technique, Fused POD that is inspired by Gappy POD was used to fuse PIV data and the theoretical knowledge that the jet is axisymmetric. This allowed for estimation of the velocity in the occluded region caused by pressure sensors. A reasonable estimate of the velocity data allows one to have the pressure data and a non-obstructed PIV window. Gappy POD would not be applicable in this situation because the error is located at one point in space for all snapshots. Gappy POD requires statistical knowledge of the flow which is not available in this situation without obtaining it from another source.

The idea of Fused POD was also applied to a problem where iterative Gappy POD was applicable. However, using the iterative approach would take months to complete. Using the idea of Fused POD the time required to repair the jet data took on the order of hours. There was a tradeoff though; Fused POD did not give the same result as iterative Gappy POD, which should be correct in the limit of using all modes and infinite iterations. The time savings decidedly were worth the slight error.

Methods of estimating a time-resolved large-window PIV data set were discussed. The methods included sLSE, stochastic modeling, Gappy POD and Dynamic Gappy POD. LWPIV, TRPIV, and pressure were fused together to accomplish the task. Because the TRPIV data was available, the estimates could be directly compared

with the actual data allowing for one to determine the accuracy of the methods. It was found that the sLSE did not work as well due to the low correlations between the pressure and the velocity. Stochastic modeling also was unable to predict the velocity due to poor long time behaviors with the model. Gappy POD was able to predict the velocity for long time periods, but was unable to capture the short time flow physics. Dynamic Gappy POD was able to take advantage of the stochastic modeling's short time behavior and fuse it with the long time behavior of Gappy POD thus creating a better estimation tool and a better estimate of the data.

Chapter 6

Concluding Remarks

In this chapter the fusion work is first summarized. All of the techniques used to perform the fusion are briefly discussed and it is shown how all of the techniques fit together. Then the different test cases where data fusion was applied are discussed briefly. The summary of this work is followed by the conclusions from this work and the chapter ends with suggestions for future work.

6.1 Summary

This work showed the usefulness and power of data fusion and technique fusion for fluid dynamic data and the importance of having a framework that is capable of performing the analysis tools and managing data efficiently. The techniques used in this study are POD, LSE, and Wavelet fusion. The connection between the fused techniques are shown in figure [6.1](#).

Gappy POD is fused with Wavelet fusion to create Fused POD, which looks to extend Gappy POD to problems with holes in the same location in all of the data as discussed in [subsection 4.6.1](#) and [section 5.4](#). Dynamic Gappy POD is the fusion of LSE and Gappy POD, which is used to improve the short time behavior of Gappy POD and long time behavior of stochastic modeling ([subsection 5.5.4](#) and

section 5.5). Wavy POD is the idea to fuse POD and wavelets, which could allow the fusion of high frequency information and low frequency information for two data sets unable to be synchronized in time. The last fused technique mLSE was developed by Bonnet *et al.* [48] and Taylor *et al.* [49], which combine POD and LSE to increase the computational efficiency and allow for low-dimensional estimates.

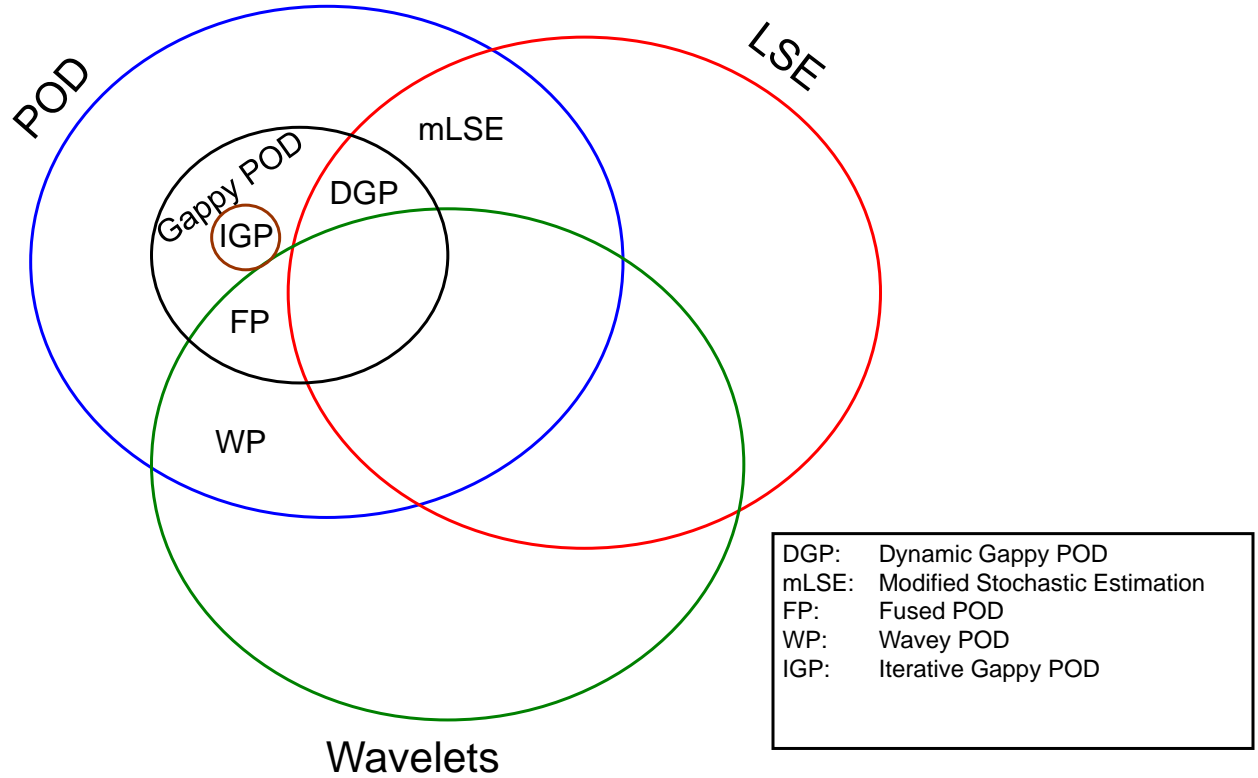


Figure 6.1: *Overview of the Fused Techniques*

Throughout the document three different cases are studied to derive, test, and understand the various techniques described herein. These include synthetic data, three-dimensional (one-time, two-space) airfoil data, and three-dimensional (one-time, two-space) jet data with the ultimate goal of using data fusion on the jet data. The diagrams below summarize the fusion techniques that are applied to each case.

The first case was a synthetic case (figure 6.2(a)) meant to represent CFD and PIV, where the CFD was represented by a diffused signal and the PIV was represented by

a noisy, down-sampled signal. These were supposed to represent some of the different limitations associated with PIV and CFD and determine if data fusion is a viable idea to compensate for the limitations. The noise in the PIV-inspired signal was removed by using wavelet fusion to fuse the CFD and PIV. This also reduced the effects of diffusion in the CFD, effectively creating a more resolved PIV-inspired signal. An early variant of Fused POD was applied to the synthetic case to remove holes in the PIV-inspired data using the CFD-inspired data, which filled in the holes better than just filling in with the CFD-inspired data. Modified LSE was applied to the problem as well, but did not produce as good of results as the other techniques.

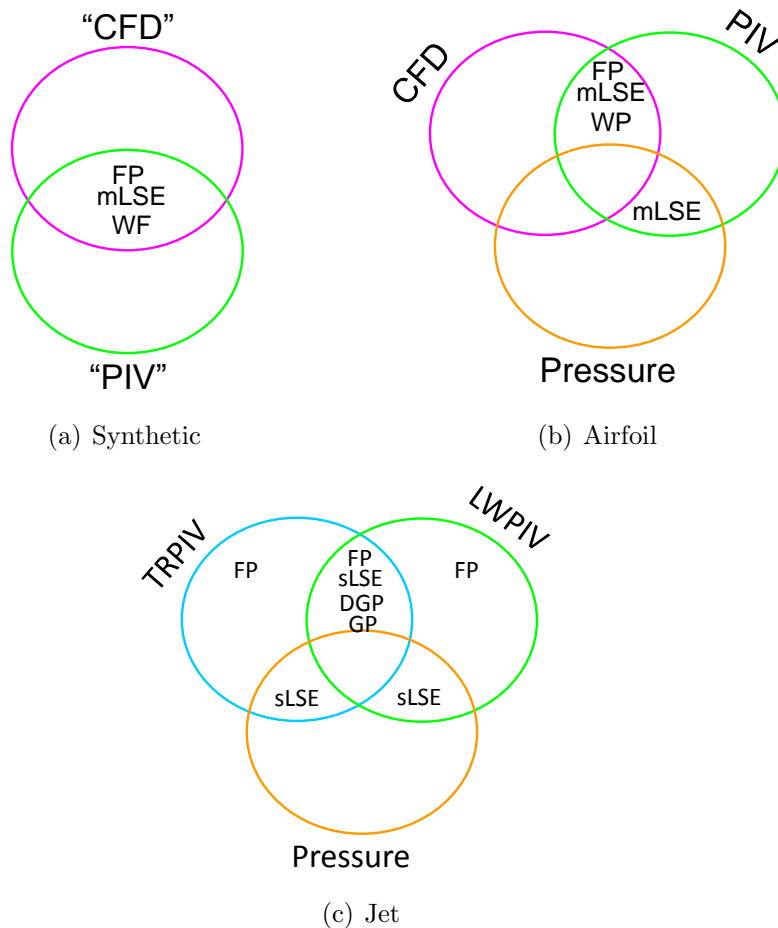


Figure 6.2: *Overview of the Fusion Test Cases*

The second case is a stationary airfoil in a stalled configuration, which produces

unsteady flow in the wake region. This case included LWPIV, pressure measurements along the surface of the airfoil [24], and a CFD simulation of the airfoil. Fused POD was used to fill an artificial hole in the PIV data meant to represent an occlusion, reflection or particle drop out. The CFD was used to generate the POD modes used with Fused POD. This shows the holes could be filled with reasonable accuracy, better than inserting the CFD solution into the hole, and if the CFD was improved the results of the Fused POD improved as well. Spectral LSE was applied to the problem as well to fuse pressure and LWPIV, which was done previously by Ausseur [67]. Similar results were shown, but the results could not be validated due to a lack of time dependent velocity measurements. A similar process was used to fuse velocity signals from the CFD and the LWPIV, which gave results that made physical sense but again were not validated.

The jet data which included TRPIV (Low *et al.*[99]), LWPIV (Berger *et al.* [107]), and pressure measurements is the final case used in this study. Fused POD was used to fill sensor occlusions, reflections, and erroneous data due to particle drop out. An artificial hole was cut in the data to understand how accurate the technique is for the jet data by comparing the Fused POD results to the experimental data. Spectral LSE, stochastic modeling, Gappy POD, and Dynamic Gappy POD were applied to the jet data to obtain a large-window, time-resolved data set, which showed the DGP produced the best results by reducing the small time fluctuations. The estimations were able to be directly compared to the actual data since time-resolved data was available.

6.2 Conclusions

Fused POD, an extension of Gappy POD, is able to repair data that traditional Gappy POD is unable to repair. Gappy POD requires either enough non-corrupt snapshots to perform POD or snapshots that have missing data randomly distributed in time

and space such that the statistics can be calculated at all points. Fused POD can handle cases where data is missing at one location in space in all snapshots, something that Gappy POD cannot do since statistical information is not available for the missing data. The missing data needs to come from another source. Fused POD was demonstrated in the LWPIV jet data for which data is missing due to a pressure sensor obstructing the field of view. These results are shown in [section 5.4](#) for the high-speed jet.

Fused POD can repair gappy data orders of magnitude faster than Gappy POD. The Fused POD and Gappy POD technique are used on the TRPIV jet data. The snapshots had missing “bad” vectors scattered in space and time in the bottom half of the plane. Iterative Gappy POD was used to estimate the “bad” data for a subset of the data. The repaired data was then used to calculate the POD modes for the entire data set, which was then repaired using the modes and the Fused POD method. Using Gappy POD on the entire data set could have been done, but would have taken on the order of months to complete. Using Fused POD as an alternative, the analysis was done on the order of hours.

Fusing Gappy POD and LSE creates a technique (Dynamic Gappy POD) that estimates the time dependence of a flow field with about 20% fewer fluctuations in error than either technique alone. Dynamic Gappy POD looks to take advantage of the short time behavior of stochastic modeling and the long time behavior of Gappy POD. This is used to stitch time-resolved windows together that were not sampled simultaneously, creating a time-resolved data set with a larger inspection region. This was attempted using LSE, sLSE, Gappy POD, and Dynamic Gappy POD. The Dynamic Gappy POD method worked the best in terms of predicting the flow. This method uses both data fusion and technique fusion. The Dynamic Gappy POD method is able to produce an estimate of the velocity field closer to the original than any of the other techniques.

A framework is useful for tracking data and performing data fusion efficiently.

The data for the fusion process required many manipulations such as extraction, masking, interpolation, etc, which created many new data sets. The vast amount of data needed to be tracked, or it could get lost and the fusion process would become more difficult. The framework is capable of tracking the data in an efficient manner and allowed for fusion to be accomplished efficiently.

The framework has allowed researchers to process twice as much data. Using a framework has allowed researchers to look at larger, more diverse data sets, such as multiple off-center plane windows to understand how a jet evolves away from the centerline. The framework made comparisons between PIV data taken in 2011 and LDA data taken in 2005 less time consuming. The framework allows the comparison of multiple data sets with less man power, allowing more data to be used, ultimately leading to better understanding of the data and physics.

The current study shows fusion can increase the usefulness of fluid dynamic data as well as decrease the time required to process the data. The study also shows a framework which tracks data, allows data manipulations, and allows multiple data sets to be viewed at a time is imperative to performing data fusion and analyzing fluid dynamic data, due to the vast quantity of data that is collected and created.

6.3 Suggested Future Work

Data fusion was performed on some shock tube data as stated throughout the document, but with very poor results that are shown in [Appendix A](#). Using different techniques may prove useful in helping to understand the data better. Moving forward, one can look at the effect of uncertainty in the initial pressure ratio. It is very likely that the pressure was not read or even calibrated correctly. Using the model, theory, data, and perhaps something similar to a Kalman filter, the error in the pressure ratio can be calculated, then used in the theory and model to produce results similar to the experiment.

An idea that combines discrete wavelet fusion and POD that has not been present in this paper has been called Wavy POD. To increase the resolution of an image, one can use the wavelet fusion technique. However, to do the wavelet fusion technique, two images are needed, one that has the high resolution information and another with low resolution. In the case of fluid dynamics, a high resolution data set and a low resolution data set with a larger spatial domain are obtained. More than likely, the two data sets are acquired separately, meaning the wavelet fusion technique does not apply. However, perhaps the wavelet fusion could be done in POD space. The process would be to break the large window data into smaller windows that match the high resolution images. POD would then be performed on the data. The POD modes would be fused using the wavelet fusion process. This would effectively increase the resolution of each POD mode. Then the flow field could be reconstructed using the POD modes and POD coefficients. This process could be repeated for each of the small window data sets. This technique could be useful in the case of trying to get LWPIV with one camera while still resolving the flow spatially. However, this technique has not been studied sufficiently to determine its effectiveness. This idea is shown in figure 6.3. Data set “A” (the black box) represents a large window of data and “B” (pink box) and “C” (blue box) represent two smaller windows with more resolution taken at different locations and different times. “B” and “C” are both in the domain of “A” though. Data set “A” is cut to match the domains of the “B” and “C” producing “A_B” and “A_C”. POD is then performed on the cut data sets and “B” and “C”. The POD modes are then fused together using the wavelet fusion process discussed in subsection 1.4.6. This produces two new modes “ Φ_{FB} ” and “ Φ_{FC} ”, the velocity is reconstructed using the POD coefficients from “A_B” and “A_C” producing the fused velocities “F_B” and “F_C”. These two velocity fields are then stitched together to obtain the full velocity field “F”.

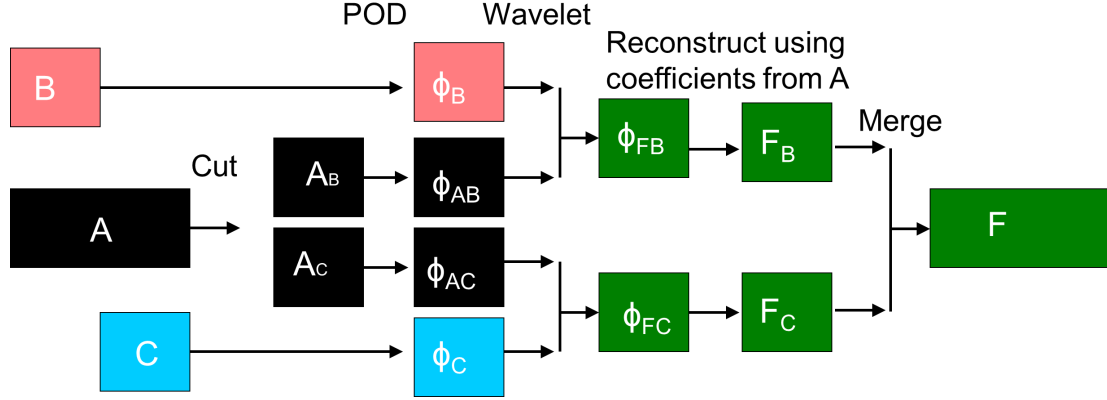


Figure 6.3: *Wavy POD*

The idea for Wavy POD came from the work done on the synthetic data in [chapter 3](#), which used wavelet fusion to increase the resolution of data that had been smoothed by adding diffusion. Using the wavelet fusion, the resolution of the smooth data was increased and noise was removed from a noisy filter. The data used in the test was synthetic and periodic and could be synchronized in time. However, the flow is real data, and in general cannot be synchronized in time with another data set. However, in the POD domain, the modes may be able to be synchronized and some of the high frequency, low energy modes could be merged together, which may increase the resolution of the modes and the velocity after reconstruction.

Another suggestion for future work is to incorporate pressure into the DGP technique. Data fusion work in other fields have shown using more sensors, even “bad” sensors increase the accuracy of the estimate. If pressure could be included into the technique, the accuracy of the technique may increase, especially in areas near the pressure sensor where the velocity-pressure correlation is highest.

In the spirit of using the pressure sensors, the near-field pressure sensors should be placed in the same azimuthal plane as the PIV to increase the correlation. The azimuthal displacement coupled with the radial displacement decreases the correlation strength; increasing the correlation strength may improve the sLSE results, and make using the pressure signals in the DGP more attractive.

In addition, a lot of this work was done using the 2011 experiments from Low *et al.* [99] since this data has been published. The new 2013 TRPIV from Berger *et al.* [100] data has only been presented at conferences and not presented yet. Therefore in the future, it would be interesting to see how data fusion could be used for that particular data set since this included more data such as, longer time records, multiple samples of the same test (to increase the time record even more), and off-center planes that could perhaps be used to estimate a full three-dimensional data set instead of the planar data provided in the study.

The last thing to mention is the fact I believe the work presented will help set the foundation for future data and technique fusion work in fluid dynamics. This work only scratches the surface of what I believe will become a major field in the coming years. Even if the techniques used above are not used, the biggest takeaway should be data and technique fusion and creating an efficient framework are imperative to improving the field of fluid dynamics, especially since it seems that no one measurement technique is capable of extracting all of the knowledge needed to understand the increasingly challenging problems fluid dynamists are facing.

Appendix A

One Dimensional and Unsteady (Shock Tube)

Data fusion is now used on an unsteady one-dimensional test case. A shock tube is used in this study because the shock tube can be considered one dimensional if the wall effects are ignored, which is usually a reasonable assumption. Like the synthetic data ([chapter 3](#)), this work helps understand how to apply fusion to fluid dynamic data sets and is done before moving to the more complex jet problem.

A.1 Shock Tube Theory

A brief introduction of shock tube theory is discussed before fusion is performed. A shock tube is a piece of equipment used to study moving shocks, which are important in detonations and acoustic propagation. [[157](#)].

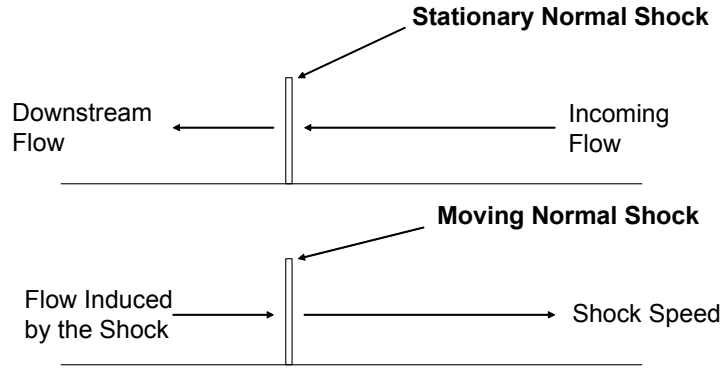


Figure A.1: *Diagram of a Stationary and Moving Shock*

A moving shock follows the same theory as a stationary shock but in a moving frame of reference. In fact, the shock only depends on the incoming velocity experienced by the shock.

The shock tube works by creating a large pressure difference in the tube that moves towards the low pressure section, forming a shock. In addition to the shock, an expansion wave is formed and travels into the high pressure section. The shock and expansion wave travel until they hit the wall of the tube and are reflected away from the wall. The strength and speed of the shock depends on the original pressure ratio in the tube ($\frac{P_4}{P_1}$). A schematic of the pressure distribution in the tube is shown in figure [A.2](#) at the initial condition, after the shock is formed, and after the shock is reflected.

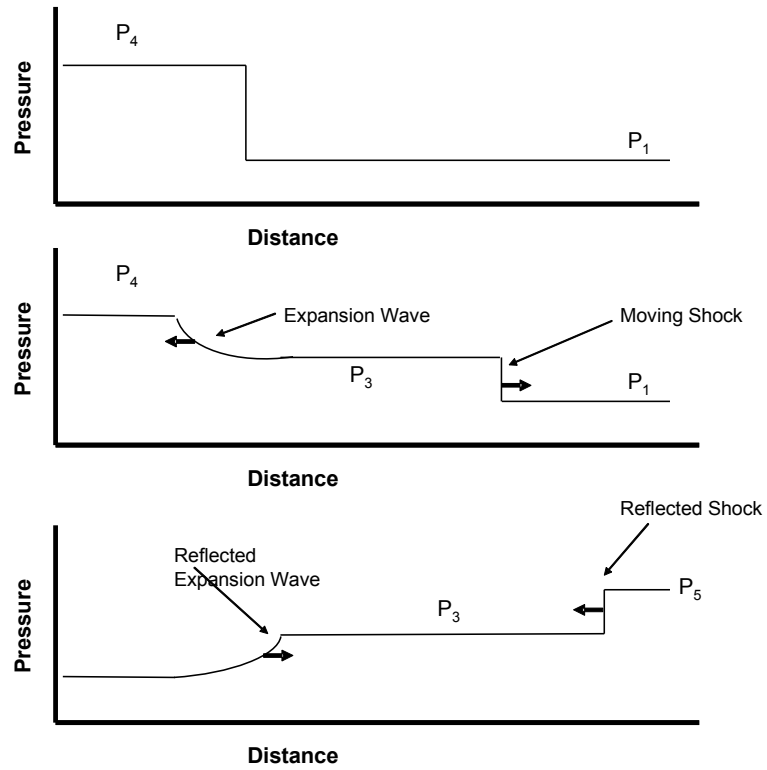


Figure A.2: *Diagram of a Shock Tube Before the Shock Wave is Released, Before the Wave is Reflected, and After the Wave is Reflected*

Now that the shock tube has been described physically, the underlying theory can be explained. The theory is derived by performing a mass, momentum, and energy balance on a control volume that moves with the shock [157].

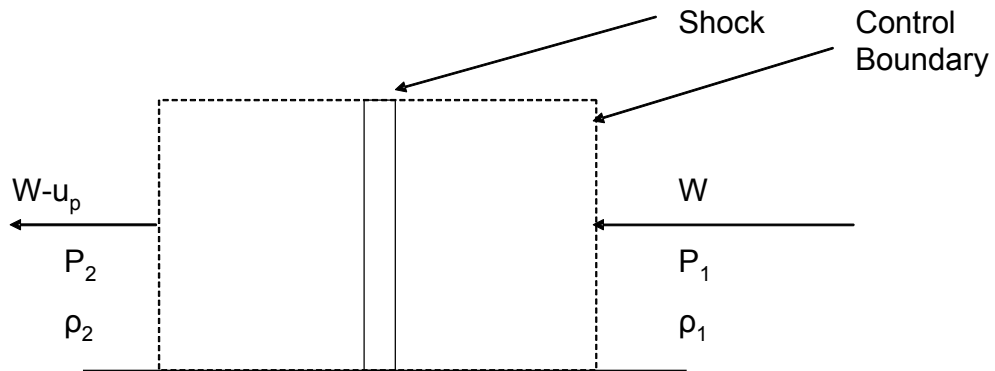


Figure A.3: *Control Volume Around a Shock*

The mass balance is

$$\rho_2(W - u_p) - \rho_1 W = 0 \quad (\text{A.1})$$

where ρ is the density, W is the speed of the shock, and u_p is the velocity induced by the shock. The momentum balance is

$$\rho_1 W^2 + P_1 - \rho_2(W - u_p)^2 - P_2 = 0 \quad (\text{A.2})$$

where P is pressure. The energy balance is as follows,

$$\frac{W^2}{2} + h_1 - \frac{(W - u_p)^2}{2} - h_2 = 0 \quad (\text{A.3})$$

where h is the enthalpy.

Equations [A.1](#), [A.2](#), and [A.3](#) can be rearranged in order to get the Hugoniot equation

$$e_2 - e_1 = \frac{P_1 - P_2}{2} \left(\frac{1}{\rho_1} - \frac{1}{\rho_2} \right) \quad (\text{A.4})$$

where e is the internal energy and is related to enthalpy by the following relation $h = e + P/\rho$. The Hugoniot equation can be simplified by assuming a calorically perfect gas, using the following relationships, $e = c_p T$ and $P = \rho R T$ which gives

$$\frac{T_2}{T_1} = \frac{P_2}{P_1} \left(\frac{\frac{\gamma+1}{\gamma-1} + \frac{P_2}{P_1}}{1 + \frac{\gamma+1}{\gamma-1} \frac{P_2}{P_1}} \right) \quad (\text{A.5})$$

where T is the temperature. The speed of the shock wave can also be calculated from equations [A.1](#), [A.2](#), and [A.3](#) and the perfect gas relations.

$$W = a_1 \sqrt{\frac{\gamma+1}{2\gamma} \left(\frac{P_2}{P_1} - 1 \right) + 1} \quad (\text{A.6})$$

where a_1 is the speed of sound. The induced velocity behind the shock can be calcu-

lated from the shock speed and density ratio by rearranging equation [A.1](#).

$$u_p = W \left(1 - \frac{\rho_1}{\rho_2} \right) \quad (\text{A.7})$$

A similar analysis gives the speed of the reflected shock wave. The rise in pressure due to the shock wave hitting the wall must satisfy the condition that the velocity of the wall is zero as is the air behind the reflected shock. Using this fact and a control volume analysis similar to the incident shock, the reflected shock Mach number is calculated using

$$\frac{M_R}{M_R^2 - 1} = \frac{M_s}{M_s^2 - 1} \sqrt{1 + \frac{2(\gamma - 1)}{(\gamma + 1)^2} (M_s^2 - 1) \left(\gamma + \frac{1}{M_s^2} \right)} \quad (\text{A.8})$$

With the isentropic relations, the shock strength $\left(\frac{P_2}{P_1} \right)$ can be defined based on the initial pressure condition of the shock tube $\left(\frac{P_4}{P_1} \right)$ with the following relation:

$$\frac{P_4}{P_1} = \frac{P_2}{P_1} \left(1 - \frac{(\gamma_4 - 1)(a_1/a_4)(P_2/P_1 - 1)}{\sqrt{2\gamma_1[2\gamma_1 + (\gamma_1 + 1)(P_2/P_1 - 1)]}} \right)^{-2\gamma_4/(\gamma_4 - 1)} \quad (\text{A.9})$$

which can be solved for the shock strength using a numerical approach. With these equations, the properties of the shock tube can be defined based on the initial pressure ratio and other initial thermodynamic quantities.

A.2 Experimental setup

The experiment for this test case was performed in the Syracuse University shock tube facility as part of an undergraduate lab. The experiment was focused on determining the speed of the incident and reflected shocks formed in the tube. A schematic of the tube is shown in figure [A.4](#).

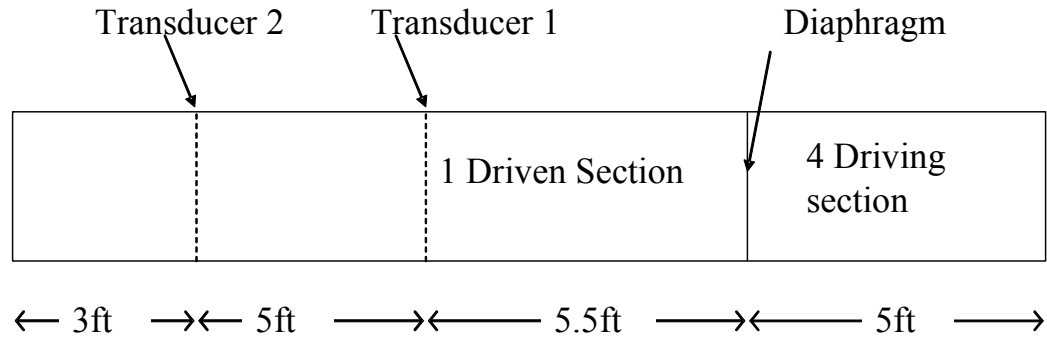


Figure A.4: *Schematic of the Syracuse University Shock Tube*

The driving section of the tube was separated from the driven section of the tube with a diaphragm made of acetate film. The driven section was then pressurized to various pressure ratios $\left(\frac{P_4}{P_1}\right)$. In this work a pressure ratio of 1.75 is studied.

The diaphragm is punctured once the pressure ratio is at the correct level, sending a blast of high pressure air down the shock tube, which produces a moving shock wave. The shock wave then passes pressure transducer 1, which is used to trigger pressure transducer 2. The pressure in the tube is then measured by both of the pressure transducers. The test lasts long enough for the incident shock and reflected shock to pass both transducers. The pressure signals from both transducers are shown in figure A.5, where the first set of peaks represent the shock passing and the second set of peaks are the reflected shock.

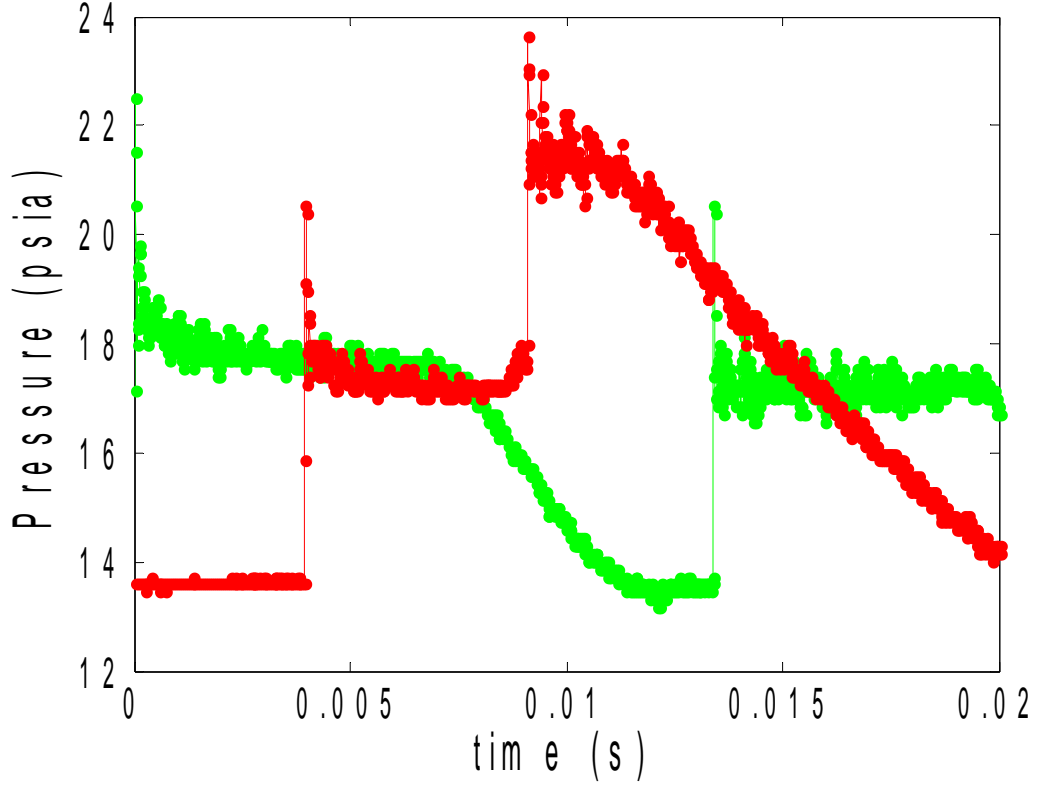


Figure A.5: *Pressure Measurements at the Two Transducers*

A.3 Simulation Setup

The Shock tube experiment is simulated by discretizing the Euler equations, which are the Navier-Stokes equations with the inviscid assumption. The one-dimensional Euler equations in the conservative form

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad (\text{A.10})$$

where $U = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}$ and $F = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ (E + P)u \end{pmatrix} = \begin{pmatrix} U_2 \\ 0.5(3 - \gamma)\frac{U_2^2}{U_1} + (\gamma - 1)U_3 \\ \gamma\frac{U_2 U_3}{U_1} - 0.5(\gamma - 1)\frac{U_2^3}{U_1^2} \end{pmatrix}$ are solved for this simulation using the Lax-Wendroff method. The above equation is

non-linear and cannot be solved in its current form using the Lax-Wendroff method. Therefore, the equations are rearranged using the chain rule ($\frac{\partial F}{\partial U} \frac{\partial U}{\partial x} = \frac{\partial F}{\partial x}$) where $\frac{\partial F}{\partial U}$ is the Jacobian

$$\frac{\partial F}{\partial U} = A = \begin{bmatrix} 0 & 1 & 0 \\ 0.5(\gamma - 3)u^2 & (3 - \gamma)u & \gamma - 1 \\ (\gamma - 1)u^3 - \gamma \frac{ue_t}{\rho} & -\frac{3}{2}(\gamma - 1)u^2 + \gamma \frac{e_t}{\rho} & \gamma u \end{bmatrix}. \quad (\text{A.11})$$

The Euler equations can now be written as a Quasi-linear equation,

$$\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial x} = 0 \quad (\text{A.12})$$

which can be discretized using the Lax-Wendroff method.

$$U_i^{n+1} = U_i^n - \frac{1}{2} \frac{\Delta t}{\Delta x} (F_{i+1}^n - F_i^n) + \frac{1}{2} \frac{\Delta t^2}{\Delta x^2} \left[\frac{A_i + A_{i+1}}{2} (F_{i+1}^n - F_i^n) - \frac{A_i + A_{i-1}}{2} (F_i^n - F_{i-1}^n) \right] \quad (\text{A.13})$$

Notice in equation A.13, that the Lax-Wendroff acts like a forward difference scheme with the addition of a diffusion term, to help keep the scheme stable. For stability purposes, the Lax-Wendroff scheme requires the CFL number to be less than one for a linear system. However, since the Euler equations are non-linear, the CFL needs to be less than one to account for non-linear effects.

A.3.1 Boundary and Initial Conditions

For the one-dimensional case, there are two boundary conditions, the wall at each end of the tube. They are simulated by setting the velocity at the wall equal to zero and setting the derivatives of density and energy to zero at the wall causing an increase in pressure, making the shock bounce off the wall.

The pressure on the driven side of the diaphragm is initially set to ambient and the pressure on the driven side is set to a desired pressure. The density is initially

set using the pressure ratio and the isentropic relationship; the velocity is initialized by setting it equal to zero everywhere in the tube.

A.3.2 Computational Results

The pressure from the simulation (figure A.7) is extracted at the same location as the pressure transducers in the experimental work.

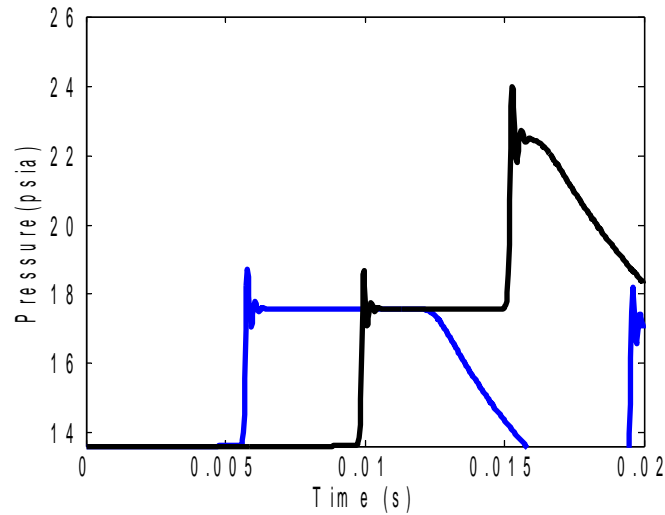


Figure A.6: *Computational Pressure Measurements at the Two Pressure Transducers*

The simulated pressure and experimental pressure have similar shape. Each profile has a constant pressure after the shock passes the first sensor and a decreasing pressure after the second sensor. However, the shock arrival time for the two different cases do not match.

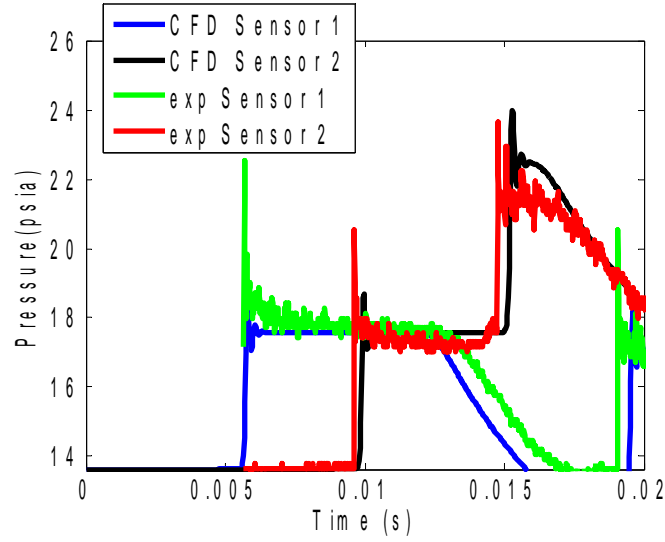


Figure A.7: Comparison of Pressure Measurements obtained using CFD and Pressure Transducers

$\frac{P_4}{P_1}$	Measurement	Incident Speed (ft/s)	Reflected Speed(ft/s)
1.74	Experiment	1272	1162
1.74	CFD	1196	1157
1.74	Theory	1208	1153
2.48	Experiment	1358	1133
2.48	CFD	1281	1142
2.48	Theory	1303	1163
3.21	Experiment	1420	1096
3.21	CFD	1328	1113
3.21	Theory	1375	1170

Table A.1: Comparison of Shock Speed Predicted by CFD and Pressure Transducers

The discrepancy in the shock arrival time causes an even larger discrepancy in the speed of the shock (see table A.1). Trying to fuse the signals without correcting this issue produced poor results, due to the large gradients. To account for this the data are shifted to match the data.

A.4 Fusion Results

A.4.1 Shock Arrival Time

The experimental shock has a slightly different shock speed than theory because theory assumes the flow is lossless. In reality, the walls of the tube have friction, causing a boundary layer to form, which acts as a constriction in area. Since the shocks are moving at supersonic speeds, this should cause the shock speed to be smaller than theoretical. This is not the case, the experiment is predicting larger speeds.

This may mean something is not correct in the experiment. Perhaps it has something to do with the spacing of the sensors, build up on the back wall of the shock tube, or errors in reading the initial pressure ratio. Changing the sensor location is studied in this analysis and shows that if the sensors are moved a few inches the computational, experimental, and theoretical results are about the same. A change in pressure could have similar effects and will be studied in future work.

A change in the distance between the sensors and between the wall could greatly effect the calculated shock speed. Using multiple pressure ratios and the corresponding speed, the RMS difference between the CFD and experimental results is minimized by changing the location of the sensors. A contour plot is generated to see the level of error vs. the position of the sensors (figure [A.8](#)). This shifting of the data will be called zero level fusion because it was required to make the data consistent for further fusion processes.

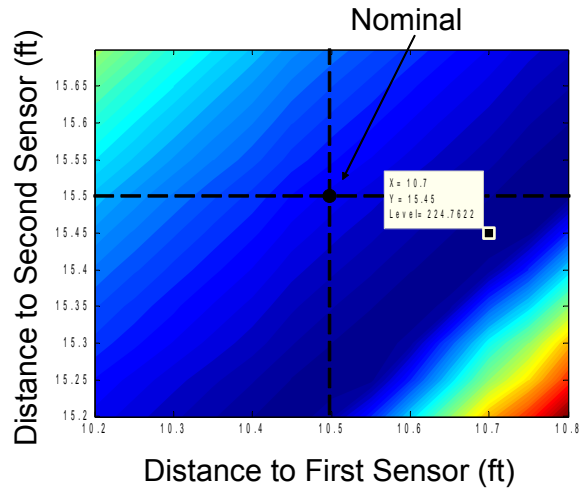


Figure A.8: *Level Zero Data Fusion for the Shock Tube Data*

By shifting the sensors a few inches the experimental, CFD, and theoretical data match more closely. The difference between the CFD and experimental results for both the fused case and non-fused case are displayed below in table A.3.

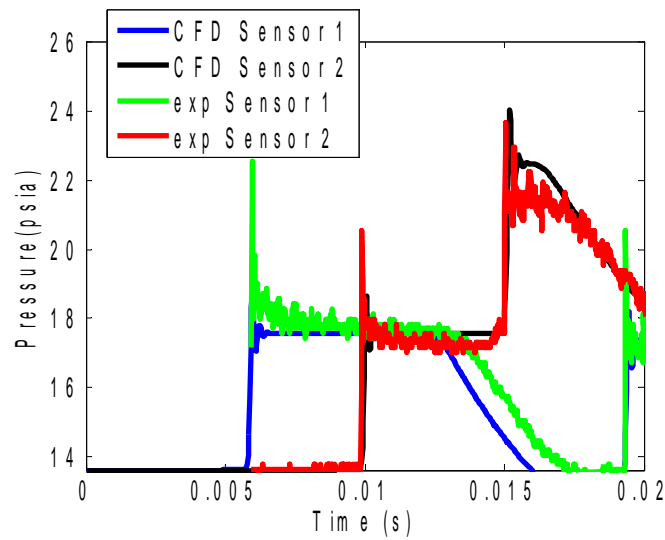


Figure A.9: *Fusion by Changing the Pressure Sensor Location*

$\frac{P_4}{P_1}$	Measurement	Incident Speed (ft/s)	Reflected Speed(ft/s)
1.74	Experiment	1221	1116
1.74	CFD	1223	1111
1.74	Theory	1208	1153
2.48	Experiment	1304	1088
2.48	CFD	1276	1133
2.48	Theory	1303	1163
3.21	Experiment	1353	1096
3.21	CFD	1334	1105
3.21	Theory	1375	1170

Table A.2: Comparison of Shock Speed Predicted by CFD, Theory, and Pressure Transducers After Data Fusion

$\frac{P_4}{P_1}$	Fused	Difference in Incident Speed (ft/s)	Difference in Reflected Speed(ft/s)
1.74	Fused	2	5
1.74	Non-Fused	76	5
2.48	Fused	28	45
2.48	Non-Fused	77	10
3.21	Fused	19	9
3.21	Non-Fused	25	17

Table A.3: Difference in Shock Speed Predicted by CFD and Pressure Transducers After Data Fusion

A.4.2 Kalman Filter Fusion

Now that the shock arrival times between the CFD and experiments are more consistent, data fusion is applied. A Kalman filter is applied to the CFD and pressure data to decrease the noise levels in the experimental data while adding some of the physics that are missing in the CFD model. However the Euler equations are non-linear and the Kalman filter requires a linear system of equations. To get around this, the CFD is used to generate a stochastic model ([subsection 1.4.4](#)). However, the model does not predict the shock ([figure A.11](#)) well.

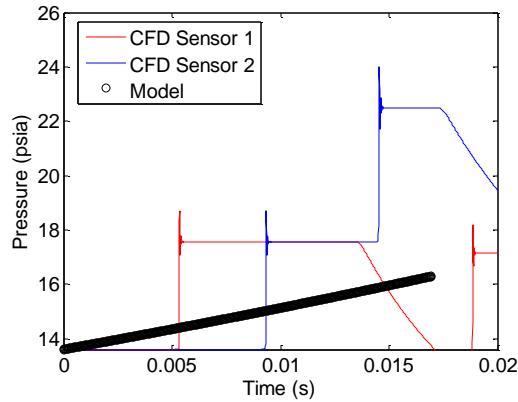


Figure A.10: *Stochastic Model Applied to the Shock Tube Data*

Applying a Kalman filter to this should help, but the model does such a poor job that Kalman Filter does not help.

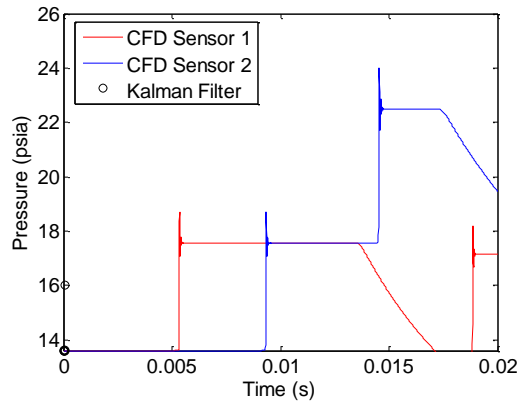


Figure A.11: *Kalman Filter Applied to the Shock Tube Data*

This could be due to the non-stationary nature of the shock tube, which does not seem to work well with this type of model. I believe the modeling is acting in this manner because of the shock which acts as an infinite jump. This could be investigated more and a model that better predicts the shock could be employed, but the focus of this work is ultimately applying data fusion to the jet.

Gappy POD Fusion

Gappy POD is able to estimate missing information as shown in [chapter 3](#). Since the modeling and Kalman filtering approach did not work well, Gappy POD is used to estimate the pressure data using a CFD basis and the two experimental pressure sensors.

The results of the Gappy POD fusion in this case are rather disappointing. The two sensor are not able to get a reasonable estimate of the pressure trace. This technique was used to see if Gappy POD is capable of estimating a signal instead of using the Kalman filtering approach as attempted previously.

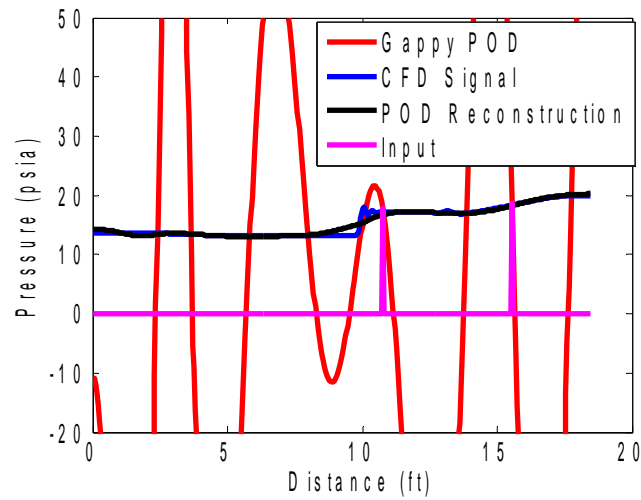


Figure A.12: *Gappy POD Fusion using a CFD Basis and Two Pressure Transducers*

The Gappy POD fusion technique was only able to capture the pressure correctly at the two sensors. The Gappy POD fusion technique produced a signal that had many oscillations, which makes sense though. First, only two sensors are being used to estimate the whole domain, which is not a lot of information and as shown by the results not enough information for fusion. Second, the physics of the problem are such that the different locations do not sense the effects of the pressure wave until it

passes the location in question.

CFD information is used as virtual pressure sensors to determine if a lack of information is really the issue causing the sub-par estimation. This results is also disappointing. Again, the Gappy POD fusion was unable to predict the missing information.

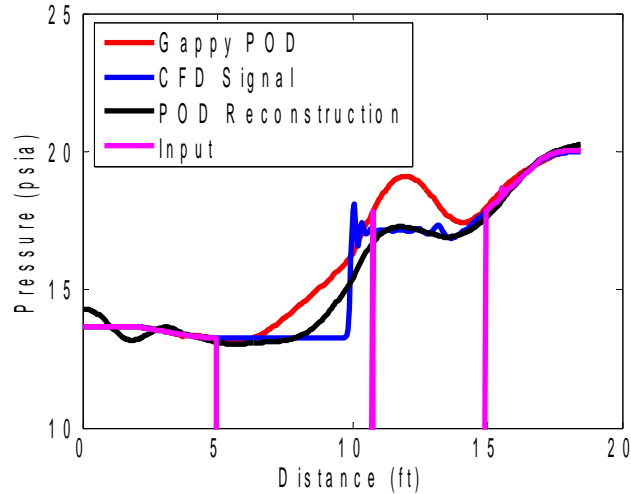


Figure A.13: *Gappy POD Fusion using a CFD Basis and Two Pressure Transducers and 5 CFD Virtual Sensors*

The addition of the virtual sensors does however show an improvement over the fusion using only two sensors. The pressure has a similar trend to that of the pressure predicted by CFD. However, the fusion is still unable to capture the sharpness of the shock.

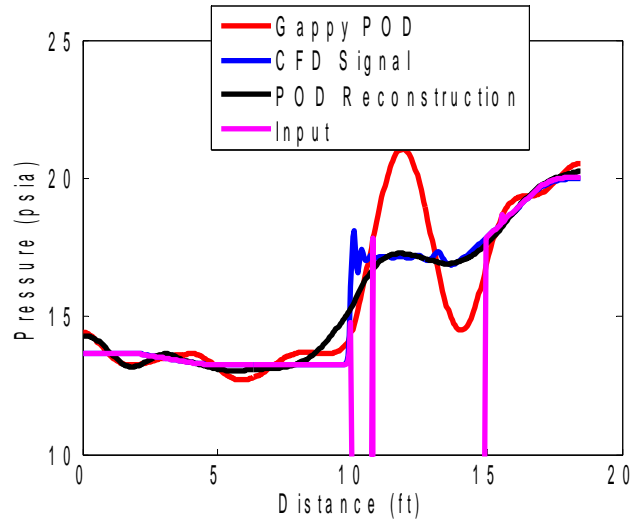


Figure A.14: *Gappy POD Fusion using a CFD Basis and Two Pressure Transducers and 10 CFD Virtual Sensors*

The addition of more virtual sensors, does not improve the shock capturing abilities of the fused signal. Although, more information did make the signal look more like the actual signal. This shows that to use Gappy POD, enough sensor information is required to predict the flow. The use of Gappy POD to estimate missing data is discussed in more detail in [chapter 4](#) and [chapter 5](#).

A.4.3 Summary and Lessons Learned

The study of the shock tube allowed for the use of the data fusion techniques on real fluid dynamic data. This work shows that some of the techniques that worked for the proof of concept did not work as well on the shock tube data. However, the technique had great success on the airfoil data (4) and the jet data (5). This demonstrates that one needs to think about how data fusion is applied and techniques that work well with one data set will not necessarily work well with another data set. One needs to understand the data and how data fusion will effect it.

The first technique presented used computational data and various experimental data to align the different data sets. Changing the location of the sensors in the

analysis of the data produced results that were more consistent between the CFD and experimental pressure in terms of incident shock speed. Using the data in this simple way greatly increased the consistency between the data. This idea is applied to the jet work to align the different data sets that are used in the fusion process.

Next, a Kalman filtering technique was applied to the data using a linear stochastic model based on the CFD to overcome the issue of non-linearity. The model used in this technique did not predict the flow and the Kalman filter was unable to help improve the model. However, the shock tube problem and the jet problem have very different physics and as such, this technique should not be ruled out for the jet work.

Next the Gappy POD technique estimates data not captured by the sensor using CFD information and the pressure measurements. These techniques seemed to give better results than the Kalman filter, but still were unable to capture the shock. This was due to lack of data to perform the fitting process. When more data was added, the Gappy POD did a better job of estimating the signal. This technique is used on the jet data and is described in more detail in [section 5.4](#).

Appendix B

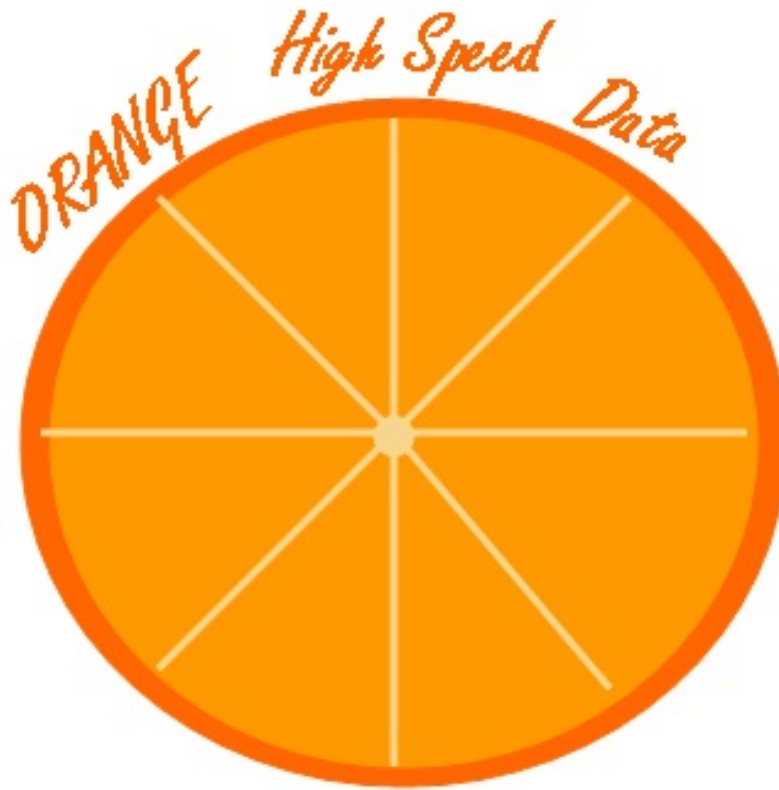
ORANGE HSD User Manual:

User Guide for Orange High Speed Data

Version 3.4.5.0-December 2013

Christopher J. Ruscher *

Syracuse University, Syracuse ,NY (cjrusche@syr.edu)



Acknowledgements

I would like to acknowledge all of the people who have helped me in creating this program.

Robert Kapuku
John F. Dannenhoffer, III
Mark N. Glauser
Zachary P. Berger
Matthew G. Berry
Dale C. Ringham
Bridget M. Dixon
Shanmugam Murugappan
Stephen Ettore
Steven Martens
Jeff Monfort
Barry V. Kiel

This project was funded by the General Electric Company and Innovative Scientific Solutions Inc.

Contents

I	Introduction	5
II	Installation and Execution	5
A	Folder Content	5
B	Installation	5
C	Note on Additional Toolboxes	6
D	Using the Software	6
III	Using ORANGE HSD	8
A	Data Management	11
1	Import Data:	13
2	Format Data:	13
3	Edit User Data:	14
4	Rescale Grid:	15
5	Remove File:	16
6	Mask	16
7	Extract Data 3D-2D	18
8	Extract Data 2D-2D	19
9	Extract 2D-1D	20
10	1D-1D Extraction	21
11	signal Conditioning	22
12	View Temporal Signal	23
13	View Field Data	24
14	Show/Collapse Time Data:	25
B	ORANGE POD	26
1	Capabilities of ORANGE POD	26
2	Using Orange POD	27
3	Running ORANGE POD	29
4	Outputs	29
C	ORANGE LSE	30
1	Capabilities of ORANGE LSE	30
2	Using ORANGE LSE	31
3	Running ORANGE LSE	32
4	Outputs	32
D	ORANGE Wavelet	33
1	Capabilities of ORANGE Wavelet	33
2	Using ORANGE Wavelet	33
3	Running ORANGE Wavelet	34
4	Outputs	34
E	ORANGE Correlate	35
1	Capabilities of ORANGE Correlate	35
2	Using ORANGE Correlate	35
3	Running ORANGE Correlate	36
4	Outputs	37
F	ORANGE Statistics	37
1	Running ORANGE Statistics	37
G	Custom Functions	38
1	Running ORANGE Calculator	38
2	Creating a New Function	39

IV Theory	39
A Proper Orthogonal Decomposition	39
1 Classical POD	39
2 Snapshot POD	40
3 Split POD	40
4 Gappy POD	41
5 Fused POD	42
B Stochastic Estimation	43
C Wavelets	44
V Trouble Shooting	45
VI Useful hints	45
VII Possible Future Work	45
VIII Code Listing	45

I. Introduction

The improvement of experimental and numerical techniques have led to an increase in the amount of data collected for test cases. The increased size of data sets can become an issue when doing post processing. The data needs to be processed in an efficient manner as well as utilized in such a way that both numerical and experimental data are used to gain a deeper understanding of the phenomena being analyzed.

ORANGE High Speed Data (HSD) is a tool that was developed to deal with fluid dynamic data. Its purpose is to make performing some of the typical fluid dynamic analyses easier. The program is also designed to look at data from different data sets side-by-side and help make comparisons. The tool is also a good way to keep track of data. The code includes: proper orthogonal decomposition (POD), stochastic estimation, wavelet analysis, spectral analysis, statistics, and a custom function creator.

II. Installation and Execution

A. Folder Content

All of the necessary Matlab functions should be contained in the ORANGE HSD folder. The folder should be placed in an accessible, writable location such as the Desktop or My Documents. The contents of the folder should be:

1. Folders:

- (a) **Subfunction:** This folder contains the codes that run the various analyses
- (b) **pivmat2.01:** This folder contains the Matlab toolbox that processes PIV data
- (c) **readimx4matlab_v1.5R1_2009:** This folder contains an image processing toolbox
- (d) **readimx4matlab_v1.5R1_2009_64:** 64 bit version of the above toolbox
- (e) **Test:** A zipped folder that contains the data used in the manual and video

2. Matlab code files:

- (a) **High_Speed_Data.m:** This is the code that creates the main GUI screen

3. Matlab figure files:

- (a) **High_Speed_Data.fig:** This is the main GUI Screen

4. Documentation

- (a) **High Speed Data Manual.pdf:** This is the user manual for the software
- (b) **ORANGE_HSD_Tutorial.mpg** This is a movie tutorial of the basic functionality of ORANGE HSD

B. Installation

The user

The user must extract the pivmat2.01 zipped folder and one of the readimx4matlab zipped folders (whichever corresponds to your version of MATLAB, 32-bit or 64-bit). Only unzip and install the readimx version that corresponds to your version of MATLAB to prevent bugs. They should be unzipped into the ORANGE HSD, so that all paths are added together. With these files accessible and unzipped, their paths can now be added to MATLAB.

ADDING THE FILE PATHS

1. Open MATLAB
2. Select File>>Set Path
3. Select "Add with subfolders"
4. Browse to the ORANGE HSD folder with all the MATLAB functions. Select it.

5. Select "OK"
6. Select "Save"
7. Close out of Set Path

If the updated paths do not save, you can perform the installation manually. The Steps to do this are as follow:

1. Open the file "path.def". This should be located in MATLABRoot\toolbox\local
2. Before `%%%END ENTRIES%%%` add the following lines

```
'local'; ...
'local\Subfunctions'; ...
'local\Subfunctions\Calculator'; ...
'local\Subfunctions\Orange_LSE_QSE'; ...
'local\Subfunctions\Orange_POD'; ...
'local\Subfunctions\Orange_Correlations'; ...
'local\Subfunctions\Orange_Wavelet'; ...
'local\Subfunctions\Conversion'; ...
'local\Subfunctions\Data_View'; ...
'local\Subfunctions\Conditioning'; ...
'local\Subfunctions\Statistic'; ...
'local\Subfunctions\Data_Extraction'; ...
'local\pivmat2.01'; ...
'local\pivmat2.01\pivmat'; ...
'local\pivmat2.01\pivmat\html'; ...
'local\pivmat2.01\pivmat\sample'; ...
'local\pivmat2.01\pivmat\sample\jet'; ...
'local\pivmat2.01\pivmat\sample\surf'; ...
'local\pivmat2.01\pivmat\sample\turb'; ...
'local\readimx4matlab_v1.5R1_2009'; ...
'local\readimx4matlab_v1_5R1_2009\readimx4matlab_v1.5R1_2009'; ...
'local\readimx4matlab_v1.5R1_2009\readimx4matlab_v1.5R1_2009\images'; ...
```

3. 'local' is the location of ORANGE HSD
4. Save path.def

C. Note on Additional Toolboxes

pivmat2.01 is a toolbox readily available to download for free. It is a toolbox developed specifically to handle large velocity fields created by LaVision by F. Moisy. See www.fast.u-psud.fr/pivmat for more details. In addition, pivmat2.01 comes with its own readme and MATLAB help sections.

readimx is software developed directly by LaVision. It is made to read LaVision's images into MATLAB. It also comes with a detailed ReadMe pdf. Those with LaVision accounts can download the most up-to-date version directly from LaVision's website.

A number of other functions included were also developed from other sources; credit and author information is given in the opening comments of those functions.

MATLAB toolboxes required to process PLIF data: Image Processing Toolbox, Signals Processing Toolbox

MATLAB toolboxes required to process PIV and Pressure data: Signals Processing Toolbox

D. Using the Software

Setup and General Information:

To begin using ORANGE HSD, either manually open the function "High_Speed_Data.m" and run it or type in "High_Speed_Data" in the command window. This will open the main menu (figure 1). All of the functions available in ORANGE HSD are accessed from this menu. The options on the main menu are color coded in order to let the user know which options are available. **Green** means the options is available, **red** means the option is not available yet, and

brown means the option might be available or it has already been used.

When the program is first opened, the user has two options, select an already existing file using the ... button (button 2 in figure 1) or use the “Setup Data Directory” button (button 3 in figure 1). Use the second option if you are starting with a new set of data. This option will ask the user to name the directory of the new data set. Then the user will be prompted to select the data set that will be moved or copied into the newly created directory. Whenever the user adds data to the directory, the user will be asked a few questions in order to register the data. (The next time the user opens the program it will automatically open to the the directory the user was using last)

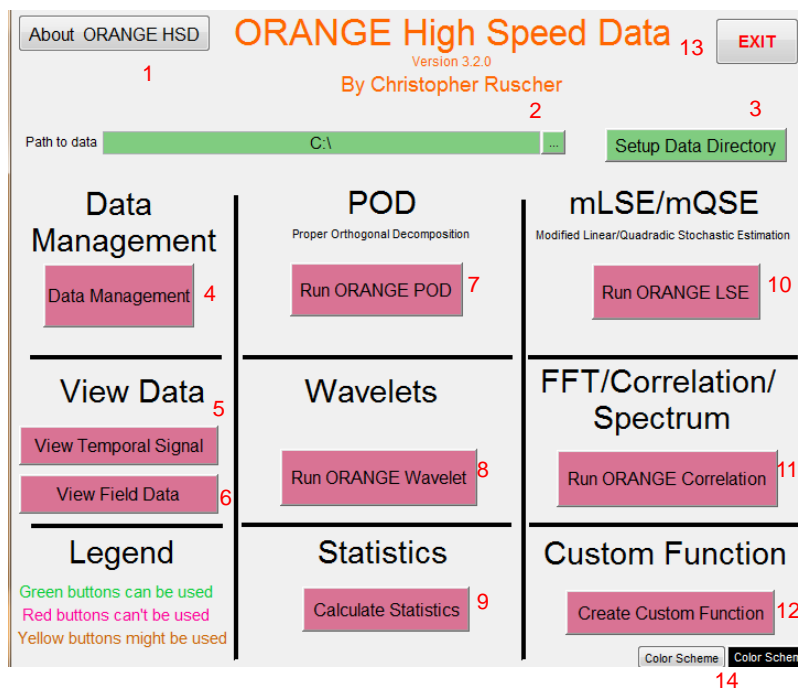


Figure 1. Main GUI Interface (High Speed Data)

1. **About ORANGE HSD:** This will bring up a screen with a short description of ORANGE HSD
2. **Select the main folder:** Select the directory that contains all of the data that will be used
3. **Setup Data Directory:** This option allows the user to setup a new data directory by moving/coping data
4. **Data Management:** This option opens the data management screen
5. **View Temporal Signal:** This option opens a window to view the raw data signal
6. **View Field Data:** This option opens a window to view the raw data contours
7. **Run ORANGE POD:** This option opens the POD GUI.
8. **Run ORANGE Wavelet:** This option opens the wavelet GUI
9. **Calculate Statistics:** This option opens the statistics GUI
10. **Run ORANGE LSE:** This option opens the linear stochastic estimation (LSE) GUI
11. **Run ORANGE Correlation:** This option opens the correlation GUI
12. **Create Custom Function:** This option opens a GUI to create custom functions
13. **EXIT:** This option close ORANGE HSD.
14. **Color Scheme:** These buttons allow the user to change the color from black-on-white to white-on-black

III. Using ORANGE HSD

To start ORANGE HSD the user must first open Matlab. Once Matlab is open, the user can type “High_Speed_Data” into the Matlab command window. This displays the ORANGE HSD splash screen for a couple of seconds and then brings up the main menu.

The user should notice that most of the options are not available at this time. If this is a new project, the user must push the “Setup Data Directory” button. This will guide the user in creating a new project and importing the data to use for the project.

The first prompt asks the user to select where the folder for the new project will be located. The user then selects the directory to create the new project.

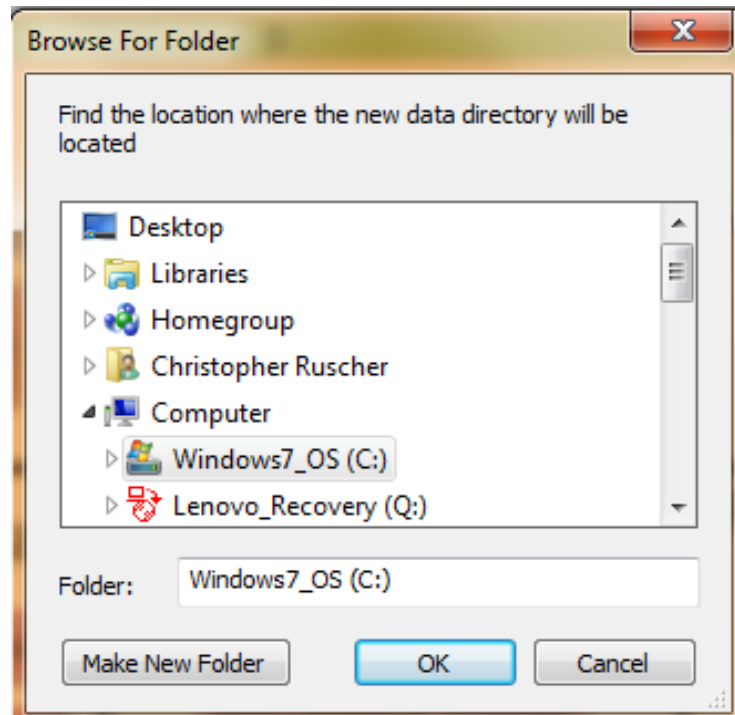


Figure 2. Where to Create a New Project

The user will then be asked to give the project a name. This process creates a new folder with the same name as the project name in the directory selected in the previous step.

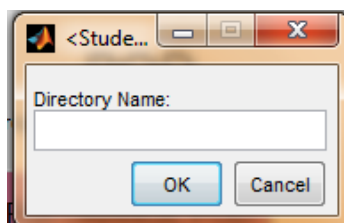


Figure 3. Name the New Project

The user is then asked to locate the data that will be used in the project. The user must decide whether to copy or move the data into the project folder from its current location. The user will then select the data. This is then repeated until the user has loaded in all of the desired data.

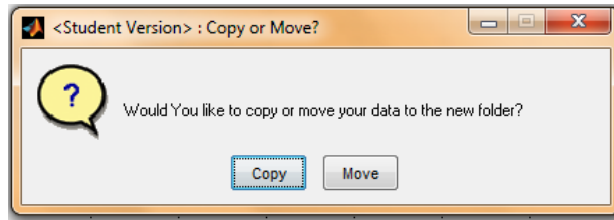


Figure 4. Move or Copy the Data Set

The user will then be asked a series of questions that will be used to register the data in the system. A few of these questions are shown in figure 6.

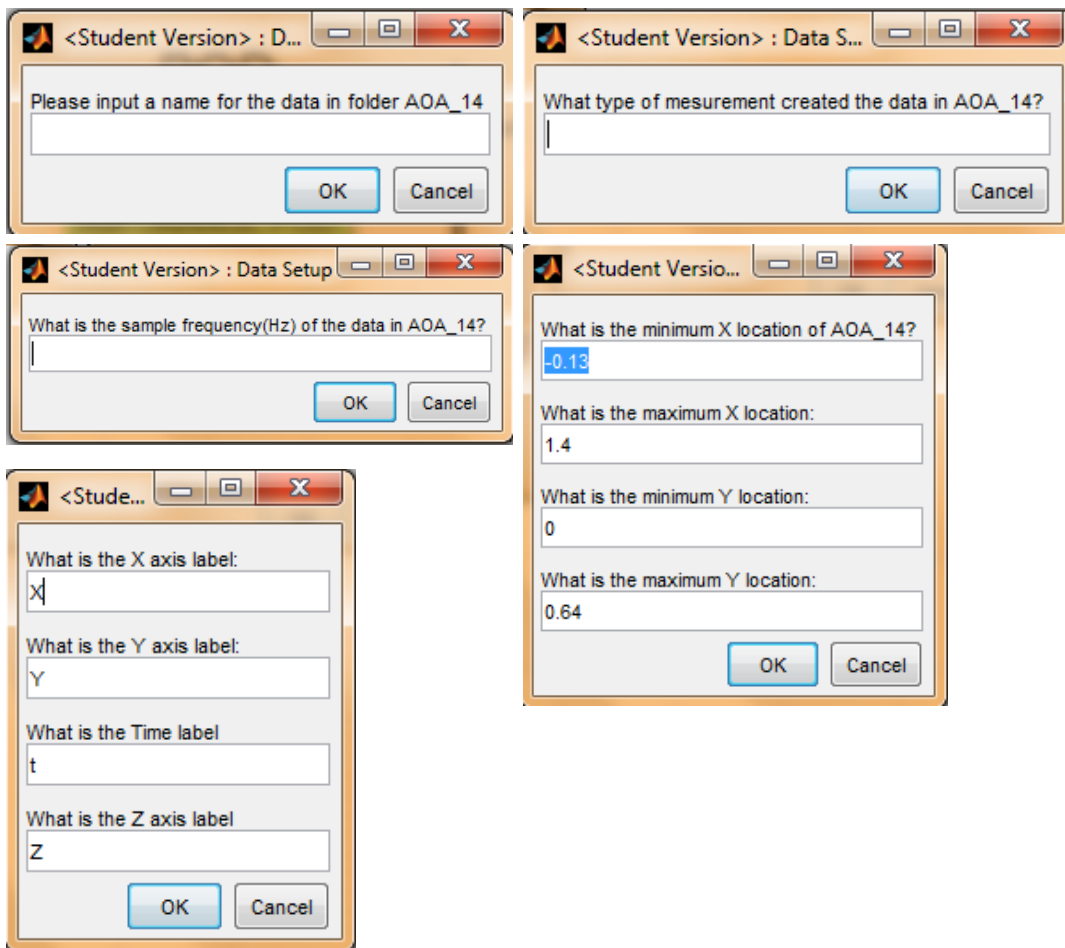


Figure 5. Some Questions to Register the Data

When the user finishes answering all of the questions for all of the data sets the main menu will appear again. Notice the Change in the color of the buttons. More options are now available.

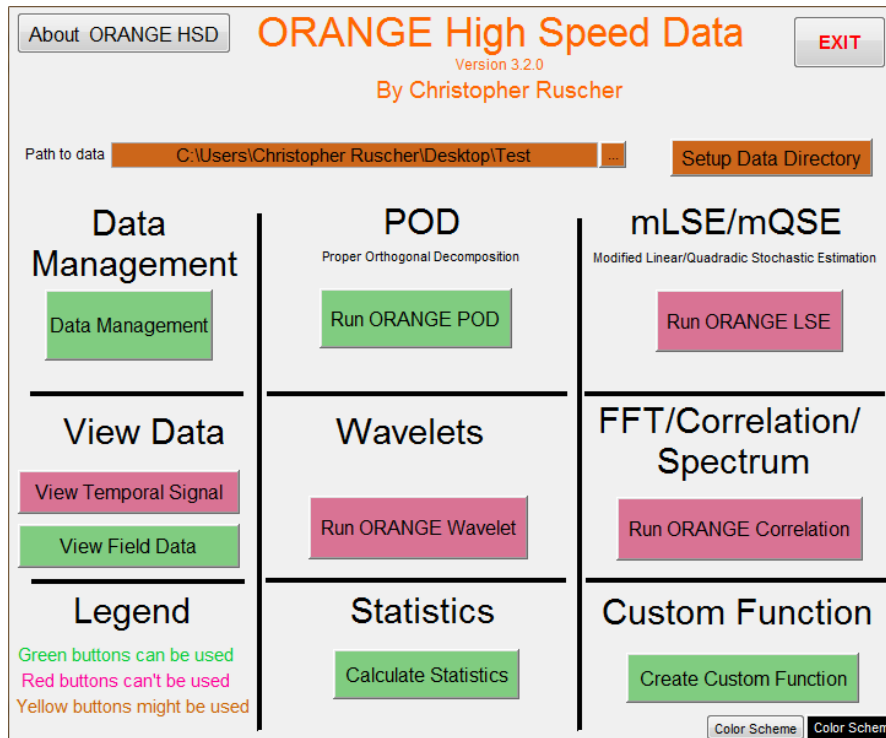


Figure 6. Main GUI Interface (High Speed Data) After Registering Data

Now that there are more options, let's start to look at them. We first look at the data management tool.

A. Data Management

Large amounts of data can be produced when processing your data, which can become confusing and hard to keep track of. To help the user track the data better the data management page has been added. The data management tool stores and displays important information about the data. It is shown in figure 8. The tool also has links to some other important features, which will be discussed later.

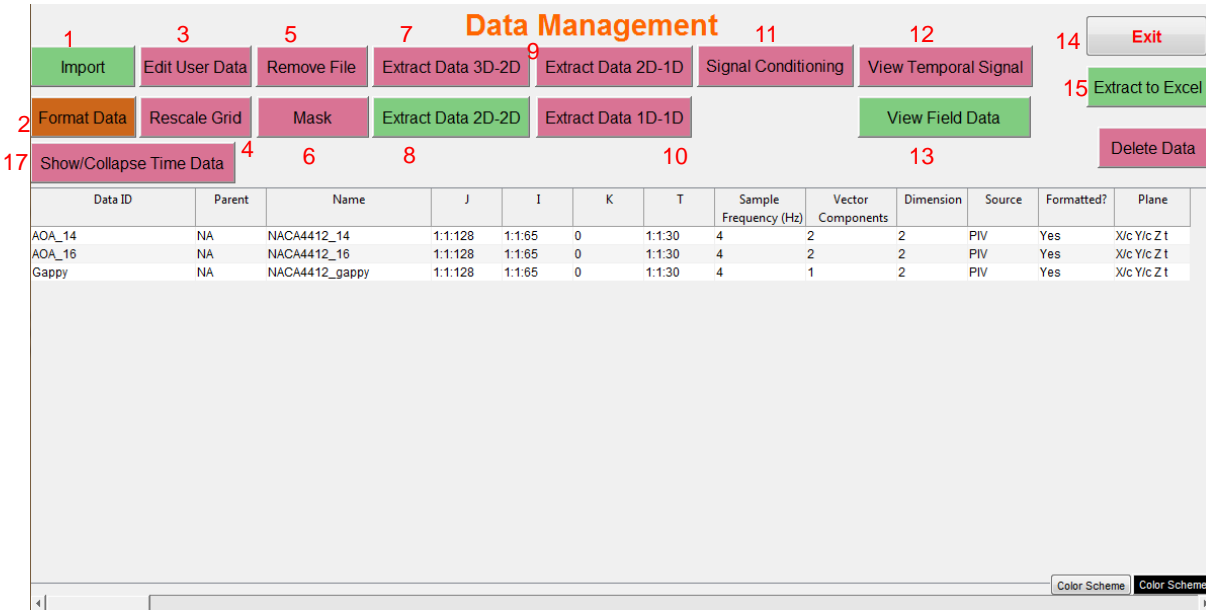


Figure 7. Data Management Page

1. **Import:** This button will allow the user to add more data to the current data set (when the data is added, the user will have to answer questions to initialize the data)
2. **Format Data:** This displays a GUI that will help format recognized data (This is discussed later)
3. **Edit User Data:** This allows the user to edit the data stored in the data management page
4. **Rescale Grid:** The user can use this option to change the axis scaling of the data
5. **Remove File:** The user can use this option if one or more snapshots in a data set are “bad”
6. **Mask:** The user can use this option to generate a mask for a data set.
7. **Extract Data 3D-2D:** Extract a 2D slice from a 3D data set
8. **Extract Data 2D-2D:** This tool can be used to extract a plane from another plane of data
9. **Extract Data 2D-1D:** Extract a signal from a set of snapshots
10. **Extract Data 1D-1D:** Extract part of a 1D data set
11. **Signal Conditioning:** This option allows the user to filter and/or smooth a signal
12. **View Temporal Signal:** View 1D data (This options is also on the main screen)
13. **View Field Data:** View 2D data (This options is also on the main screen)
14. **Exit:** Close the data management page and opens the main screen
15. **Extract to Excel:** Writes the information about the data to an excel sheet
16. **Delete Data:** Delete the selected data
17. **Show/Collapse Time Data:** This option allows the user to view a list of 1D signals contained in a data set

1. Import Data:

If the user needs to add a new data set to the current project, the user can use the “Import” button (button 1 in figure 8). This process is similar to starting a new project. The user is asked to locate the new data and whether to move or copy the data. The user is then asked questions about the data.

2. Format Data:

The data must be converted to the correct format before any analysis can be performed. All of the analyses are performed on .mat files. The format tool include in the ORANGE High Speed Data package will perform the file conversions. This can be done without the code as well. The format tool is capable of formatting PIV data (.vc7 files, .txt files), PLIF data (.tif files), CFD data (OVERFLOW files, fluent data formatted as point or block tecplot files, and GE CFD data), and pressure data (.txt file where the pressure data is stored in columns). The following screen capture (Figure 9) shows the layout of the format tool.

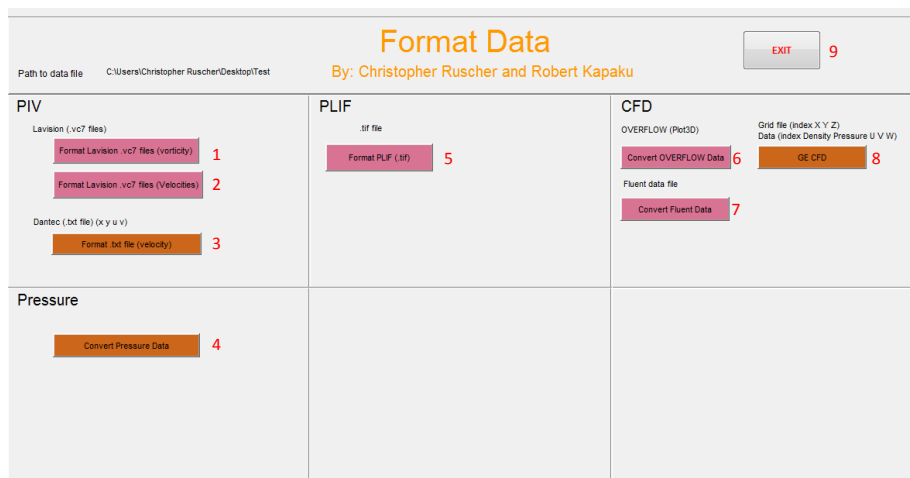


Figure 8. Format Data GUI

1. **Format Lavisson .vc7 file (vorticity):** This will read the Lavisson .vc7 file and write a .mat vorticity file
2. **Format Lavisson .vc7 file (velocity):** This converts the vc7 data into a .mat grid file and .mat velocity files
3. **Format .txt file (Velocity):** This converts a PIV file stored in a text file to a .mat file.
4. **Convert Pressure Data:** This converts a text file with data into a .mat file for each column of the text file
5. **Format PLIF (.tif):** This converts the pixel brightness of a .tif file into a .mat file
6. **Convert OVERFLOW Data:** This will convert the plot3d formatted file from OVERFLOW into a .mat file.
7. **Convert Fluent Data:** This opens a GUI that will help convert Fluent data
8. **GE CFD:** This will convert a text file that contains columns of data to a .mat file. data file (index,Density,Pressure,U,V,W) grid file (index,X,Y,Z)

The conversion of the PIV, PLIF, CFD, etc. data will create a grid.mat file as well as a file for each vector component at each time step. For example, a data set with N snapshots with three components of velocity requires files called grid.mat, U1.mat, V1.mat, W1.mat ... UN.mat, VN.mat, WN.mat. (V and W are only needed if the data has three components. The program will run with just U data.) The format of the .mat files is shown below in figure 10.

U1.mat	U2.mat		UM.mat	Grid.mat	
U(X1,Y1,t1)	U(X1,Y1,t2)		U(X1,Y1,tM)	X1	Y1
U(X2,Y2,t2)	U(X2,Y2,t2)		U(X2,Y2,tM)	X2	Y2
U(X3,Y3,t3)	U(X3,Y3,t2)	· · ·	U(X3,Y3,tM)	X3	Y3
·	·		·	·	·
·	·		·	·	·
·	·		·	·	·
U(XN,YN,t1)	U(XN,YN,t2)		U(XN,YN,tM)	XN	YN

Figure 9. File Format for ORANGE HSD Field Data

Similarly, the conversion of pressure signals, and other signals will be converted into a P1.mat, P2.mat, . . . , PN.mat. These files will each represent a different pressure signal. The format of these .mat file is shown below in figure 11.

P1.mat	P2.mat				Pm.mat
P(x1,y1,z1,t1)	P(x2,y2,z2,t1)				P(xm,ym,zm,t1)
P(x1,y1,z1,t2)	P(x2,y2,z2,t2)				P(xm,ym,zm,t2)
P(x1,y1,z1,t3)	P(x2,y2,z2,t3)	·	·	·	P(xm,ym,zm,t3)
·	·				·
·	·				·
·	·				·
P(x1,y1,z1,tn)	P(x2,y2,z2,tn)				P(xm,ym,zm,tn)

Figure 10. File Format for ORANGE HSD signal data

3. Edit User Data:

The user may want to change the value stored in the data management tool. This option allows the user to change some of the desired values. The user needs to select the cell that will be changed and press the “Edit User Data” button (button 3 in 8) if it is allowed to be changed. A pop-up window will appear and the user can type the new value. The user can then click the “ok” button for the change to occur. The data management page will disappear and the reappear as the change is made.

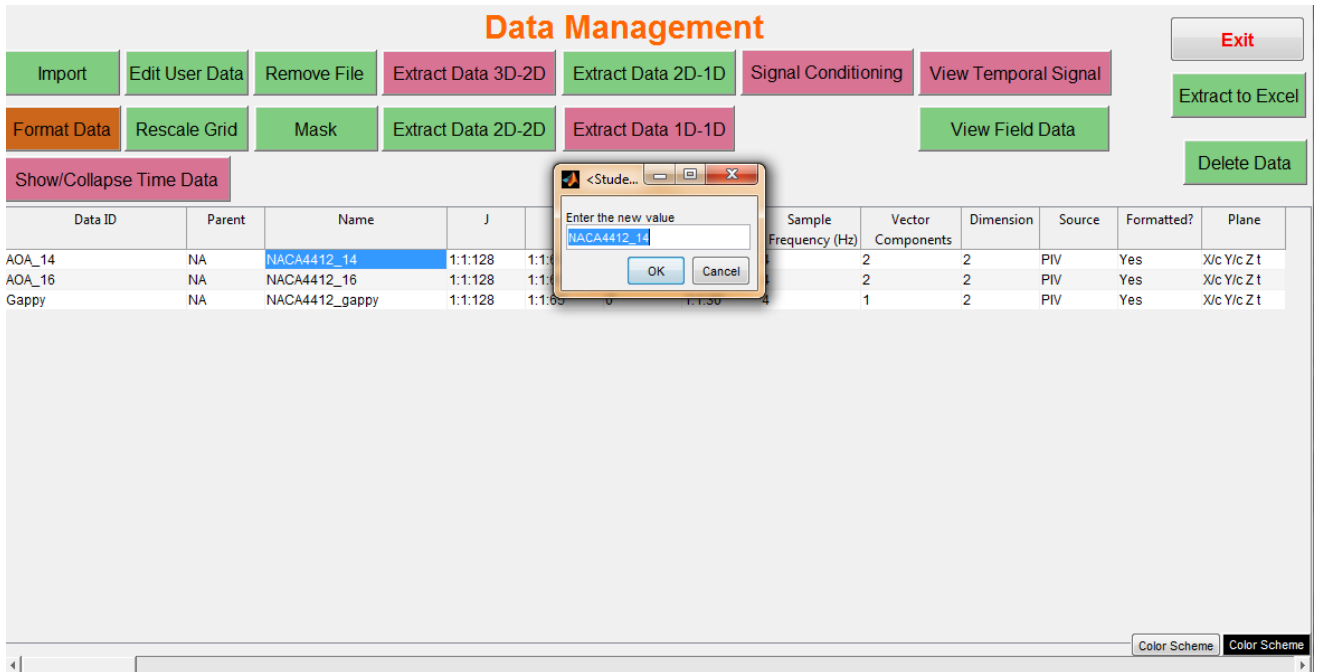


Figure 11. Edit the User Data

4. Rescale Grid:

The user can change the scaling of the x and y coordinates. This could be useful to get normalized units, for example X/D. To do this the user can select the Name cell of the data to be scaled and push the “Rescale Grid” button.

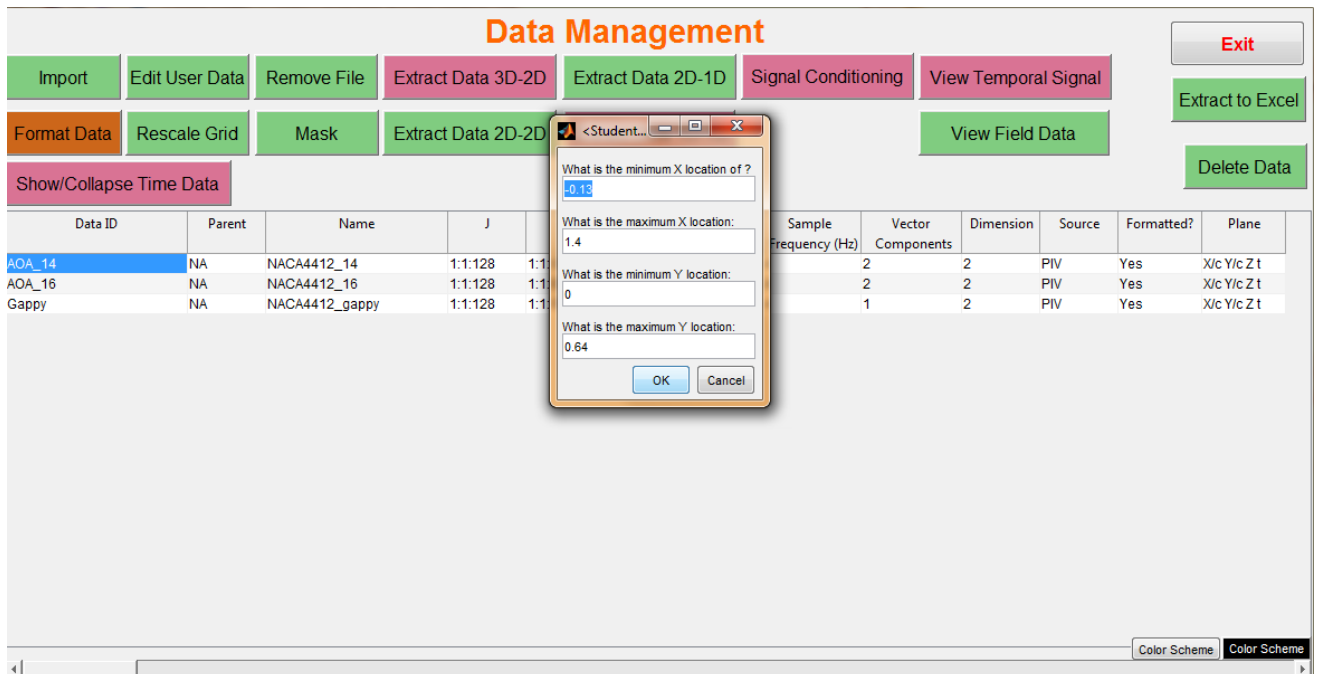


Figure 12. Rescale the Grid Data

The user enters max and min values and the grid is scaled accordingly. (This assumes a uniformly spaced grid.) The user then pushes the “ok” button when done.

5. Remove File:

The user can remove specific snapshots from a data set if they need to. The user needs to select the Name cell of the desired data set and push the ”Remove File” button. The user will enter the snapshot numbers that are to be removed and press “ok”. This process is permanent, so make sure this is what you really want to do.

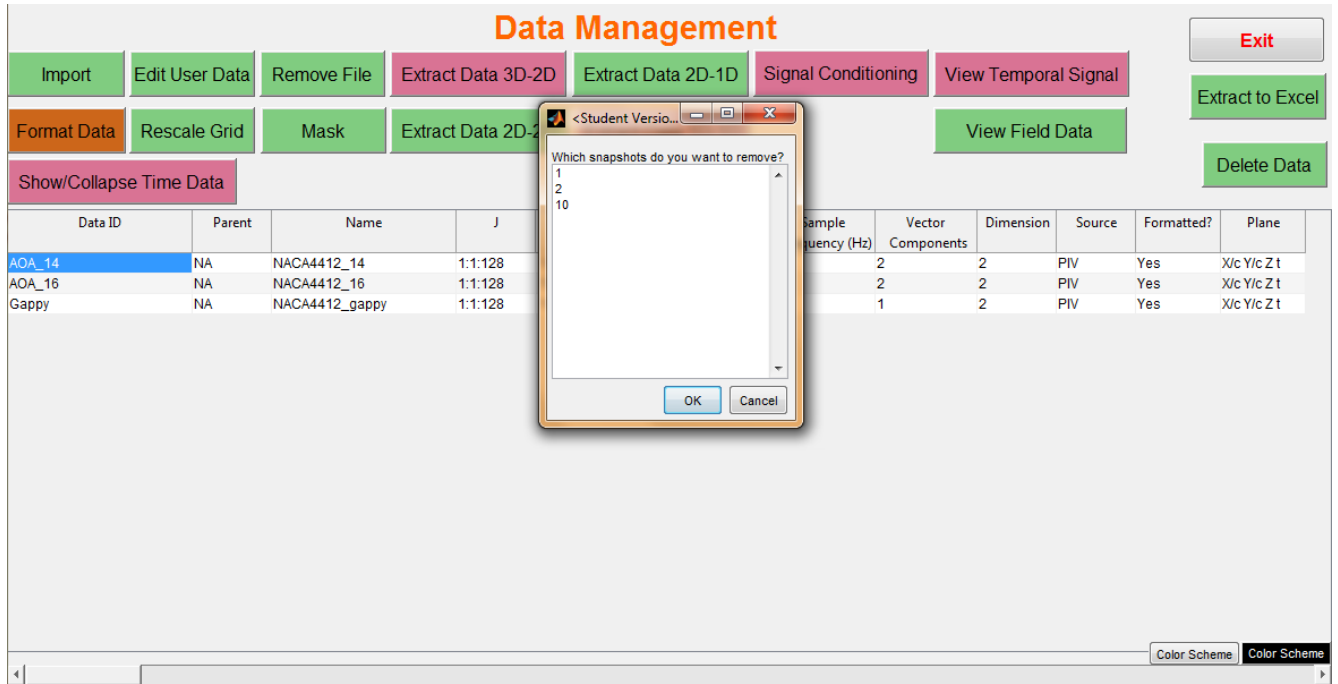


Figure 13. Remove Snapshots from a Data set

6. Mask

Sometimes data can have portions that are undesirable for some reason. The data could have been caused by a reflection, or a solid body could show up as zeros in the data. This tool allows the user to create a mask. This mask tells the program to ignore the masked data points when performing analyzes and when plotting the data.

To create a mask, select a data set on the data management screen. If the data set is a candidate for a mask (2 spatial dimensions) the “Mask” button will turn green. Press the “Mask” button and an image of the data will be displayed with a cursor. Place the cursor at one of the corners and left click to place the first corner. Then repeat for the other corners.

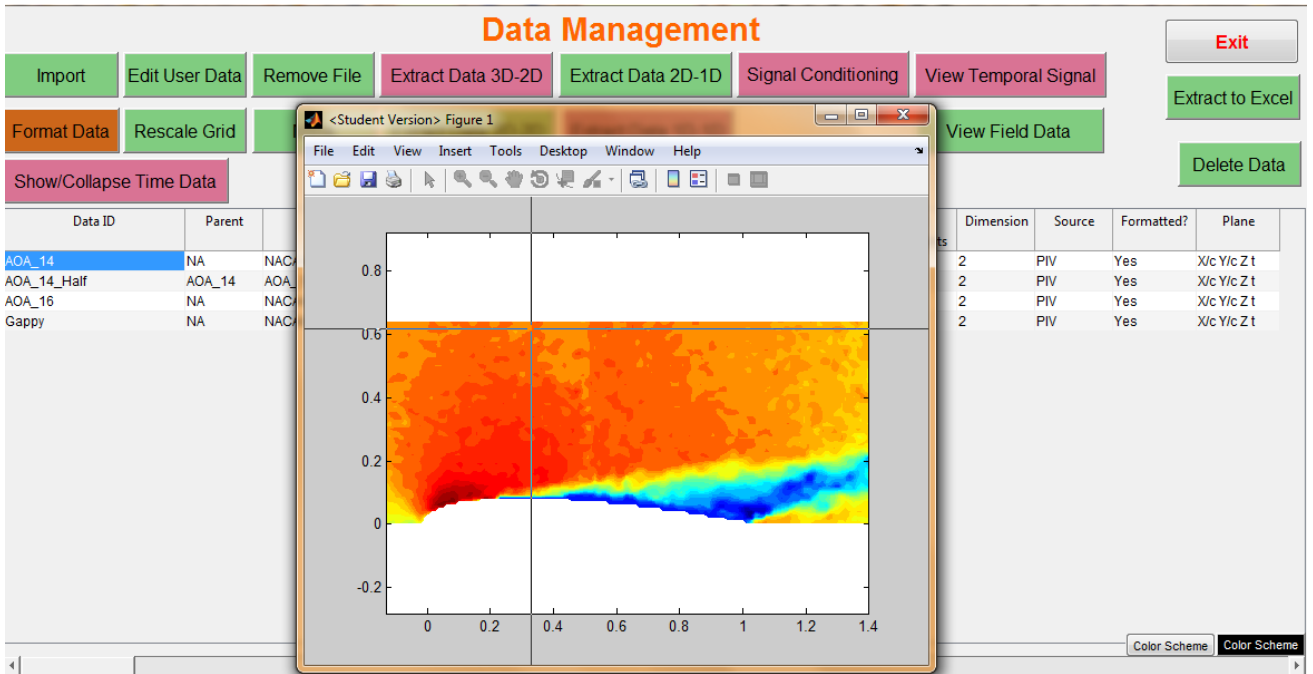


Figure 14. Create a mask for a data set (Step 1)

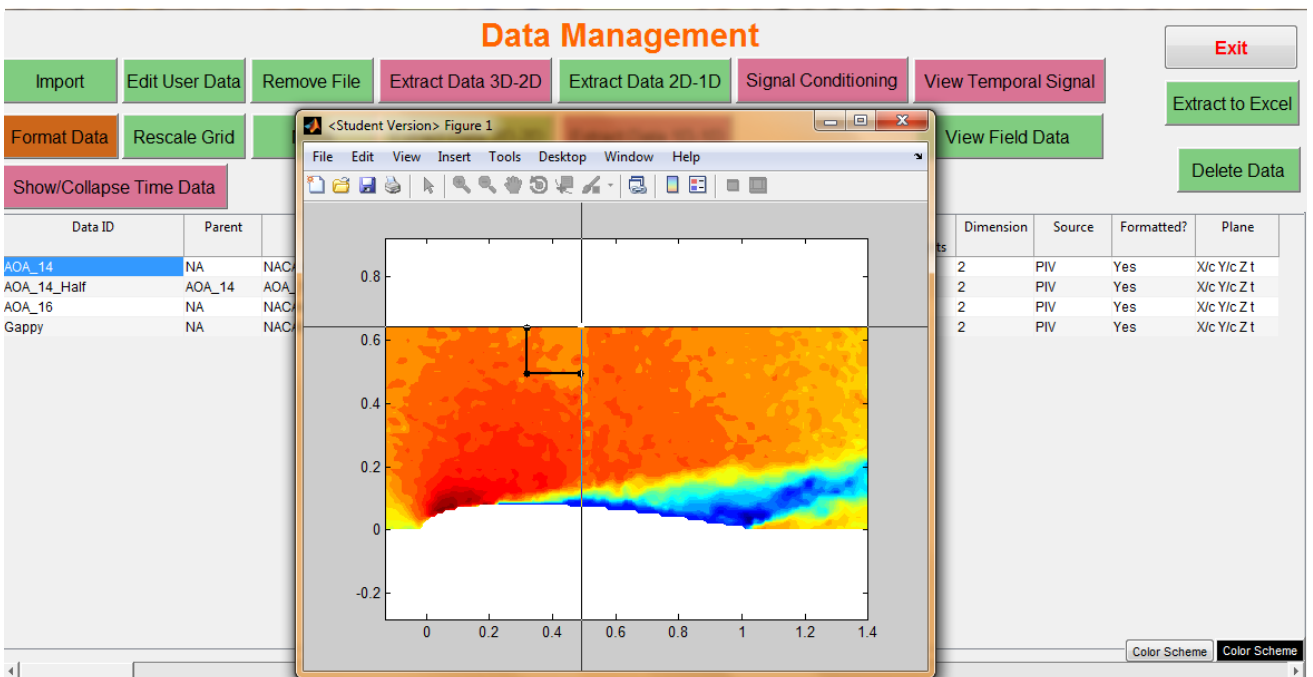


Figure 15. Create a mask for a data set (Step 2)

The last cursor is selected by using a right click (figure 16).

The Mask is then generated by selecting the points closest to the selected corners. All of the points inside the closed shape will be masked out (figure 17).

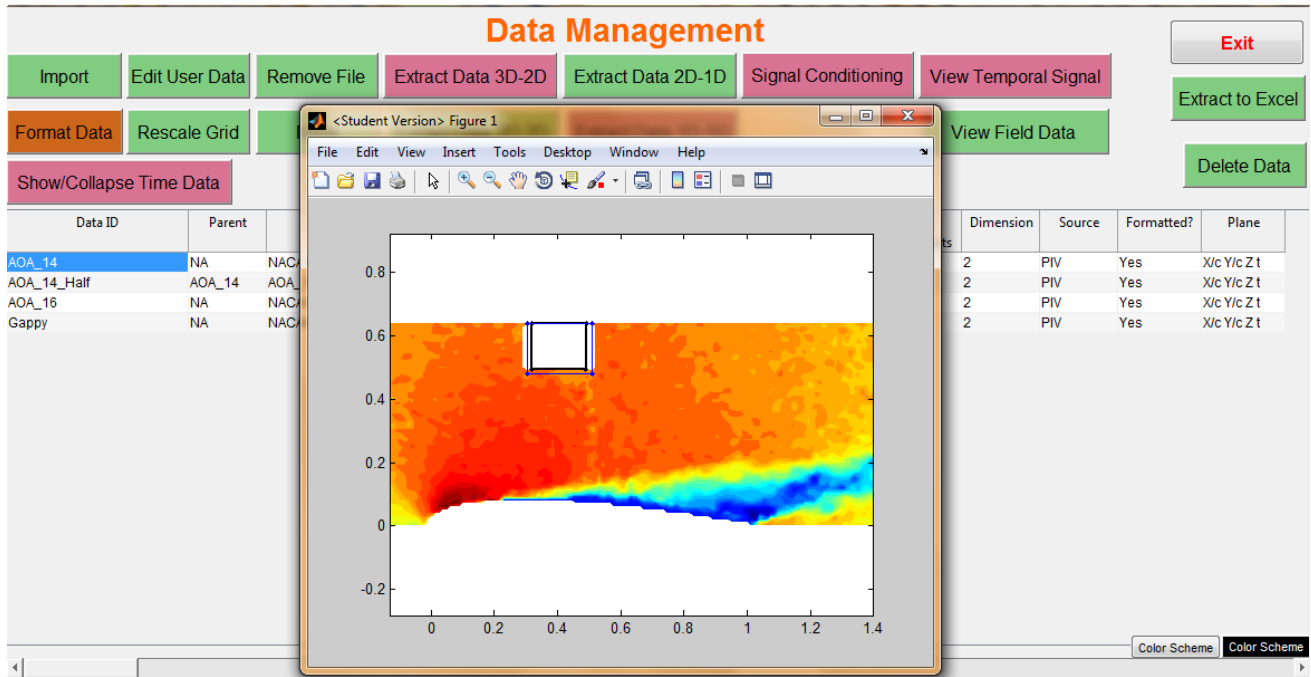


Figure 16. Create a mask for a data set (Step 3)

One should note that sometimes for a complex shape the mask may not select all of the infernal points. The process can be repeated to select the points that were not masked on the first try.

7. Extract Data 3D-2D

This tool extracts a set of planes from a set of 3D data. The GUI that performs this task is shown in figure 18. Running the extractor creates a new data set registered with ORANGE HSD.

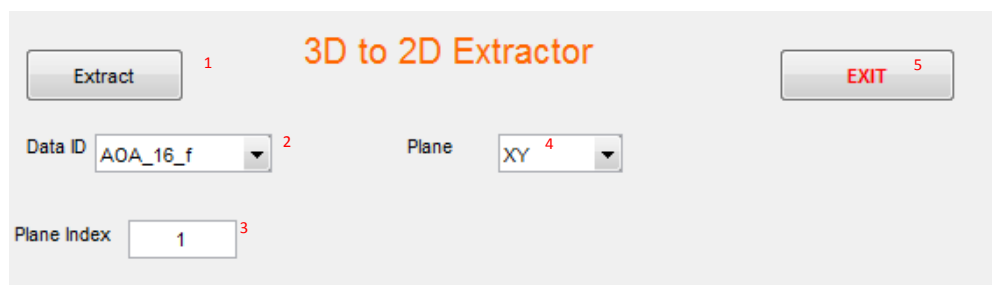


Figure 17. 3D to 2D extraction tool

1. **Extract:** Perform the extraction
2. **Data ID:** Select the data that will be extracted
3. **Plane Index:** Select the index of the plane that will be extracted
4. **Plane:** Select the plane that will be extract (i.e. XY plane)
5. **Exit:** Close the data extractor and return to the data management page

8. Extract Data 2D-2D

This tool will extract a plane of data from a larger plane of data. The size of the plane can be reduced by removing rows or columns or clipping the edges of the plane. This tool is shown in figure 19.

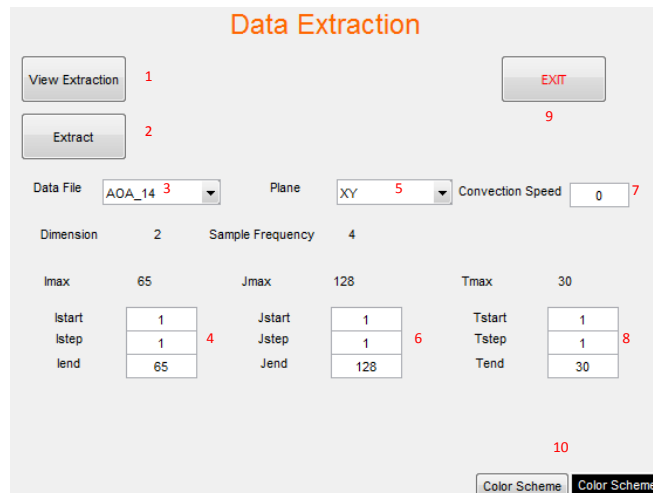


Figure 18. 2D to 2D extraction tool

1. **View Extraction:** View the extraction before spending the time to write the new data
2. **Extract:** Run the extraction program
3. **Data ID:** Select the data file to extract from
4. **I information:** Input the first location, how many points to skip, and the ending point in the I direction for the extraction
5. **Plane:** Select which plane to view (you can extract XY, XT, or YT planes)
6. **J information:** Input the first location, how many points to skip, and the ending point in the J direction for the extraction
7. **Convection Speed:** This is used to look at a plane at an angle (i.e. XY plane, but the time is changing as you move downstream)
8. **Time information:** Input the first location, how many points to skip, and the ending point in the time extraction
9. **EXIT:** Close the extractor and return to the data management page
10. **Color Scheme:** Change the color scheme

After running the extractor and returning to the data management pager one can notice the addition of a new data set; the one just created. All of the information for this data set was automatically registered in ORANGE HSD.

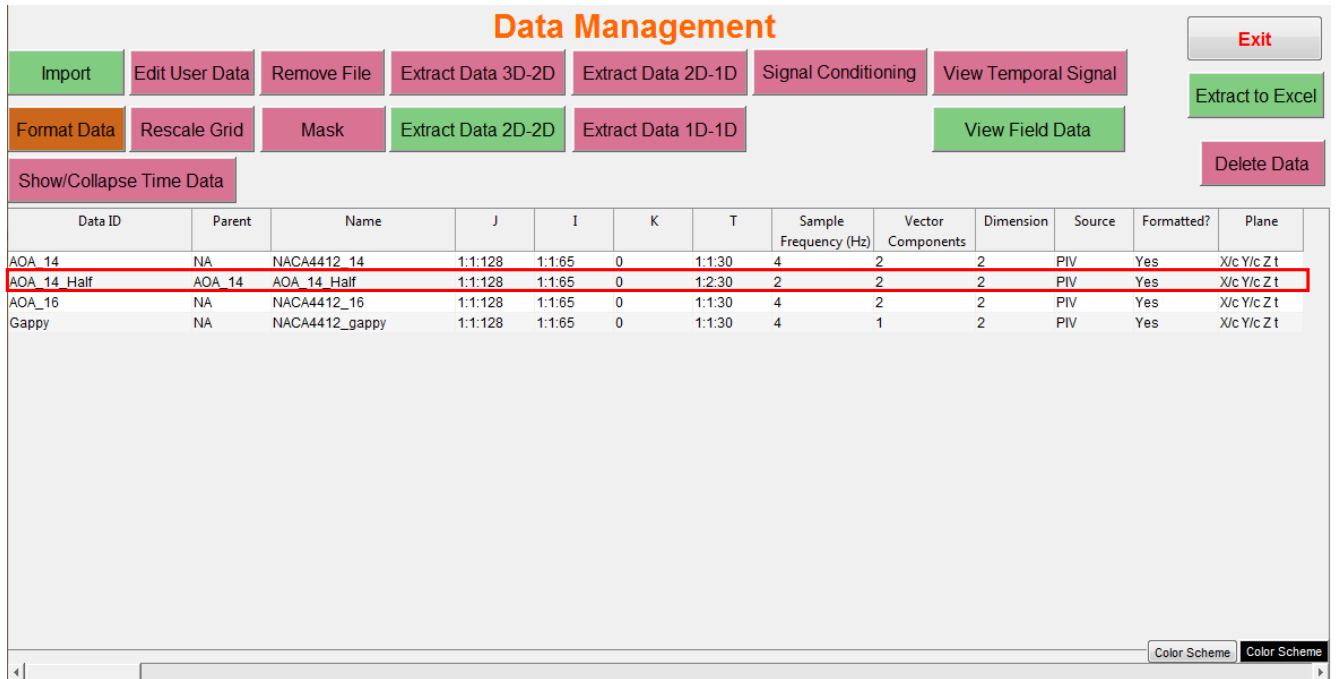


Figure 19. Data Management Page

9. Extract 2D-1D

The 2D-1D option allows the user to pick spatial locations in a plane and extract temporal information at that point. This is shown in figure 21. The user must select the Name cell of the data set to extract from and push the “Extract Data 2D-1D” button.

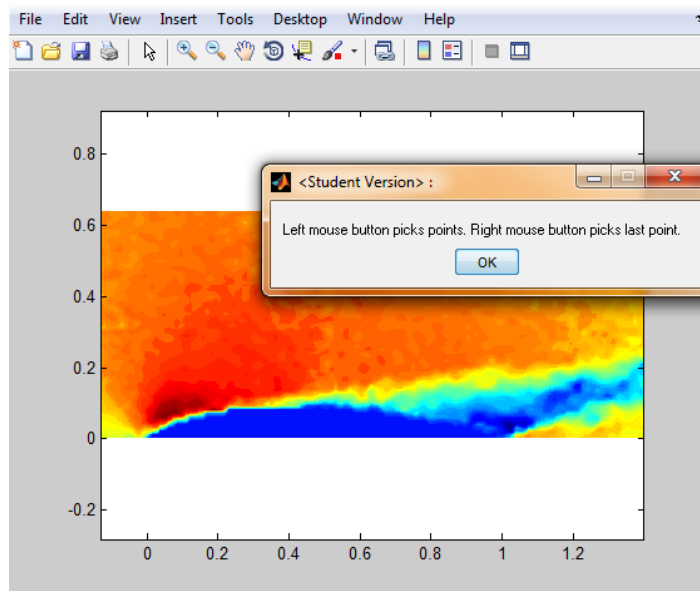


Figure 20. Extract Temporal Information

In order to extract the points click on the “ok” button and then left mouse click where you want to extract a tempo-

ral signal. You can repeat this as many times as desired. After each click, a window will pop up asking if the selected point is ok. At this point the select “yes” if the point is the desired point, “no (repick)” if the point is not what is desired and repick the point, or “no (enter)” to manually input the coordinates. The last point desired should be chosen with the right mouse click. This will exit the sampling window and write the data.

After returning to the data management screen one can notice that a new data set was added. This is the temporal data that was just extracted.

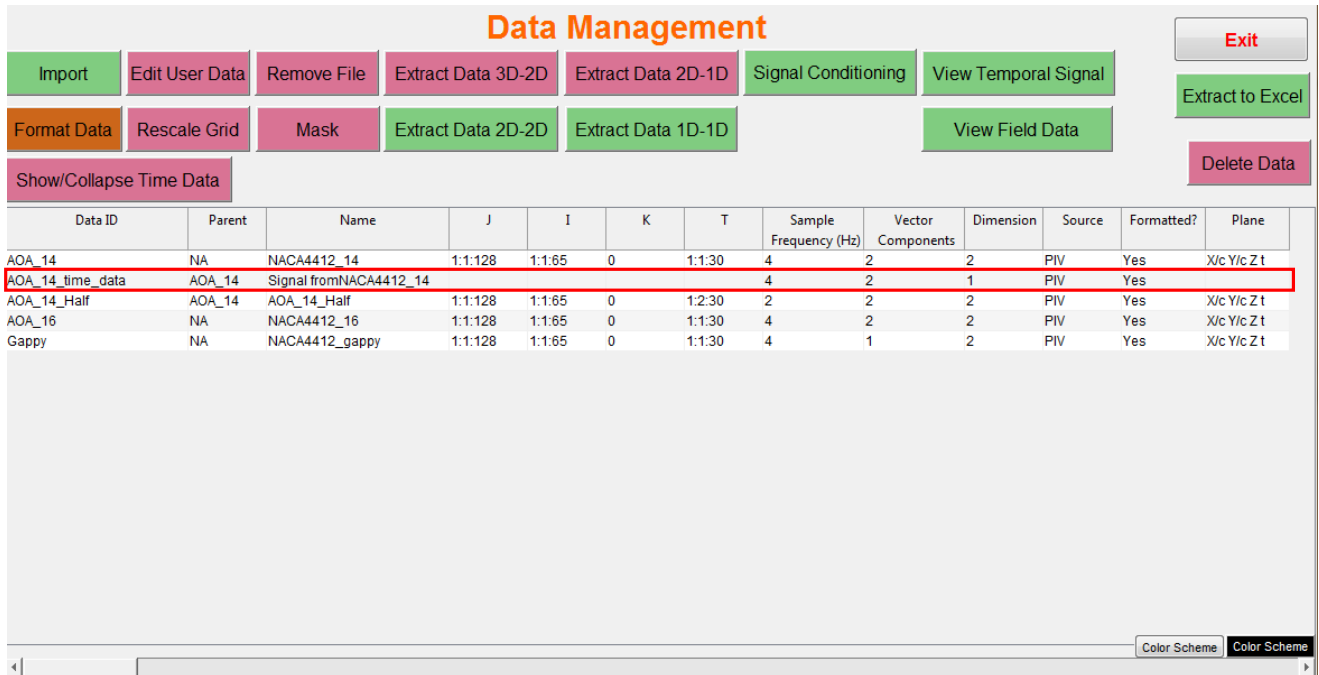


Figure 21. Data Management Screen

10. 1D-1D Extraction

This tool is used to extract data from time series. This can be used to downsample or extract a portion of a signal. The tool is displayed in figure 23.

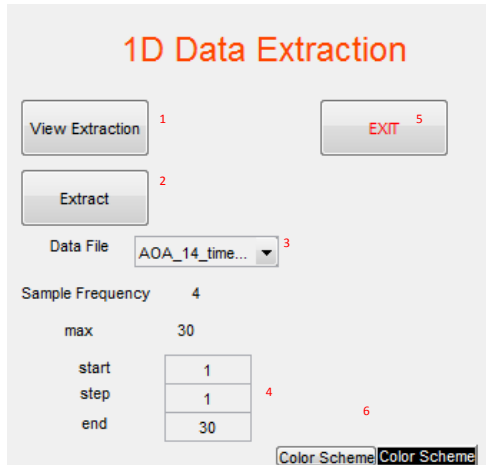


Figure 22. 1D to 1D extraction tool

1. **View Extraction:** View the extraction before spending the time to write the new data
2. **Extract:** Run the extraction program
3. **Data ID:** Select the data file to extract from
4. **Time information:** Input the first location, how many points to skip(ratio of current sampling frequency to desired sampling frequency), and the ending point in time for the extraction
5. **EXIT:** Close the extractor and return to the data management page
6. **Color Scheme:** Change the color scheme

Once again, this creates a new entry on the data management page, which represents the newly created data.

Data ID	Parent	Name	J	I	K	T	Sample Frequency (Hz)	Vector Components	Dimension	Source	Formatted?	Plane
AOA_14	NA	NACA4412_14	1:1:128	1:1:65	0	1:1:30	4	2	2	PIV	Yes	X/c Y/c Z/1
AOA_14_time_data	AOA_14	Signal from NACA4412_14					4	2	1	PIV	Yes	
Half_time	AOA_14_time_data	Signal from NACA4412_14					2	2	1	PIV	Yes	
AOA_14_Half	AOA_14	AOA_14_Half	1:1:128	1:1:65	0	1:2:30	2	2	2	PIV	Yes	X/c Y/c Z/1
AOA_16	NA	NACA4412_16	1:1:128	1:1:65	0	1:1:30	4	2	2	PIV	Yes	X/c Y/c Z/1
Gappy	NA	NACA4412_gappy	1:1:128	1:1:65	0	1:1:30	4	1	2	PIV	Yes	X/c Y/c Z/1

Figure 23. Data Management Screen

11. signal Conditioning

This tool is used to condition a signal. The tool is used to filter and smooth signals. The tool is displayed in figure 25.

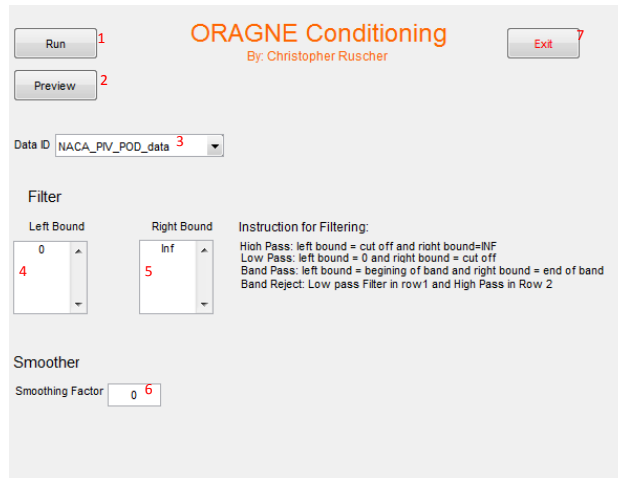


Figure 24. Signal Conditioning

1. **Run:** Perform the desired conditioning
2. **Preview:** look at the conditioned signal before saving the data
3. **Data ID:** Select the data to condition
4. **Filter Left Bound:** Select the lowest frequency in the range (it is possible to use multiple entries)
5. **Filter Right Bound:** Select the highest frequency in the range (it is possible to use multiple entries)
6. **Smoothing Factor:** The larger this number the more smoothing that will be applied to the signal
7. **Color Scheme:** Change the color scheme

This will also create a new entry on the data management screen.

12. View Temporal Signal

This tool is used to view signals such as time series. The tool is displayed in figure 26.

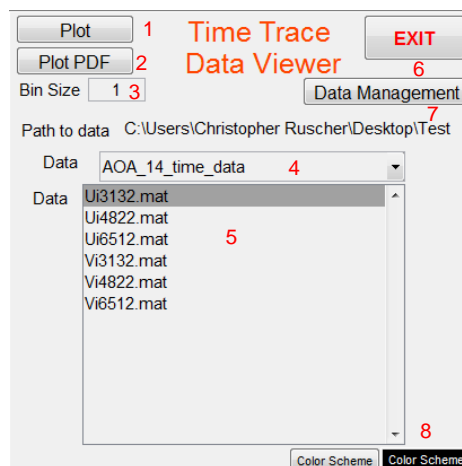


Figure 25. Temporal Viewer

1. **Plot:** Plot the selected data
2. **Plot PDF:** Creates a bar plot showing the distribution of a signal
3. **Bin Size:** This sets the size of the bins used to plot the distribution bar plot
4. **Data ID:** Select the data to view
5. **Data:** Select the data file to view
6. **EXIT:** Close the viewer and return to the previous page
7. **Data Management:** View the data management screen
8. **Color Scheme:** Change the color scheme

13. View Field Data

This tool is used to view field data such as velocity contours. The tool is displayed in figure 27.

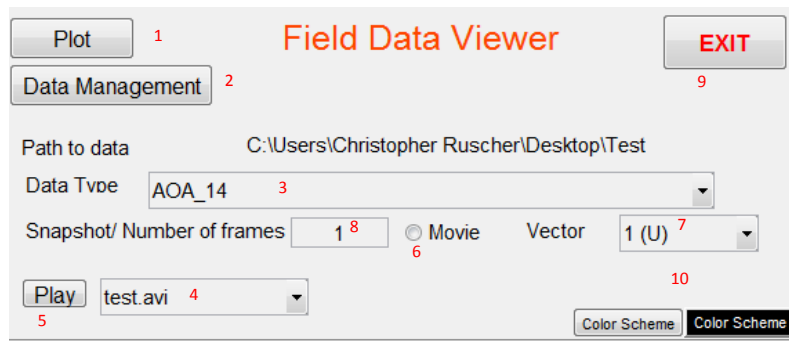


Figure 26. View Field Data

1. **Plot:** Plot the selected data
2. **Data Management:** View the data management screen
3. **Data ID:** Select the data to view
4. **Movie Select:** Select a movie to play if one exists
5. **Play:** Play the selected movie (This will open the default video player)
6. **Movie:** Select this if you want to create a movie
7. **Vector Index** Select the vector component to plot
8. **Snapshot Number:** Plot the selected snapshot or the number of frames if creating a movie
9. **EXIT:** Close the viewer and return to the data management page
10. **Color Scheme:** Change the color scheme

When the user presses the “Plot” button, the user will be asked a few questions. The user is asked if instantaneous or fluctuation contours should be plotted (if the average has been calculated).

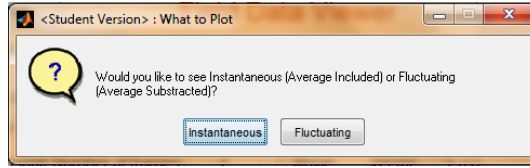


Figure 27. Instantaneous or Fluctuating

The user is then asked what to use as a maximum and minimum value, the number of contours, and the color scheme to use.

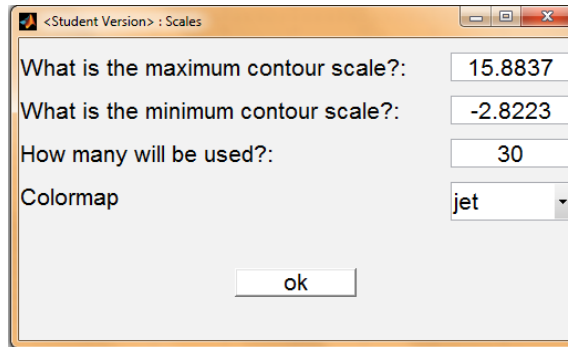


Figure 28. Scaling the Contour Plots

The windows shown in figure 28 and 29 are used throughout ORANGE HSD when creating contour plots.

14. Show/Collapse Time Data:

The user can use the “Show/Collapse Time Data button” to show the signals that are stored in a data set.

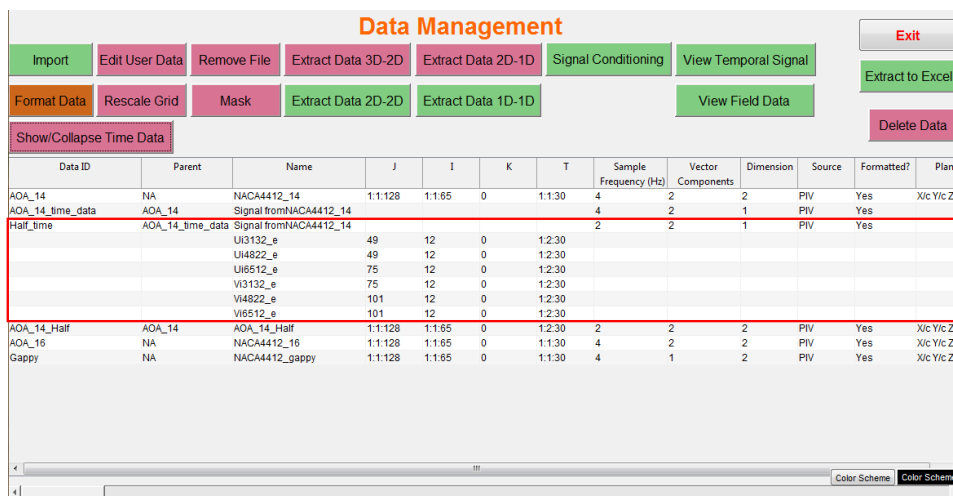


Figure 29. Data Management Screen

After finishing with the data management page, the user will go back to the main screen. One can notice that even more options are now available.

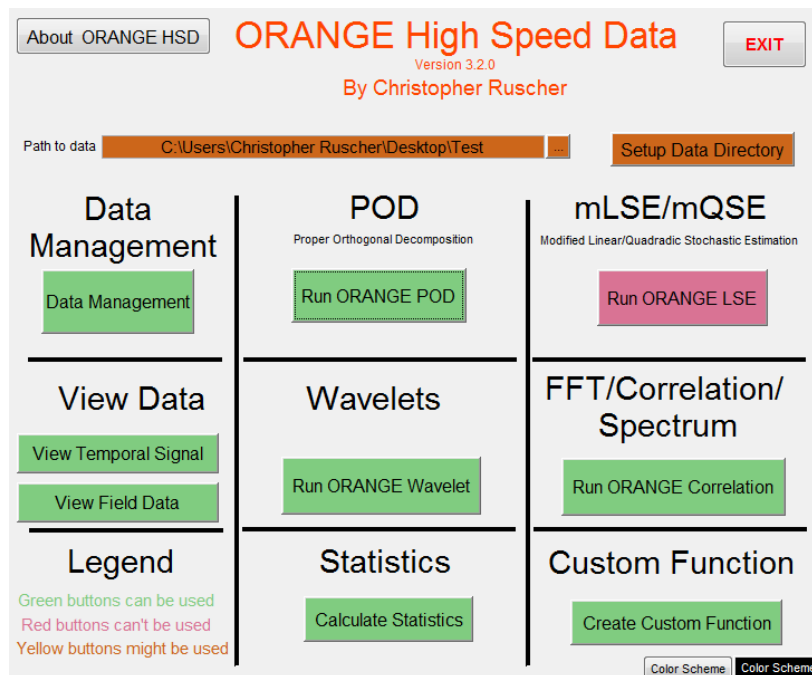


Figure 30. High Speed Data Main GUI Screen

B. ORANGE POD

Let's move on to ORANGE POD. ORANGE POD is a tool that will perform Proper Orthogonal Decomposition (POD). POD is a tool that is used to decompose a data set into a set of optimal basis functions. The basis functions are optimal in the sense of the mean square error. POD is discussed in more depth in section IV subsection A. It is highly recommended that a user unfamiliar with POD read this section!

1. Capabilities of ORANGE POD

Here is a list of the current capabilities of ORANGE POD

- Compute the basis functions and POD coefficients
- Plot the basis functions
- Plot the time dependent POD Coefficients
- Plot reconstructed velocity image
- Create a reconstructed velocity movie
- Create a low dimensional data set
- Perform Snapshot, Classic, Split, and Gappy POD

2. Using Orange POD

A screen shot of the ORANGE POD GUI is displayed in figure 32.

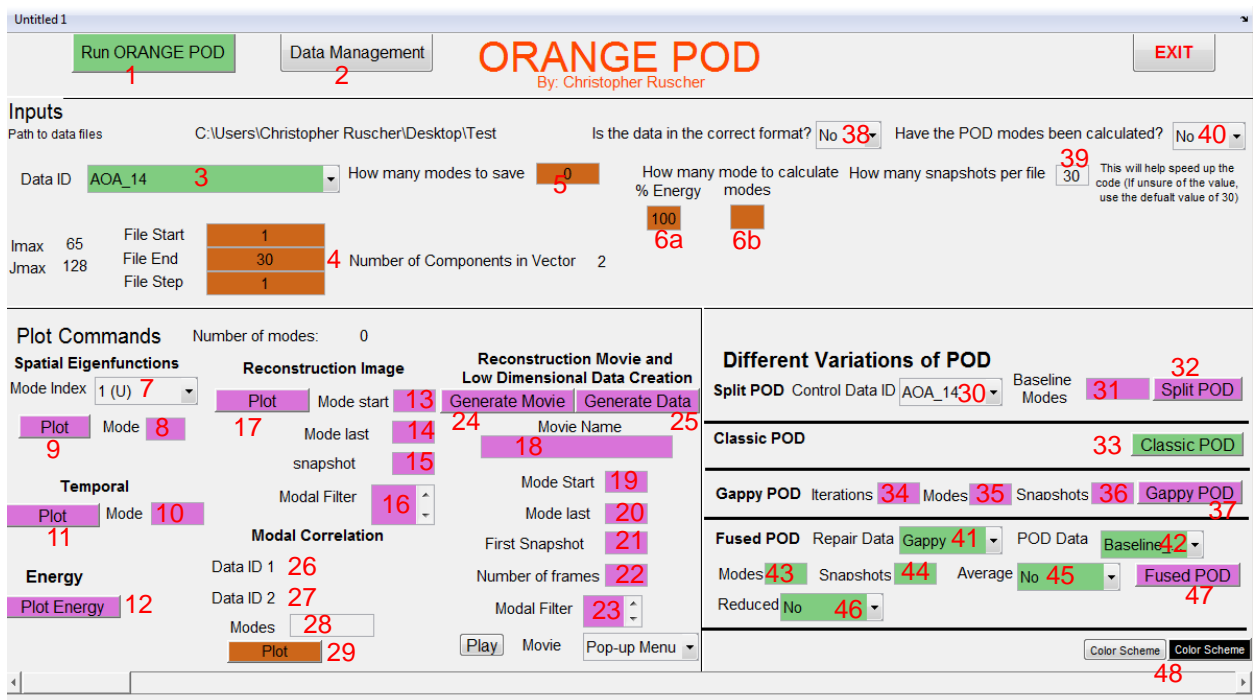


Figure 31. ORANGE POD GUI

1. **Run ORANGE POD:** This button calculates the basis functions and coefficients
2. **Data Management:** This button opens the data management page
3. **Data ID:** Select the data set that POD will be performed on
4. **Number of Snapshots:** The user can input the number of snapshots to be used in the calculations. The default is to use all of the snapshots
5. **Modes to Save:** Enter the number of modes that you want to save (Use this if you know you want to look at the first N modes)
6. **How many modes to save:** a.) Enter the percent of energy for the calculated modes; or b.) Enter the number of modes to calculate. ***This helps large data sets run faster and usually the higher modes are not of interest
7. **Mode Index:** The user can select from the drop down menu which modes to look at the modes (U component, V component or W component).
8. **Mode Number:** Enter the mode number to plot (Mode 0 is the mean)
9. **Plot:** This will plot the selected spatial mode
10. **Temporal Mode Number:** Input the mode number to be plotted
11. **Temporal Plot:** Plot the selected time dependent POD coefficient
12. **Plot Energy:** Plot the energy convergence
13. **Mode Start:** Input the first mode to use in the reconstruction

14. **Mode Last:** Input the last mode to use in the reconstruction
15. **Snapshot Number:** Input the snapshot that will be reconstructed
16. **Modal Filter:** Input the modes in the selected range that are not desired in the reconstruction
17. **Reconstruction Plot:** Plot the reconstruction
18. **Movie Name:** Input a name for the reconstruction movie file
19. **Mode Start:** Input the first mode to use in the reconstruction
20. **Mode Last:** Input the last mode to use in the reconstruction
21. **First Snapshot:** Input the first snapshot to use in the reconstruction
22. **Number of Frames:** Input the number of snapshots to be used in the reconstruction movie
23. **Modal Filter:** Input the modes in the selected range that are not desired in the reconstruction
24. **Generate Movie:** This generates a reconstruction movie
25. **Generate Data:** This generates a new data set based on the movie criterion.
26. **Data ID 1:** Select a POD data set
27. **Data ID 2:** Select another POD data set
28. **Modes:** Input the number of modes to compare
29. **Plot:** Pressing the plot button creates a contour plot showing the correlation between the two sets of modes.
30. **Control Data ID:** Select the control data if you are performing Split POD
31. **Baseline Modes:** Select the number of baseline modes to be used in the split POD
32. **Split POD:** Perform Split POD
33. **Classic POD:** This will perform classic POD on the data set selected in Data ID
34. **Iterations:** Number of iteration for the Gappy POD process
35. **Modes:** Number of modes to use in the Gappy POD process
36. **Snapshots:** Number of snapshots to use in the Gappy POD process
37. **Gappy POD:** This button will run Gappy POD and create and new estimate of the data using Gappy POD
38. **Correct Format:** This is used if the data has already been processed for POD. This should generally be set to "No"
39. **Snapshots per file:** Enter how many snapshots are processed at one time. ***This helps speed up the code. The higher the number the faster the code will perform. However, this number is limited by memory.
40. **POD modes Calculated:** This is used if the modes have been processed. This is generally set to "No". This can be used if more spatial modes need to be plotted than were originally plotted(not used anymore).
41. **Repair Data:** This is the data set that has missing or corrupt data. The data set must have a "gappy_mask" file.
42. **POD Data:** This is the data that will be used to repair the corrupted data.
43. **Modes:** The number of modes to use in the repair process.
44. **Snapshots:** The number of snapshots to be repaired. This will always start with the first snapshot and repair the number of snapshots desired
45. **Average:** If this is yes the average will not be subtracted from the data set.

46. **Reduced:** If this is yes, the repair process will repair a reconstructed data set using the selected amount of modes. (The POD must have been performed on the data set)
47. **Fused POD:** Perform fused POD
48. **Color Scheme:** This will change the color scheme

3. *Running ORANGE POD*

Below is set of instructions to run ORANGE POD after the installation process.

1. Open Matlab
2. Type “High_Speed_Data” in the command window
3. This will open the main screen(figure 1).
4. Select the data set you are working with
5. Perform any formatting that needs to be done
6. Click the “Run ORANGE POD” button
7. This will open ORANGE POD
8. Select the data ID to be processed (3 in figure 32)
9. Button 1 in figure 32 should turn green if POD has not been performed yet and brown if POD has been performed on the data already
10. Press the “Run ORANGE POD” button (1 in figure 32)
11. This can take a long time depending on the size of the data set.

4. *Outputs*

Below is a list of the important outputs from ORANGE POD.

- **“Data ID”_POD_data:** ORANGE POD will create this folder in order to store the data that will be output by the code. This folder will be located in the project directory
- **CSnapPOD.mat:** This file is the kernel of the eigenvalue problem used to determine the basis function. It is located in the POD_data directory for the correct data type
- **mode#.fig:** This is a Matlab figure of the basis function. These are located in the POD_data directory for the correct data type. This file is created when one views the basis function using the button 9.
- **mode#.tif:** This is a .tif figure of the basis function. These are located in the POD_data directory for the correct data type. This file is created when one views the basis function using the button 9.
- **PhiU.mat:** This contains the basis functions in the i direction. The different columns correspond to the mode number. It is located in the POD_data directory for the correct data type
- **PhiV.mat:** This contains the basis functions in the j direction. The different columns correspond to the mode number. It is located in the POD_data directory for the correct data type
- **aPOD.mat:** This is the time dependent POD coefficients. Each column corresponds to the mode number. It is located in the POD_data directory for the correct data type
- **Energy.mat:** This contains the relative energy contained in each mode. It is located in the POD_data directory for the correct data type

When done using ORANGE POD press the exit button to return to the main screen. One will notice that ORANGE LSE is now usable if you performed POD on a data set.

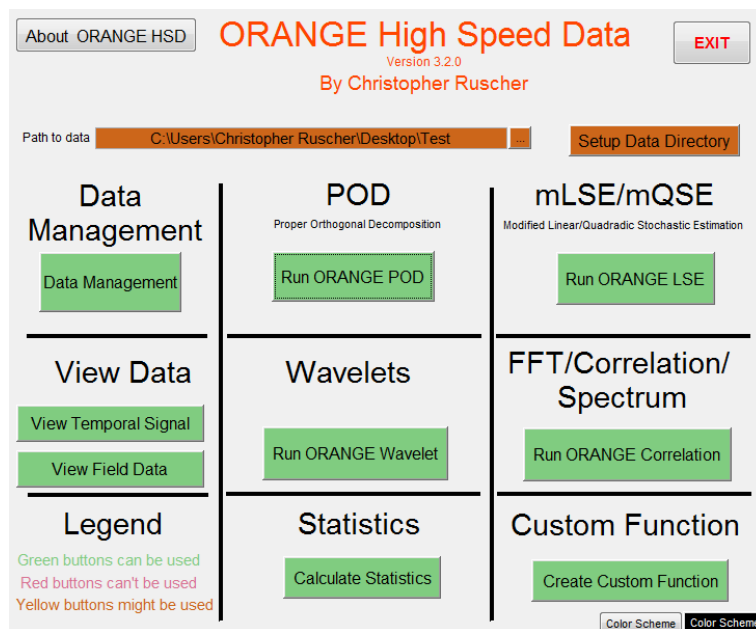


Figure 32. ORANGE HSD Main Screen

Let's now look at the stochastic estimation tool.

C. ORANGE LSE

ORANGE LSE is code that will perform modified linear stochastic estimation (mLSE), and modified quadratic stochastic estimation (mQSE). This technique estimates the field from a few points and is useful for flow control applications, but can also be useful in processing “slow” data. For example, the velocity field can be estimated from a pressure signal in order to get time dependent velocity information if PIV data is taken simultaneously with pressure. It is highly recommended that a user unfamiliar with the technique read section IV subsection B to understand the basics of stochastic estimation before using the code.

1. Capabilities of ORANGE LSE

Here is a list of the current capabilities of ORANGE LSE:

- **Perform stochastic estimation:** The user can use mLSE and mQSE
- **Plot the coefficients** The user can plot the coefficients that are used for each pressure sensor
- **Plot Temporal Estimates:** The user can plot the temporal POD coefficient from the estimate and compare it with the original in spectral and temporal space.
- **Reconstruction:** The user can create a static image or movie of the estimated field. This is compared with the POD reconstruction and the original field.
- **Estimate the field:** The user can take a time signal and use it to estimate the field.

2. Using ORANGE LSE

A screen shot of ORANGE LSE is shown below in figure 34.

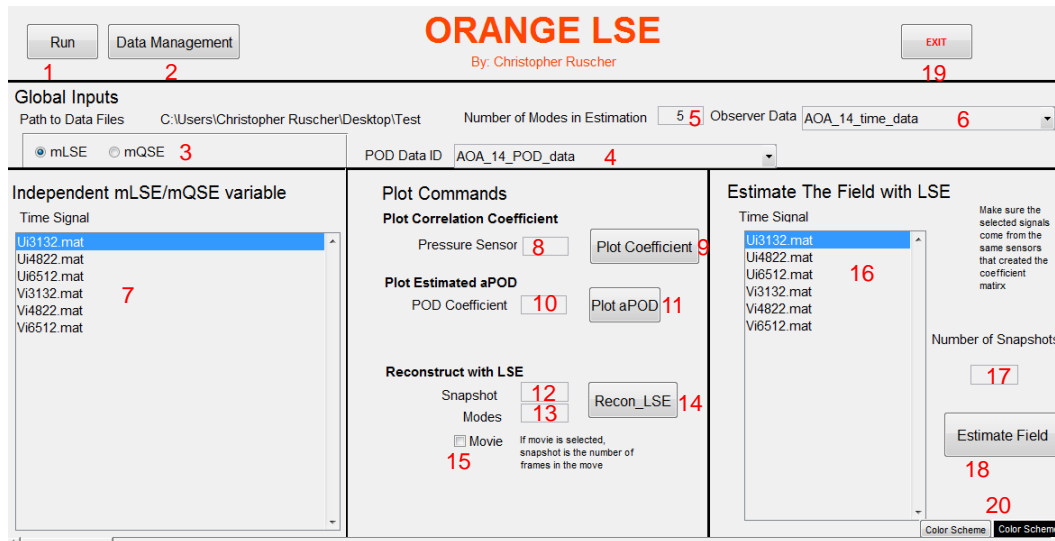


Figure 33. ORANGE LSE Screen Shot

1. **Run:** This runs ORANGE LSE after the all inputs have been entered
2. **Data Management:** This opens the data management window
3. **mLSE mQSE sLSE sQSE:** Click the appropriate bubble for mLSE or mQSE
4. **POD Data ID:** Use the drop down list to select which POD coefficients estimate
5. **Number of Modes:** Input the number of modes to be estimated.
6. **Observer Data ID:** Use the drop down list to select the conditional signals to estimate the POD coefficients
7. **Time Signals:** Select which time signals will be used in the estimation
8. **Pressure Sensor:** Enter the sensor number for the coefficients that will be plotted
9. **Plot Coefficient:** Press this button to generate the coefficient plot
10. **POD Coefficient:** Input the mode number to be plotted
11. **Plot aPOD:** Press this button to plot the time dependent coefficients of the estimate and original in time and spectral domain
12. **Snapshot:** Enter the snapshot that will be reconstructed. If the movies option is selected (15) this is how many frames will be in the movie
13. **Modes:** This is how many modes will be used in the reconstruction.
14. **Recon_LSE:** Pressing this button will create a static image or movie of the estimate, POD Reconstruction, and Original field
15. **Movie:** If this box is checked, a movie of the reconstruction will be created.
16. **Time Signal:** Enter the number of the time trace that will be used for the estimation. This does not have to be the same time trace as 7, but must be from the same sensor.

17. **Number of Snapshots:** Enter the number of snapshots to be estimated.
18. **Estimate Field:** Press this button to write velocity files created by stochastic estimation. These files will be registered in the data management tool
19. **EXIT:** Exit ORANGE LSE and return to the main menu
20. **Color Scheme:** Change the color scheme of ORANGE LSE

The signal files should be down sampled to match the aPOD time trace if it is going to be used to generate the coefficient matrix. This can be done in the data extraction tools that were discussed earlier. If the signal is being used to estimate the field, the sampling frequency does not have to match the aPOD signal.

3. *Running ORANGE LSE*

Below is a set of instructions to run ORANGE LSE.

1. Open Matlab
2. Type “High_Speed_Data” in the command window
3. This will open the main menu
4. Select the desired data folder
5. Perform any formatting that needs to be done
6. Compute the POD coefficients using ORANGE POD
7. Click the “Run ORANGE LSE” button
8. This will open ORANGE LSE
9. select the POD Data ID (4 in figure 34)
10. Enter the number of modes to be estimated (5 in figure 34)
11. Select the data type to estimate with (6 in figure 34)
12. Select the whether to mLSE, mQSE, sLSE, or sQSE (3 in figure 34)
13. Select the sensors to be used (7 in figure 34) ***The sensors used in the estimation should have a high correlation
14. Press “Run” to begin the analysis.

4. *Outputs*

Below is a list of the important outputs from ORANGE LSE.

- **“Data ID”_LSE_QSE:** This is a folder that is created and contains the outputs from ORANGE LSE. This is contained in the project directory.
- **A.mat:** This is the coefficient matrix that is output by the ORANGE LSE. This is located in the LSE_QSE directory inside the project directory.
- **a_LSE.mat:** This is the estimate of the time dependent POD coefficients. This is located in the LSE_QSE directory inside the project directory.
- **Data_ID__LSE_number of modes_Number of sensors:** A new folder is created for the estimated data when using button 18 in figure 34

When the user exits the ORANGE LSE, the main screen will appear again. The user can the choose to perform more analyses on the data. Let’s now move to ORANGE Wavelet.

D. ORANGE Wavelet

ORANGE Wavelet is a code that can perform a wavelet transform on a temporal data signal. The tool creates plots of the wavelet coefficients, which can be used to show a time-frequency relationship.

1. Capabilities of ORANGE Wavelet

- Mexican Hat Wavelet
- Morlet Wavelet
- View wavelet coefficients

2. Using ORANGE Wavelet

The gui used to run ORANGE wavelet is displayed in Figure 35.

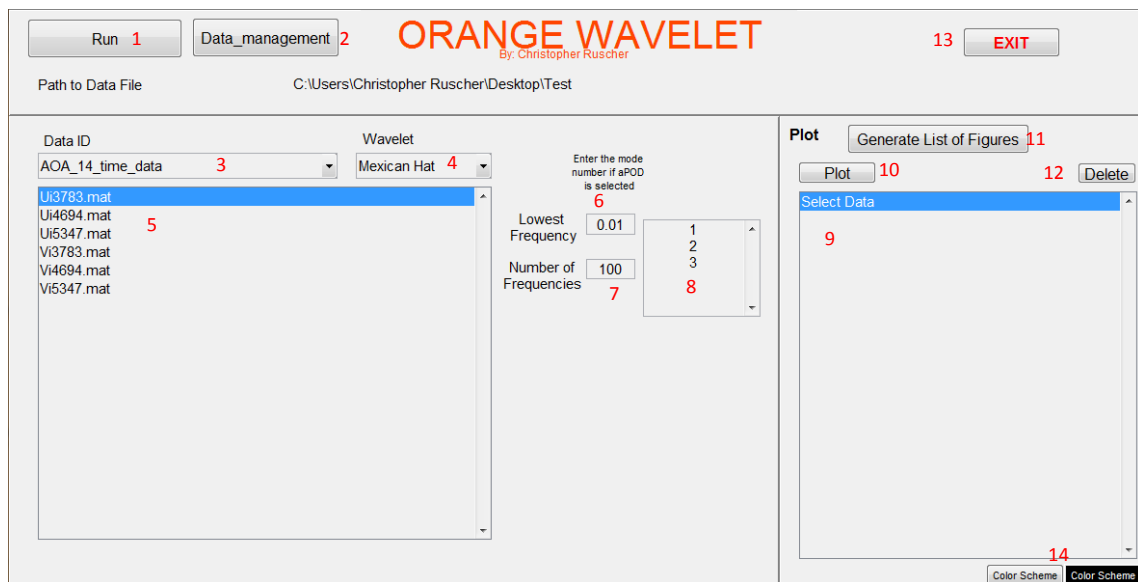


Figure 34. ORANGE Wavelet

1. **Run:** This will perform the wavelet transformation on the selected data
2. **Data Management:** This opens the data management page
3. **Data ID:** Select the data set on which wavelet analysis will be performed
4. **Wavelet Type:** Select the type of wavelet to use (currently you can use Mexican Hat or Morlet)
5. **Data Selection:** Select the signal on which wavelet analysis will be performed (You can select multiple signals and perform the analysis in batch)
6. **Lowest Frequency:** Select the smallest frequency that will be resolved (This needs to be larger than zero)
7. **Number of Frequencies:** Select how many frequencies will be resolved
8. **Mode number:** If performing wavelet analysis on the POD coefficient enter the mode numbers it will be performed on
9. **Wavelet Maps:** This will list all of the wavelet maps that have been computed. The user can select one and press plot (button 10)

10. **Plot:** This will plot the selected wavelet map
11. **Generate List of Figures:** This will generate a list of all the wavelet maps
12. **Delete:** This will delete the selected wavelet map
13. **EXIT:** This closes ORANGE Wavelet and opens the main screen
14. **Color Scheme:** Change the color scheme

3. *Running ORANGE Wavelet*

To run ORANGE Wavelet, follow the directions below:

1. Open Matlab
2. Type “High_Speed_Data” in the command window
3. This will open the main menu
4. Select the data folder
5. Perform any formatting that needs to be done
6. Click the “Run ORANGE Wavelet” button
7. This will open ORANGE Wavelet
8. Select the Data ID (3 in figure 35)
9. Select the files that will be processed (5 in figure 35)
10. Enter the lowest frequency to be measured (6 in figure 35)
11. Enter the number of frequencies to be used between the cut off and lowest frequency (7 in figure 35)
12. If these are time dependent modes, enter the mode numbers to be processed (8 in figure 35)
13. Press the run button (1 in figure 35)
14. Once finished processing the data, you can view the plots by selecting the plot to be viewed (9 in figure 35)
15. Then Press the plot button (10 in figure 35)

4. *Outputs*

Below is a list of the important outputs from ORANGE Wavelets.

- **Wavelet_data:** This is a folder that is created and contains the outputs from ORANGE Wavelet. This is contained in the project directory.
- **Mex_Wavelet_X.fig:** This is a Matlab figure file that contains the wavelet map and time trace where “X” is a description of the time trace that was put through the Wavelet analysis. This is located in the Wavelet_data directory inside the project directory.
- **Mex_Wavelet_X.mat:** This is a Matlab data file that contains the wavelet coefficients where “X” is a description of the time trace that was put through the Wavelet analysis. This is located in the Wavelet_data directory inside the project directory.
- **Morlet_Wavelet_X.fig:** This is a Matlab figure file that contains the wavelet map and time trace where “X” is a description of the time trace that was put through the Wavelet analysis. This is located in the Wavelet_data directory inside the project directory.
- **Morlet_Wavelet_X.mat:** This is a Matlab data file that contains the wavelet coefficients where “X” is a description of the time trace that was put through the Wavelet analysis. This is located in the Wavelet_data directory inside the project directory.

When done the user can press the Exit button and return to the main menu and select a different option. Let’s look at ORANGE Correlate.

E. ORANGE Correlate

ORANGE correlate is the tool to perform correlations, fast Fourier transforms (FFT), and spectrum

1. Capabilities of ORANGE Correlate

- FFT
- Auto and cross correlations
- Auto and cross spectrum
- plot multiple sets of data on a single axes

2. Using ORANGE Correlate

The gui to use ORANGE correlate is shown in figure 36.

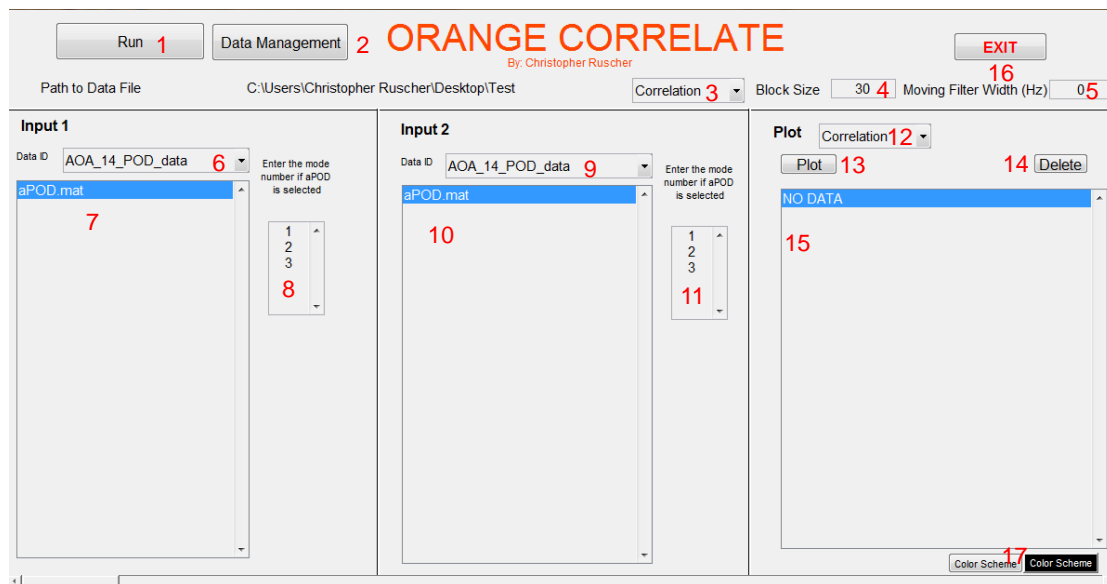


Figure 35. ORANGE Correlate

1. **Run:** This button will run the selected analysis
2. **Data Management:** This will bring up the data management page
3. **Analysis Type:** Select Correlation, Spectrum, or FFT from the drop down menu
4. **Block Size:** Enter the size of the block you will use for block averaging (This defaults to the size of the signal)
5. **Moving Filter Width:** This is used to smooth the output if the record length is too small. It will average the data in over the width of the filter
6. **Data ID 1:** Select the data that will be analyzed
7. **Signal Selection :** Select the first signal that will be analyzed (more than one can be selected)
8. **Mode number 1:** If using POD coefficients, enter the mode numbers here
9. **Data ID 2:** Select the data to be analyzed

10. **Signal Selection 2:** Select the second set of signals that will be analyzed (This is only for correlations and spectrum)
11. **Mode number 2:** If using POD coefficients, enter the mode numbers here
12. **Plot Analysis:** Select the analysis that will be plotted (This matches item 3)
13. **Plot:** Plot the data from the selected analyses
14. **Delete:** Delete the selected data
15. **Plot Selection:** Select what to plot (more than one can be selected)
16. **EXIT:** This will close the ORANGE Correlate window and open the main menu
17. **Color Scheme:** Change the color scheme

3. *Running ORANGE Correlate*

To run ORANGE correlate follow the directions below:

1. Open Matlab
2. Type “High_Speed_Data” in the command window
3. This will open the main menu.
4. Select the data folder
5. Perform any formatting that needs to be done
6. Click the “Run ORANGE correlate” button
7. This will open ORANGE Correlate
8. Select the analysis to be performed (3 in figure 36)
9. Select the Data ID (6 in figure 36)
10. Select the files that will be processed (7 in figure 36)
11. If these are time dependent modes, enter the mode numbers to be processed (8 in figure 35)
12. Select the data set to correlate against if performing correlations or spectrum (9 in figure 36)
13. Select the data files to correlate against if performing correlations of spectrum (10 in figure 36)
14. If these are time dependent modes, enter the mode numbers to be processed (11 in figure 36)
15. Press the run button (1 in figure 36)
16. Once finished processing the data, you can view the plots by selecting the plot to be viewed (you may select as many as you want (15 in figure 36)
17. Plot the data (13 in figure 36)

4. Outputs

Below is a list of the important outputs from ORANGE Correlate.

- **Correlation** This is a directory inside the project directory that contain all of the correlation data
- **Correlation_name_name1.mat:** This file contains the correlation between name and name1. It is located in the directory Correlation
- **Spectrum** This is a directory inside the project directory that contain all of the spectrum data
- **Spectrum_name_name1.mat:** This file contains the spectrum between name and name1. It is located in the directory Spectrum
- **FFT** This is a directory inside the project directory that contain all of the FFT data
- **fft_name.mat:** This file contains the FFT of name. It is located in the directory FFT

The user can exit ORANGE Correlate and return to the main screen and use another tool. Let's look at the statistics tool

F. ORANGE Statistics

The ORANGE HSD tool includes a statistics tool. This tool calculates the average, RMS, skew, and kurtosis of field data. The tool also shows a convergence of the various statistics. The tool is shown in figure 37.

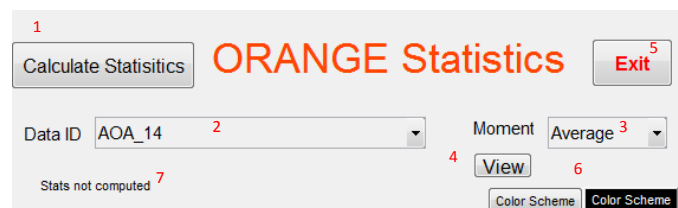


Figure 36. Statistics Tool

1. **Calculate Statistics:** This will calculate the statistics for the selected data set
2. **Data ID:** Select the data for which the statistics will be calculated
3. **Moment:** Use this to select which statistic to calculate. One can chose Average, RMS, Skewness, Kurtosis, or all. This is also used to determine which statistic to view.
4. **View:** View the selected statistic
5. **EXIT:** Close the statistics screen and return to the main menu
6. **Color Scheme:** Change the color scheme

1. Running ORANGE Statistics

1. Open Matlab
2. Type "High_Speed_Data" in the command window
3. This will open the main menu
4. Select the data folder
5. Perform any formatting that needs to be done

6. Push the “Calculate Statistics” button (this opens ORANGE Statistics)
7. Select the Data ID (2 in figure 37)
8. Select which statistic to calculate (3 in figure 37)
9. Push “Calculate Statistics” (1 in figure 37)

The user can exit ORANGE Statistics and return to the main screen and select another tool. Let’s look at the custom function tool.

G. Custom Functions

The ORANGE HSD tool also comes with a custom function tool. This can be used to calculate different functions for the data sets. (i.e. curl) The tool is shown in figure 38.

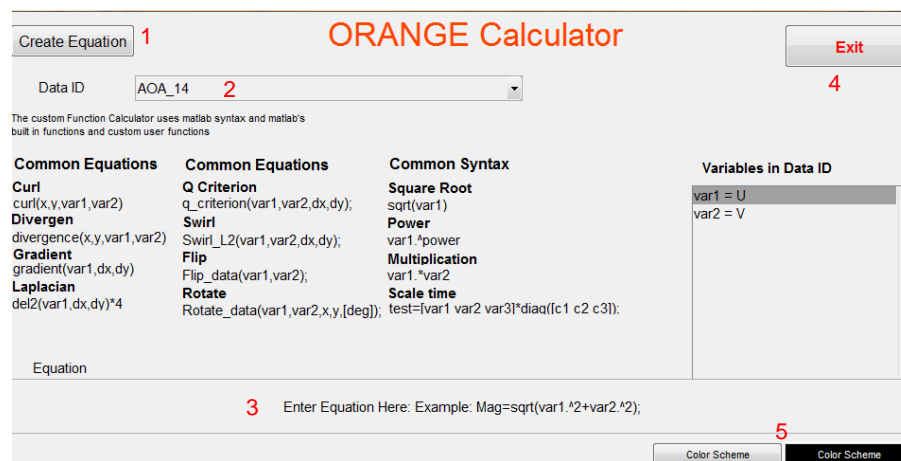


Figure 37. Custom Function

1. **Create Equation:** This will calculate the custom function on the desired data set
2. **Data ID:** Select the data that the calculator will use
3. **Custom Function:** Enter the custom function in this location using Matlab syntax
4. **EXIT:** Close the ORANGE Calculator screen and return to the main menu
5. **Color Scheme:** Change the color scheme

1. Running ORANGE Calculator

1. Open Matlab
2. Type “High_Speed_Data” in the command window
3. This will open the main menu
4. Select the data folder
5. Perform any formatting that needs to be done
6. Push the “Create Custom Function” button (this opens ORANGE Calculator)
7. Select the Data ID (2 in figure 38)

8. Input you equation into the equation box (3 in figure 38)
9. Press the “Create Equation” button (1 in figure 38)
10. This will create a new data set that is stored in the data management tool

2. Creating a New Function

A new function can be created and placed in the ORANGE calculator folder in the subfunction folder in the ORANGE HSD folder. This new function can then be used by the custom function tool. See `q_criterion.m` for an example of how to do this.

The above section describes the tools included in ORANGE HSD and how to use them. A video tutorial will also be available to show the execution of the code.

We now will briefly discuss the theory behind some of the tools.

IV. Theory

A. Proper Orthogonal Decomposition

POD is a reduced order modeling technique first introduced to fluid dynamics by Lumley in 1967.¹ It is basically a Karhunen-Loeve expansion or principle component analysis depending on the field the method is being used. This can be used for a variety of applications, anywhere from turbulence¹ to facial recognition.² POD is a way of building a basis for a flow that is based on the energy content of the flow. POD will create bases based on energy content unlike Fourier analysis, for example, where the basis is arbitrarily chosen to be a combination of sines and cosines. The first mode will be the most energetic and the energy content of the higher modes will continue to decrease. This method is used to capture the structures that have the most energy but not necessarily the dynamics of the flow field.

The basis is created by maximizing the mean square projection of the velocity³ or in other words trying to find the best fit for your data set. The maximization of this becomes an eigenvalue problem that must be solved to determine the basis. There are many different variations of POD, some that can be used to study flow control such as split POD,⁴ Filtered POD⁵ which filters out higher frequency noise, snapshot POD⁶ which is a modification to classical POD¹ used in order to decrease the size of the eigenvalue problem for data sets with large amounts of spatial resolution. The method for performing Classical POD, Snapshot POD, split POD, and Gappy POD will be explained in the following sections.

1. Classical POD

As stated above, the maximizing of the mean squared projection of the velocity creates an eigenvalue problem that can be seen below in equation 1.

$$\int R_{ij}(x, x') \phi_j^{(n)}(x) dx' = \lambda^n \phi_i^{(n)}(x) \quad (1)$$

Where ϕ is the eigenvector, λ is the eigenvalue (energy), and $R_{ij}(x, x')$ is the time averaged spatial two-point velocity correlation give by equation 2. The two point correlation is a tensor that includes the autocorrelations for all the components in the data field as well as the cross correlations between the components.

$$R_{ij}(x, x') = \langle u_i(\vec{x}, t_o), u_j(\vec{x}', t_o) \rangle \quad (2)$$

In equation 2 u is the velocity of the flow or any other variable depending on what is being studied, t_o is time, and x is the position vector. The kernel for this problem $R_{ij}(x, x')$ can be written as a block matrix shown in equation 3.

$$\begin{bmatrix} \langle u(\vec{x}, t_o), u(\vec{x}, t_o) \rangle & \langle u(\vec{x}, t_o), v(\vec{x}, t_o) \rangle & \langle u(\vec{x}, t_o), w(\vec{x}, t_o) \rangle \\ \langle v(\vec{x}, t_o), u(\vec{x}, t_o) \rangle & \langle v(\vec{x}, t_o), v(\vec{x}, t_o) \rangle & \langle v(\vec{x}, t_o), w(\vec{x}, t_o) \rangle \\ \langle w(\vec{x}, t_o), u(\vec{x}, t_o) \rangle & \langle w(\vec{x}, t_o), v(\vec{x}, t_o) \rangle & \langle w(\vec{x}, t_o), w(\vec{x}, t_o) \rangle \end{bmatrix} \quad (3)$$

Solving for the eigenvalues and eigenvectors for equation 3 will give the spatial modes for the flow. With the spatial modes, the POD coefficients can be found by projecting the velocity fields onto the spatial modes. This can be seen in equation 4.

$$a_n(t) = \int u_i(\vec{x}, t) \phi_i^{(n)}(\vec{x}) d\vec{x} \quad (4)$$

With the POD modes and the POD coefficients, the flow can be reconstructed. If all of the POD modes are used in the reconstruction, the flow field is reconstructed completely. The reconstruction is shown below in equation 5.

$$u_i(\vec{x}, t) = \sum_{n=1}^{Nm} a_n(t) \phi_i^{(n)}(\vec{x}) \quad (5)$$

2. Snapshot POD

Another method of POD that is a modification of Classical POD is called Snapshot POD. This method was introduced in 1987 by Sirovich.⁶ The method transforms the eigenvalue problem from dimension of the number of spatial points multiplied by number of components to dimension of temporal points or snapshots. This method is preferred over Classical POD when analyzing a large spatial data set, such as two camera PIV³ or large eddy simulation (LES) or direct numerical simulation (DNS) data sets. This methods starts first by calculating the temporal eigenfunctions using equation 6.

$$\int C(t, t') a_n(t') dt' = \lambda^{(n)} a_n(t) \quad (6)$$

In equation 6, $C(t, t')$ is defined in equation 7 and $a_n(t)$ are the temporal modes or time dependent POD coefficients. You can notice in equation 7 that the kernel $C(t, t')$ is not the summation of the autocorrelations of the velocity components in the flow. This means that the eigenvalue problem is no longer coupled. This coupling will be taken care of when the spatial modes are calculated. The uncoupling of the eigenvalue problem along with doing a temporal correlation instead of a spatial correlation reduces the problem size significantly for data sets with a large amount of spatial resolution.

$$C(t, t') = \frac{1}{T} \int u_i(\vec{x}, t) u_i(\vec{x}, t') d\vec{x} \quad (7)$$

In equation 7, T is the total number of snapshots and u is the velocity field.

In order to ensure the modes are the same for both Snapshot and Classical POD, equation 8 is used to scale the temporal modes correctly.

$$\langle a_n \cdot a_m \rangle = \delta_{mn} \lambda^{(m)} \quad (8)$$

Equation 8 basically states that the temporal modes should be orthogonal (which they are by definition) and have a magnitude equal to the square root of the eigenvalues, which is consistent with Classical POD.

Equation 9 is used to get the spatial modes. By projecting each component onto the temporal POD coefficients, the spatial POD modes can be found. This takes care of the coupling as mentioned earlier.

$$\phi^{(n)}(\vec{x}) = \frac{1}{T \lambda^{(n)}} \int a_n(t) u_i(\vec{x}, t) dt \quad (9)$$

The spatial eigenfunctions are orthogonal as are the spatial eigenvalues for Classical POD.

3. Split POD

Another variant of POD is known as Split POD.⁴ In this version of POD, one can use two different data sets and find modes that correspond to differences in the data sets. This has been used by Camphouse *et. al.* to determine the effects of flow control. The idea for this came from the fact that the controller is adding very little energy and the control information will be dominated by baseline information. Using the split POD procedure, the baseline information can be removed from the control data. The procedure is as follows: Perform POD on the baseline data. This will give basis functions $\Phi^{(b)}$. The control data (secondary data set) will then be projected onto the baseline basis functions.

$$b_{ij} = \langle U_i, \Phi_j^{(b)} \rangle \quad (10)$$

U_i are snapshots from the secondary data set and \langle , \rangle denotes an inner product.

The part of the secondary data that is not contained in the baseline data is extracted.

$$\bar{U}_i = U_i - \sum_{j=1}^{M_b} b_{ij} \Phi_j^{(b)} \quad (11)$$

\bar{U}_i is the part of the secondary data not contained in the baseline information, and M_b is the number of baseline modes to use.

POD is then performed on the new data (\bar{U}_i). This gives basis functions for information not contained in the baseline data. The secondary data can now be expressed in terms of both baseline and secondary information.

4. Gappy POD

Gappy POD was created by Everson and Sirovich in 1995⁷ to handle “gappy” data. Gappy data is data that has gaps or missing information for one reason or another. The technique estimates the missing information. Everson and Sirovich first demonstrated this technique using marred photographs and repairing them with Gappy POD. Specifically, they showed missing data from human faces could be determined from a basis formed by similar types of faces. They also observed that if not careful, errors can arise. This was shown by attempting to repair an image of a monkey’s face with a human face basis. Gappy POD was also used for aerodynamic applications by Bui-Thanh *et. al.*⁸ to reconstruct a flow field based on airfoil surface data and was used by Ruscher *et. al.*⁹ to perform POD on a set of snapshots with an airfoil moving through the image.

Gappy POD works by performing POD on non-gappy data. The data with missing information is then reconstructed using a linear combination of the POD basis calculated from the non-gappy data. The coefficients of the linear combination are then determined by solving a least squares problem that minimizes the difference between the linear reconstruction and the gappy data. The data will be repaired correctly if the data is contained in the span of the original non-gappy data.

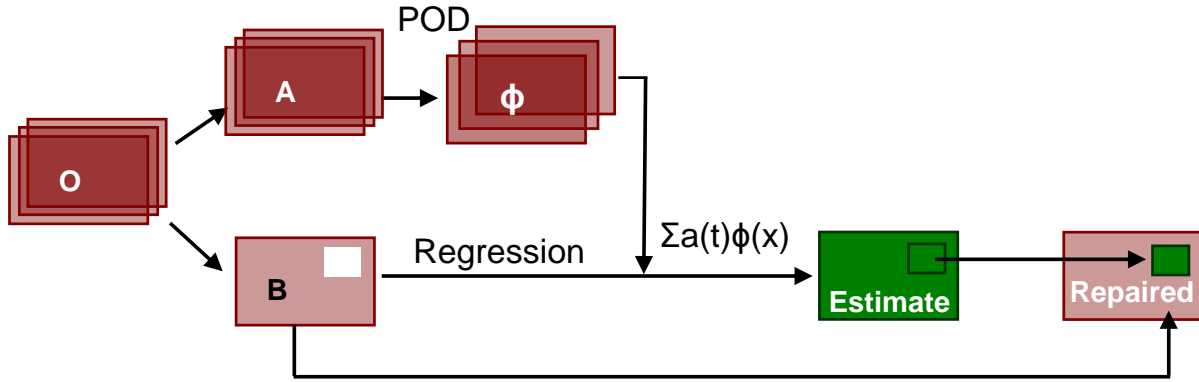


Figure 38. Diagram of the Gappy POD Method

The math behind Gappy POD is presented below. First, the gappy points must be located and stored in a mask vector. The mask vector is defined such that if the data is missing the mask is zero and if the data is not missing it is one.

$$mask_i^t = \begin{cases} 1 & \text{if data is not missing} \\ 0 & \text{if data is missing} \end{cases} \quad (12)$$

With the mask vector, a “gappy inner product” is defined as:

$$\langle u, v \rangle_{gappy} = \langle \langle u, mask \rangle, \langle v, mask \rangle \rangle \quad (13)$$

In equation 13, u and v are two vectors. The estimate of the gappy data is then defined as a linear combination of the non-gappy modes.

$$\tilde{g} = \sum_{i=1}^n b_i \phi^{(i)} \quad (14)$$

In equation 14, b is the linear combination coefficient, ϕ is the POD basis from the non-gappy data and \tilde{g} is the POD estimated gappy data. Then the error between the estimate and original is minimized at the non-gappy points.

$$\langle g - \tilde{g}, g - \tilde{g} \rangle_{gappy} \quad (15)$$

In equation 15, g is the original gappy data. The minimization problem is solved by taking the derivative of the system with respect to b_i and setting it equal to zero. This will produce the following linear system that can be solved for b_i .

$$\langle \phi^i, \phi^j \rangle_{gappy} b_i = \langle g, \phi_i \rangle_{gappy} \quad (16)$$

Now, the repaired data can be constructed using the following equation.

$$g = \begin{cases} g & \text{if mask} = 1 \\ \tilde{g} & \text{if mask} = 0 \end{cases} \quad (17)$$

In addition to repairing a single marred image, Gappy POD can be extended to repair a set of marred images using an iterative process (figure 40). The process works by filling in the missing data with a spatial average for each image. Then POD is performed on the images producing a POD basis. Then equation 16 is solved for each image and each image is modified using equation 17. Once this has been done for all of the images, the process is repeated until the solution has converged.

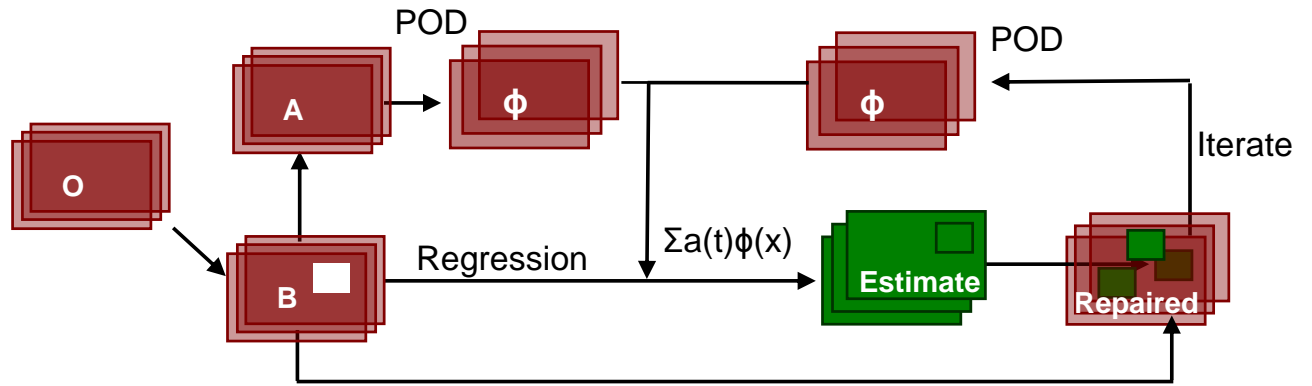


Figure 39. Diagram of the Iterative Gappy POD Method

5. Fused POD

Fused POD is a new data processing technique developed to improve aerodynamic data. It estimates missing experimental data caused by shadows, reflections, or poor data. The technique is inspired by the Gappy POD technique. The techniques both use a linear regression to determine the POD coefficients that best fit the data, so the math is quite similar.

However, the application of the techniques are very different. The Gappy POD technique uses only the corrupted data set to estimate the “bad” data. The Gappy POD procedure requires that only a few bad snapshots exist, meaning the statistics of the rest of the data are “good” enough to calculate the POD modes. Then those modes can be used to estimate the data in the “bad” snapshots. Gappy POD can also be used if all the snapshots have missing data randomly distributed in time and space. An iterative process can be used to estimate the missing data.

If you have corruption at a spatial location in all snapshots, one would not be able to use Gappy POD. This is where Fused POD is needed. In Fused POD, one obtains the POD modes from another source. The other source can be another set of data that is capturing the same phenomena or from the same data set and some physical knowledge of how the data is related. This gives statistical information about the data in the missing regions and allows estimation in that region.

The other difference between the two techniques is the implementation of a wavelet fusion process to help smooth some of the data. When filling in the data, a discontinuity can form. The discontinuity is a product of the estimated

data not fitting the actual data perfectly, and the fact the estimated data is low dimensional and missing high energy information.

A diagram of Fused POD is shown in figure 41. The technique works by calculating the POD modes from another source of data. In figure 41, the other data set is represented by CFD data though it could come from any source. The POD coefficients are then calculated using a linear regression (this is the same way Gappy POD fits the data). The estimated POD coefficients and modes are used to reconstruct the data (Green box). The estimated data is then inserted where the missing data is located. As stated before, this can cause a discontinuity. So, the reconstructed data (green data) and filled-in data (pink with green) are fused using wavelet image fusion.

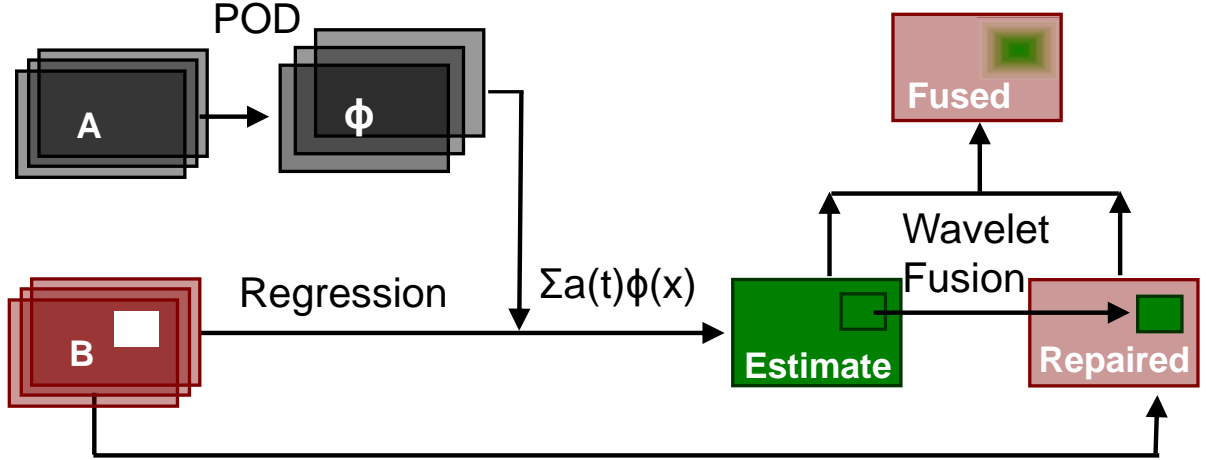


Figure 40. Fused POD Data Fusion Technique Flow Chart

B. Stochastic Estimation

Stochastic estimation is a tool first proposed by Adrian in 1977.¹⁰ In this work, Adrian show the velocity at a new point can be estimated from information at the current location. Adrian states the velocity at a point is a function of time, Δx , and conditional event E .

$$u(x, t) = F(E, x, t) \quad (18)$$

When given E , the best mean square representation is given by a conditional average.¹¹ The conditional average can be estimated as a linear combination of the conditional data. Minimizing the error between the actual velocity and the linear estimate produces the following equation as stated by Bonnet *et. al.*¹²

$$u_i(x') = A_{ij}(x')u_j(x) \quad (19)$$

In equation 19, A_{ij} is given by the following relation: $\overline{u_j(x)u_k(x)}A_{ij} = \overline{u_j(x)u_i(x')}$ where $\overline{u_j(x)u_k(x)}$ is the Reynolds stress and $\overline{u_j(x)u_i(x')}$ is the two-point correlation. This technique was expanded upon by Taylor *et. al.*¹³ Taylor's technique involves performing POD and then using another signal to estimate the POD coefficients. Then the field is rebuilt using the POD basis and estimated coefficients. The modified stochastic estimation, as it is called, requires the conditional signal be well correlated with the POD coefficients. The technique works by assuming the POD coefficients can be written as a power series expansion of the observer signal.

$$\tilde{a}_n(t) \approx A_{ni}P_i(t) + B_{nij}P_i(t)P_j(t) + C_{nijk}P_i(t)P_j(t)P_k(t) + \dots \quad (20)$$

The coefficients (A_{ni} , B_{nij} , C_{nijk} , ...) are determined by minimizing the mean square error between the estimate and actual value of the POD coefficients.

$$(\tilde{a}_n(t) - a_n(t))^2 \quad (21)$$

The power series estimate is then truncated. If only the first term is used, the estimate is known as linear stochastic estimation (LSE). If the first two terms are used, the estimate is known as quadratic stochastic estimation (QSE). Tung

et. al. shows not much information is gained from higher order estimates.¹⁴ The minimization problem reduces to a linear set of equations (LSE equation 22 and QSE equation 23).

$$\begin{bmatrix} \langle P_1 P_1 \rangle & \dots & \langle P_q P_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle P_1 P_q \rangle & \dots & \langle P_q P_q \rangle \end{bmatrix} \begin{bmatrix} A_{n1} \\ \vdots \\ A_{nq} \end{bmatrix} = \begin{bmatrix} \langle a_n P_1 \rangle \\ \vdots \\ \langle a_n P_q \rangle \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} \langle P_1 P_1 \rangle & \dots & \langle P_q P_1 \rangle & \langle P_1 P_1 P_1 \rangle & \dots & \langle P_q P_q P_1 \rangle \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \langle P_1 P_q \rangle & \dots & \langle P_q P_q \rangle & \langle P_1 P_1 P_q \rangle & \dots & \langle P_q P_q P_q \rangle \\ \langle P_1 P_1 P_1 \rangle & \dots & \langle P_q P_1 P_1 \rangle & \langle P_1 P_1 P_1 P_1 \rangle & \dots & \langle P_q P_q P_1 P_1 \rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle P_1 P_j P_k \rangle & \ddots & \langle P_q P_j P_k \rangle & \langle P_1 P_1 P_j P_k \rangle & \ddots & \langle P_q P_q P_j P_k \rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle P_1 P_q P_q \rangle & \dots & \langle P_q P_q P_q \rangle & \langle P_1 P_1 P_q P_q \rangle & \ddots & \langle P_q P_q P_q P_q \rangle \end{bmatrix} \begin{bmatrix} A_{n1} \\ \vdots \\ A_{nq} \\ B_{n11} \\ \vdots \\ B_{njk} \\ \vdots \\ B_{nqq} \end{bmatrix} = \begin{bmatrix} \langle a_n P_1 \rangle \\ \vdots \\ \langle a_n P_q \rangle \\ \langle a_n P_1 P_1 \rangle \\ \vdots \\ \langle a_n P_j P_k \rangle \\ \vdots \\ \langle a_n P_q P_q \rangle \end{bmatrix} \quad (23)$$

The stochastic estimation technique has been used for many different reasons in the fluid dynamic community. One such example is work done by Pinier *et. al.*³ where mLSE was used to estimate the time resolved velocity field around a stalled NACA4412 airfoil.

Bonnet *et. al.*¹² also used stochastic estimation. This was done to estimate the shear layer in a jet. The measurements were obtained using “X” wires. This work shows the stochastic estimation can be used for shear layer type flows.

Stochastic estimation was used by Gutmark *et. al.* to estimate the flow field in a swirling jet.¹⁵ They used stochastic estimation to reconstruct the entire field using hot wires and pressure sensors. The technique proved useful in identify large structures in the flow.

An axisymmetric jet was estimated using stochastic estimation by Tinney *et. al.*¹⁶ Using PIV measurements and pressure sensors, a full 3D time resolved estimate of the jet flow was created.

C. Wavelets

Wavelet decomposition is a technique that has been used in signal processing in order to compress signals, but has also been used as a way of analyzing different scales in turbulent flow.¹⁷ This tool has been used to extract coherent structures in turbulence such as work done by Lewalle *et. al.*¹⁸

There are many different wavelet functions . A few are the Mexican Hat wavelet

$$W(\sigma, t) = \frac{1}{2\sqrt{\pi}\sigma} e^{-\frac{t^2}{4\sigma}} \quad (24)$$

and the Morlet wavelet.

$$W(t) = (e^{2i\pi t} - e^{-z_0^2}) e^{-2\pi^2 t^2 / z_0^2} \quad (25)$$

These are two common wavelets used in the analysis of fluid dynamic data.

The math behind the wavelet transform involves convolving a signal with the wavelet function for the scales of interest.

$$\int_{-\infty}^{\infty} u(t') * W(\sigma, t - t') dt' \quad (26)$$

where u is the signal being analyzed, W is the wavelet function and σ is the scale.

Schobeiri *et. al.* uses wavelet analysis to understand the flow structure in a boundary layer on a concave plate during transition.¹⁹ The wavelet analysis was chosen for this work due to the complex nature of the transitional flow. The wavelet analysis was able to pick out the different scales present.

Schneider *et. al.* uses wavelets to show that a flow field can be decomposed into two parts, coherent structures and background structures. Schneider *et. al.* further demonstrates that this decomposition can be used to simulate a 3D mixing layer. A similar idea is discussed by Farge *et. al.*²⁰

V. Trouble Shooting

The code may at times not work in the desired way at times. Here is a list of things that can be done to fix the code.

1. Make sure the Matlab path is set to the project directory
 - (a) If the path is incorrect the code will not find the data correctly. The path can change if another error occurs
 - (b) Usually typing “cd ..” into the command window will solve the issue
2. Make sure the files are correctly named.
 - (a) If a file is created manually the names of the required files could be incorrect.
3. Try restarting ORANGE HSD
4. Try restarting Matlab

VI. Useful hints

1. When viewing a spacial image, the average can be viewed by typing a “0” into the box. (8 in figure 32 and 8 in figure 27)

VII. Possible Future Work

1. Continue to fix bugs that are found
2. Add spectral stochastic estimation to the LSE tool
3. Add a data fusion tool to the code
4. Add an OID tool to the code

VIII. Code Listing

Folder ORANGE HSD 3.0: This is the Main directory for ORANGE HSD

High_Speed_Data.m: This is the code that calls the ORANGE HSD main screen

Subfunctions: This directory contains all of the functions that are used in the program

Calculator: This directory contains the files used to run the custom function tool

Calculator.m: This creates and controls the custom function GUI

Custom_function.m: This will apply a function to the selected data set

Flip_data.m: This will flip a data data set. It will place the data located at the bottom on top and top on the bottom. It also negates the v velocity.

q_criterion.m: This is a user defined function that calculates q criterion on 2D data sets

Rotate_data.m: This rotates the data some desired degree. It uses interpolation to accomplish this.

Swirl_L2.m: This a user defined function that calculates the swirl on 2D data sets

Conditioning: This directory contains the codes that run the signal-conditioning tool

Conditioning.m: This will perform the signal conditioning on the selected data set

ORANGE_Conditioning.m: This controls and runs the signal conditioning GUI.

Preview_Conditioning.m: This shows what the conditioned signal looks like.

Conversion: This directory contains the files used to convert data into a format usable by ORANGE HSD

Create_Velocity_file.m: This function will write the .mat file when converting OVERFLOW data

Extract_Velocity_Temporal.m: This function will extract a temporal signal at user-selected locations

Format_Data.m: This function controls the format data GUI

Format_Fluent.m This controls and runs the gui for converting Fluent data

format_GE_data.m: This will read an ascii grid file (i x y z) and an ascii data file (i rho P U V W)

format_lumin_pod2.m: This function converts the .tif files into the .mat files needed to run ORANGE HSD

Format_OVERFLOW.m: This function controls the GUI for formatting the OVERFLOW data(Not used anymore)

format_piv_pod_dantec.m: This function formats Dantec PIV data stored in a text file

format_piv_pod_lavision.m: This function converts the Lavision .vc7 files into .mat files

format_piv_pod_vort.m: This function calculates the vorticity for the Lavision files and writes the vorticity as a .mat file

Format_Pressure_Data.m: This function converts pressure data stored as .txt files into .mat files

OVERFLOW_Converter.m: This will read overflow files and convert them to .mat files

OVERFLOW_Converter_Pressure.m: This will read overflow files, calculate the pressure and convert it to .mat files

Read_Fluent_Data.m: This will read data from fluent that is in block-formatted tecplot

Read_Fluent_Data_tecplot_point.m: This will read data from fluent in point formatted tecplot

Read_OVERFLOW_Data.m This function reads the OVERFLOW grid and solution files(not used anymore)

Data_Extraction: This folder contains the files to perform data extraction

Data_Extract.m: This performs the 2D-2D extraction

Data_Extract_View.m: This previews the 2D-2D extraction

Data_Extract_View1D.m: This previews the 1D-1D extraction

Data_Extract1D.m: This performs the 1D-1D extraction

Extract3D_2D.m: This performs the 3D-2D extraction

ORANGE_1D_Extraction.m: This runs and controls the GUI to perform the 1D-1D extraction

ORANGE_Extraction.m: This controls and runs the GUI to run perform the 2D-1D extraction

ORANGE_Extraction_1D.m: This controls the 1D-1D extraction GUI (not used anymore)

ORANGE_Extraction_3D_2D.m: This controls and runs the 3D-2D extraction GUI

Data_View: This directory contains the files used to view the data

Field_Data_Viewer.m: This controls and runs the GUI used to view contours plots

Plot_PDF.m: This separates the data into bins and plots a bar graph of the bins (PDF plot)

Plot_signal.m: This plots the 1D data

Plot_Spatial.m: This plots the 2D data

Time_Trace_Data_Veiwer.m: This controls and runs the gui to plot line plots

Orange_Correlations: This directory contains the files to run ORANGE Correlations

cor_Data.m: This function calculates the correlation function

FFT_Data.m: This function calculates the Fourier transform of the data

Orange_Correlate.m: This function controls the ORANGE Correlate GUI

plot_sfc.m: This function plots the correlation, fft, or spectrum for the data

spectrum_Data.m: This function calculates the spectrum of the data

Orange_LSE_QSE: This directory contains the files to run ORANGE LSE

estimate_field.m: This function uses the POD eigenfunctions and estimate POD coefficients to estimate the velocity field

LSE.m: This function performs the LSE analysis and outputs the estimate POD coefficients and mapping LSE mapping matrix

Orange_LSE.m This function controls the ORANGE LSE GUI

plot_aPOD_Estimate.m: This plots the estimated POD coefficients on the same axis as the actual POD coefficients

QSE.m: This function performs the QSE analysis and outputs the estimate POD coefficients and mapping QSE mapping matrix

recon_pic_LSE: This functions plots a reconstructed velocity field using the original POD coefficients and the estimated coefficients and compares it with the original velocity field

Orange_POD: This directory contains the files to run ORANGE POD

aPOD_plot.m: This function will plot the time dependent POD coefficients

AVE_Fluc_Kernel.m: This function calculates the average and POD kernel

Classic_POD.m: This runs the classic version of POD

Fused_POD.m: This runs the fused POD process

Gappy_POD.m: This runs Gappy POD

kernel.exe: This is the executable that calculates the kernel in Fortran

kernel.F90: This is the source code for the kernel calculation in Fortran

load_data.m This function reads .mat files for the parallel version of Matlab

Modal_correlation.m: This compares two different sets of modes and produces a correlation plot

ORANGE_POD_V1_1_Beta.m: This function controls all of the inputs

ORANGE_POD_V2_2.m: This function controls the POD GUI

ORANGE_Split_POD.m: This runs split POD

parallel_AVE_Fluc_Kernel.m: This function calculates the average and POD kernel for parallel Matlab

Parallel_Snapshot_POD.m: This function calculates the POD modes and coefficients for parallel Matlab

Plot_POD_modes.m: This function creates plots of the POD modes (not used anymore)

recon_data.m: This function creates a reconstruction of a data set, which can be used in later analyses

recon_pic.m: This function creates a reconstruction of an individual snapshot

recon_plif_pic.m: This function creates a .tif image of the reconstructed PLIF data

recon_vid_contour.m: This function creates a reconstructed video of the data

reconpicmov.m: This function creates a reconstructed video of the PLIF data

Snapshot_POD.m: This function calculates the POD modes and coefficients

Write_output_file.m: This function writes an output file that tells the time it took to compute the POD modes

Orange_Wavelet: This directory contains the files to run ORANGE Wavelet

Mexhat_wavelet.m This function computes the Mexican hat wavelet coefficients

Morlet_wavelet.m This function computes the Morlet wavelet coefficients

Orange_Wavelet.m This function controls the ORANGE wavelet GUI

plot_wavelet.m This function plots the wavelet map

Statistics: This directory contains the files to run the statistics tool

ORANGE_Stats.m: This is the file that controls and runs the statistics GUI

Statistics.m: This file calculates the statistics

Data_management.m: This registers the data when it is first imported

Data_management_screen.m: This controls and runs the data management GUI

Masking.m: This is used to create a mask for a data set

Organize_data.m: This sets up new projects and loads in the new data

progressbar.m: This makes a progress bar when the code is running (Created by Steve Hoelzer)

Splash.m: This creates a splash screen at the start up of the program

References

- ¹Lumley, J. L., "The Structures of Inhomogeneous Turbulent Flows," *Atmospheric Turbulence and Radio Wave Propagation*, 1967.
- ²Kirby, M. and Sirovich, L., "Low Dimensional Procedure for the characterization of the human faces," *Journal of Optical Society of America*, Vol. 102, 1987.
- ³Pinier, J. T., Ausseur, J. M., Glauser, M. N., and Higuchi, H., "Proportion Closed-Loop Feedback Control of Flow Separation," *AIAA*, Vol. 45, 2007.
- ⁴Camphouse, R. C., Myatt, J. H., Schmit, R. F., Glauser, M. N., Auesseur, J. M., Andinio, M. Y., and Wallace, R. D., "A Snapshot Decomposition Method for Reduced Order Modeling and Boundary Feedback Control," 2008, AIAA-2008-4195.
- ⁵Aradag, S., Siegel, S., Seidel, J., Cohen, K., and McLaughlin, T., "Filtered POD-baessed low-dimensional modeling of the 3D turbulent flow behind a circulare cylinder," *Internation Journal for Numerical Methods in Fluids*, Vol. 66, 2011.
- ⁶Sirovich, L., "Turbulence and the Dynamic Coherent structures Part 1: Coherent Structures," *Quartely Applied Mathematics*, Vol. 45, 1987.
- ⁷Everson, R. and Sirovich, L., "Karhunen–Loève procedure for gappy data," *Journal of Optical Society of America*, Vol. 12, No. 8, 1995, pp. 1657–1664.
- ⁸Bui-Thanh, R., Damodaran, M., and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1505–1516.
- ⁹Ruscher, C. J., Dannenhoffer, J. F., and Gluaser, M. N., "Application of Snapshot POD on a Varying Grid," *51st AIAA ASM*, Grapevine, Texas, 7-10 January 2013, AIAA-2013-0256.
- ¹⁰Adrian, R. J., "On the role of Conditional Averages in Turbulence Theory," *Turbulence in Liquids: Preceedings of the Fourth Biennial Symposium on Turbulence in Liquids*, Vol. 66, 1977, pp. 323–332.
- ¹¹Tropea, C., Yarin, A., and Foss, J., *Springer Handbook of Experimental Fluid Mechanics*, Springer-Verlag Berlin Heidelberg, 2007.
- ¹²Bonnet, J. P., Cole, D., Delville, J., Glauser, M. N., and Ukeiley, L., "Stochastic Estimation and Proper Orthogonal Decomposition: Complementary Techniques for Identifying Structure," *Experiments in Fluids*, Vol. 17, 1994, pp. 307–314.
- ¹³Taylor, J. A. and Glauser, M. N., "Towards Practical Flow Sensing and Control via POD and LSE Based Low-Dimensional Tools," *Journal of Fluids Engineering*, Vol. 126, No. 3, 2004, pp. 337–345.
- ¹⁴Tung, T. and Adrian, R. J., "Higher Order Estimates of Conditional Eddies in Isotropic Turbulence," *Physics of Fluids*, Vol. 23, 1980, pp. 1469–1470.
- ¹⁵Gutmark, E. J., Veraille, S., Bonnet, J.-P., and Grinstein, F., "Linear Stochastic Estimation of a Swirling Jet," *AIAA Journal*, Vol. 44, No. 3, 2006, pp. 457 – 468.
- ¹⁶Tinney, C., Jordan, P., Hall, A. M., Deville, J., and Glauser, M. N., "A Time-Resolved Estimate of the Turbulence and Sound Source Mechanisms in a Subsonic jet flow," *Journal of Turbulence*, Vol. 8, No. 7, 2007.
- ¹⁷Farge, M., "Wavelet Transforms and their applications to turbulence," *Annual Review of Fluid Mechanics*, Vol. 24, 1992, pp. 395–457.
- ¹⁸Lewalle, J., Delville, J., and Bonnet, J.-P., "Decomposition of Mixing Layer Turbulence into Coherent Structures and Background Fluctuations," .
- ¹⁹Schobeiri, M. T., Read, K., and Lewalle, J., "Effect of Unsteady Wake Passing Frequency on Boundary Layer Transition, Experimental Investigation, and Wavelet Analysis," *Journal of Fluids Engineering*, Vol. 125, 2003, pp. 251–266.
- ²⁰Farge, M., Schneider, K., Pellegrino, G., Wray, A. A., and Rogallo, R. S., "Coherent vortex extraction in three-dimensional homogeneous turbulence: Comparison between CVS-wavelet and POD-Fourier Decompositions," *Physics of Fluids*, Vol. 15, No. 10, 2003, pp. 2886–2896.

Appendix C

Pre-submission of High Speed Optical Measurements in a Jet Flow:

High Speed Optical Measurements in a Jet Flow

Christopher J. Ruscher · Zachary P. Berger · Patrick R. Shea · Andre M. Hall ·
Charles E. Tinney · Matthew G. Berry · Mark N. Glauser · Sivaram Gogineni ·
Barry V. Kiel

Received: date / Accepted: date

Abstract In the current investigations Laser Doppler Velocimetry (LDV) and Time-Resolved Particle Image Velocimetry (TRPIV) are implemented to acquire the velocity in a Mach 0.6 compressible jet. Single and multi-point (in time) statistics, specifically power-spectral densities, for the LDV and PIV are within the calculated uncertainty of one another. Wave number and frequency information is obtained from the TRPIV due to its accurate temporal characteristics and sufficient spatial resolution. Previous studies have used wave number-frequency information or estimation techniques to determine noise sources. This work focuses on similar studies using experimental data.

C.J. Ruscher (✉) · Z. P. Berger · P.R. Shea · M.G. Berry · M.N. Glauser
Syracuse University, Link Hall, Syracuse NY, 13210, USA
E-mail: cjrusche@syr.edu

Z.P. Berger
E-mail: zpberger@syr.edu

P.R. Shea
E-mail: prshea@syr.edu

M.G. Berry
E-mail: mgberry@syr.edu

M.N. Glauser
E-mail: mglauser@syr.edu

A.M. Hall
Pratt & Whitney, 400 Main Street, East Hartford, CT 06118
E-mail:

C.E. Tinney
University of Texas at Austin, Austin, TX 78712
E-mail: cetinney@utexas.edu

S. Gogineni
Spectral Energies, LLC, 5100 Springfield St. Suite 301, Dayton, OH 45431
E-mail: sgogineni@spectralenergies.com

B.V. Kiel
AFRL, WPAFB, OH, 45433, USA
E-mail: Barry.Kiel@wpafb.af.mil

Keywords Laser Doppler Velocimetry · Particle Image Velocimetry · High-Speed Jet Flow

Nomenclature

D	=	Jet diameter
r	=	Radial jet direction
R	=	Jet radius
Re	=	Reynolds number
S_{uu}	=	Streamwise autospectral density
u	=	Instantaneous streamwise velocity
U	=	Mean streamwise velocity
U_{∞}	=	Velocity at the exit of the jet
u'	=	Root-mean-square streamwise velocity
z	=	Streamwise jet direction
σ_x	=	Standard deviation

1 Introduction

Laser Doppler Velocimetry (LDV) and Time-Resolved Particle Image Velocimetry (TRPIV) are two techniques currently used to measure fluid velocities in a flow field. The current investigation focuses on comparing 10 kHz TRPIV with burst-mode LDV for a high Reynolds number, compressible jet flow in the spectral domain. Sufficient times and spatial resolution obtained with TRPIV yields wave number-frequency information. This information has also been obtained via stochastic estimation by Tinney *et al.* (2008a) and large eddy simulation (LES) by Freund (2001) and Fievet *et al.* (2013). This work exploits the temporal and spatial resolution inherent to TRPIV measurements, not obtainable with other measurement techniques.

The early forms of studying fluid flow included qualitative measurement techniques such as flow visualization

using smoke, dye or other particles. When quantitative measurements became of interest, pitot tubes were used to probe the flow field in order to obtain velocity. For more complex flows in which high accuracy was needed, hot-wire anemometry became the standard for single point velocity measurements in a flow field (Comte-Bellot (1976), Ewing *et al.* (1995), Bruun and Principles *et al.* (1995), Wyngaard (1968), Klewicki and Falco (1990), Zhu and Antonia (1996)). More advanced modifications were later developed to include cross-wires (Burattini and Antonia (2005), Shabbir *et al.* (1996), Browne *et al.* (1988), Johnson and Eckelmann (1984), Tagawa *et al.* (1992), Tutu and Chevray (1975)). and flying hot-wires (Cantwell (1975), Coles and Wadcock (1979), Coles and Glauser (1998), Hussein and George (1989), Thompson and Whitelaw (1984), Watmuff *et al.* (1983)). Hot-wire probes provide single point velocity measurements with high sampling rates, however these measurements are intrusive and can be disruptive to the flow physics if not positioned carefully. Moreover, hot-wires are more desirable for incompressible flows since they measure momentum. In the cases where the density is constant, velocity can easily be obtained, however hot-wire measurements are not ideal for compressible flows.

Two optically based techniques (LDV and PIV) were developed for non-intrusive velocity measurements. These two techniques use different principles to obtain the velocity, however both methods have been extensively used in the field. Both methods involve seeding the flow and using laser-based diagnostics to track the particles. LDV relies on the Doppler shift, tracking the particles moving through a pre-defined volume of space. The burst mode LDV captures velocity at a single point in space with a high sampling rate in time. Once again the main advantage of LDV is its non-intrusive nature as well as its capability of high sampling rates. However, it is also important to keep in mind that LDV is not sampled at uniform time intervals. This technique has been used to measure turbulence as seen in the work by Buchhave *et al.* (1979) and George & Lumley (1973), and was used to measure the velocity in a free jet flow by Lau *et al.* (1979). Adrian & Yao (1986) also investigated velocity measurements by means of LDV, using spectral approaches. In addition, LDV has been used in turbulent pipes and channels (Durst *et al.* (1995), Romano *et al.* (1999)).

For standard PIV, which relies on the correlation of flow snapshots, typical sampling rates are of $\mathcal{O}(\text{Hz})$ between image pairs. Therefore, for high-speed flows, the PIV does not yield time-resolved velocity fields. Moreover, if only a single camera is used, only two components of velocity can be measured. With the use of a second camera, stereoscopic three component velocity fields can be obtained. The issue of spatial resolution often comes up when referring to PIV since this is usually dependent on the camera specifications, particle size and seeding density. In terms of resolving rele-

vant scales, Lavoie *et al.* (2007) compared mean turbulence quantities obtained from PIV and hot-wire measurements for grid turbulence. It was concluded that hot-wire measurements are more resolved than PIV in terms of small scale turbulence structures. Each measurement technique has its advantages and disadvantages depending on the quantity of interest and the experimental setup.

Many single point comparisons have been made between LDV and PIV such as work by Grosjean *et al.* (1997), Kerhervé *et al.* (2010), Cenedse *et al.* (1994) and Leder & Brede (2004). Grosjean *et al.* used LDV and PIV to study an unsteady swirling flow, while Kerhervé *et al.* examined a single stream and co-axial coplanar jet, using the two techniques. LDV and PIV have been taken simultaneously by Cenedse *et al.* for a circular and annular jet, where it was found that the mean velocity profiles showed good agreement in the statistical sense. PIV and phase-averaged LDV were compared for the wake behind a circular cylinder ($Re = 5000$) by Leder and Brede (2004). This work showed qualitative agreement in the large scale structures shed from the cylinder from the phase-averaged LDV and PIV measurements. More in depth comparisons in terms of spectra and time series analyses were not presented in the above cases because the sampling rates of the PIV system were not sufficient enough to obtain the high frequency information.

Direct comparison between simultaneously sampled TR-PIV and LDV measurements were made in the wake of an axial fan by Mattern *et al.* (2012). Good agreement was shown in the velocity distributions along the centerline of the measurement plane, and the results in the current paper are compared with that of Mattern *et al.* In recent years, TR-PIV has been implemented for high-speed flow applications such as jet flows (Low *et al.* (2013), Murray *et al.* (2012), Wernet (2007)). In the work of Low *et al.*, a 10kHz time-resolved system was used to make direct comparisons between the low-dimensional velocity modes in the near-field and the far-field acoustics. Murray *et al.* implemented MHz PIV in a high speed jet, however only captured on the order of 10 snapshots per data set. This is compared to the current investigation where sufficiently long time records are collected to compare single and multi-point (in time) statistics, specifically spectra, between the TRPIV and LDV. Wernet collected both TRPIV (25kHz down-sampled to 10kHz) and hot-wire measurements in the shear layer of a Mach 0.5 jet. These measurements show good agreement in the power-spectral densities for the two techniques at multiple downstream locations. Therefore, in Wernet's experiments, the TRPIV measurements are validated by the hot-wire measurements by comparing the spectral content in the shear layer of the jet. Comparisons are made to Wernet's work in the current paper, using LDV instead of hot-wire to compare to TRPIV, for a high Reynolds number, compressible jet flow. The results to be presented discuss the single point

statistics as well as spectral content for the two optical techniques. Associated uncertainties as well as the advantages and drawbacks of each technique is also addressed.

2 Experimental Setup

The two experiments conducted for this study were independently carried out in the Syracuse University large scale anechoic chamber and high speed jet facility. The anechoic chamber is constructed from 12 inch thick reinforced, single poured concrete, having a total volume of 367m^3 . The interior surfaces of the chamber are acoustically treated with fiberglass wedges having a cut-off frequency of 150 Hz. The inner chamber volume from the tips of the wedges is 206m^3 . The facility was updated in order to perform the experiments discussed in this paper, as described by Tinney *et al.* (2004). For the studies of high Reynolds number compressible jets, an axisymmetric nozzle with an exit diameter of 0.0508 m is investigated. The convergent nozzle uses a matched 5th order polynomial for its interior profile contraction. Each of the studies discussed in the following sections were conducted at a Mach number of 0.6, corresponding to a Reynolds number of 690,000 based on the exit diameter of the jet. The reference axes with respect to the jet nozzle exit are shown in figure 1.

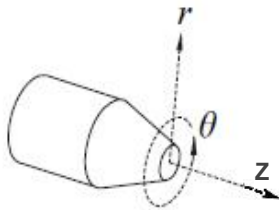


Fig. 1: Reference axes with respect to the jet exit

2.1 LDV Setup

Hall *et al.* (2005) explored velocity-pressure correlations of the axisymmetric jet using near-field Kulite pressure transducers as well as an LDV system. An array of fifteen Kulites were positioned azimuthally just outside the jet's shear layer and were located at $0.875D$ downstream of the nozzle exit, and $0.875D$ from the centerline. The velocity was measured with a Dantec Dynamics LDV system, powered by a 3 Watt argon-ion laser head, using a burst spectrum analyzer (BSA) for processing. Single component measurements were acquired in the streamwise direction with respect to the jet exit. The LDV setup is seen in figure 2.

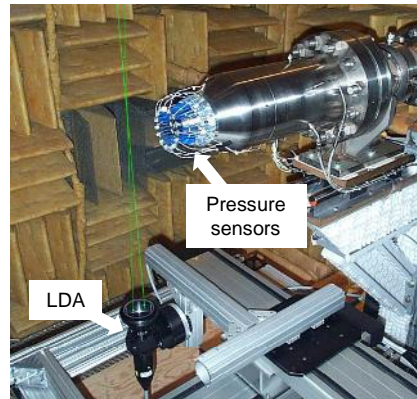


Fig. 2: Laser orientation with respect to the jet

Single point measurements were taken at various radial and streamwise locations using a traverse system fixed at zero degrees in the azimuthal direction with respect to the horizontal. Measurements were made along the centerline of the potential core, at the the potential core-shear layer interface, along the centerline of the shear layer and at the outer edge of the shear layer. For more information on the specific locations of the LDV measurements see Hall *et al.* (2005). Approximately 500,000 samples were collected at an average rate of 30kHz to ensure time-resolved measurements.

2.2 TRPIV Setup

The TRPIV system provided by Spectral Energies, LLC. employs a Quantronix Hawk-Duo series high energy diode-pumped Neodym-Yag (Nd:YAG) Laser, ($\lambda = 532\text{nm}$), and a Photron FASTCAM CCD camera. The laser outputs 4 mJ per pulse at 10kHz. For more details regarding the seeding of the jet and co-flow, refer to Low *et al.* (2013).

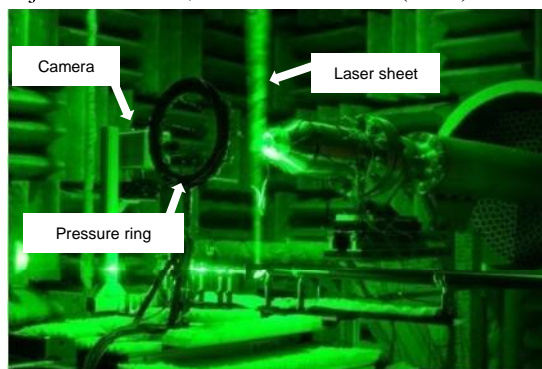


Fig. 3: TRPIV setup with pressure arrays

The PIV windows consisted of two-component measurements in the streamwise plane of the jet. The interrogation region was $1.5D \times 1.5D$ and several measurement planes were taken using a traverse system. Several Mach numbers were

collected; however Mach 0.6 is the focus of this study to compare with the LDV. LaVision's DaVis software was used for data collection and post-processing of the PIV images. A total of 8,623 snapshots were acquired, making the total record length 0.8623 seconds. The TRPIV measurements used in the analysis presented are part of a more comprehensive set of experiments (Low *et al.* (2013)), in which low-dimensional velocity modes are correlated with near and far-field pressure to identify "loud" modes in the flow field. The complete TRPIV configuration can be seen in figure 3.

TRPIV has the advantage over LDV in that multiple points can be sampled simultaneously to form a full vector field. With LDV, the advantage over TRPIV is that larger ensembles of single point measurements can be sampled. While each of the two techniques has their advantages, comparisons can be made for both single point and multi-point (in time) statistics.

2.3 Experimental Uncertainties

The measurement uncertainties in the LDV system are stated to be 1% of the freestream velocity (Dantec (1995; 2006)). The measurement uncertainty in the PIV system is approximately 7.5% based on the method outlined by Raffel *et al.* (1998). Other uncertainties include the location of the measurements, the ambient conditions (the test were performed at different times), and post processing differences. The uncertainty in location is set to 0.02D, which is the distance between two grid points in the PIV measurement.

In addition to the measurement uncertainties, the statistical uncertainty in the mean is given as the standard deviation (σ_x) divided by \sqrt{N} , the number of statistically independent samples. The statistical uncertainty in the mean and RMS is computed using approximately 1,000 and 25,000 statistically independent snapshots for the PIV and LDV, respectively. This assumes two integral scales separating the measurements, and it was found that the overall uncertainty is dominated by the statistical uncertainty in this case.

3 PIV Signal Extraction

In order to compare the LDV data and the PIV data, pre-processing must be applied, since PIV produces a vector map while LDV takes a measurement at one spatial location. Data were extracted from the PIV snapshots that corresponds to the location of the LDV measurements. However, upon comparing the mean velocity profiles, it was discovered that the PIV and the LDV did not line up correctly in the radial direction. The PIV image was shifted until the mean velocity at the center of the shear layer measured by the PIV and the LDV matched within 0.02D (spacial resolution of

the PIV grid) to account for the mismatch in the mean profiles. The locations of the LDV and the PIV do not match exactly since the PIV data are defined at discrete points and are not continuous. An interpolation could have been performed to match the data at the center of the shear layer, but this would add error, so it was decided to evaluate the velocity at known data points.

The PIV data are extracted at the point closest to where the LDV measurement was taken. The locations of the extracted points are shown as black dots on the PIV contours of streamwise Mach number (figure 4).

The white space in figure 4 is the result of a pressure sensor in between the laser plane and the camera. The sensor was not in the flow and the only consequence of the sensor is the missing data in the sensor region. For that reason, the data are extracted from the bottom half of the PIV plane.

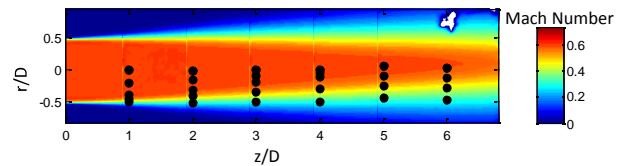


Fig. 4: Contours of mean Mach number from PIV measurements and corresponding locations of LDV measurements

The LDV system records velocity when a particle travels through the integration volume creating a non-uniform time interval, which is accounted for by linearly interpolating the signal onto a uniform time grid. The interpolation allows the computation of the autospectra using an FFT. A sample signal and the corresponding interpolated signal are shown in figure 6. One can see the two signals are similar and the interpolated signal is passing through many of the points in the original signal.

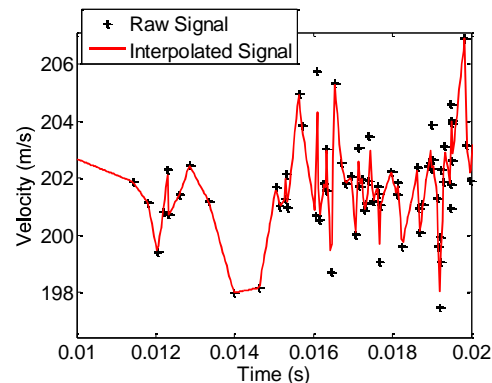


Fig. 5: Comparison of the raw time series and the interpolated data

In this paper, the results for $z/D=3.0, 4.0, 5.0,$ and 6.0 are discussed at the centerline of the jet ($r/D=0.0$) and center of the shear layer ($r/D=0.5$) for Mach 0.6. The other locations show similar results, but for the sake of brevity only a few locations are shown. Also, the shear layer is very thin at upstream locations, almost the size of the LDV inspection region and the PIV grid spacing. The shear layer at that streamwise location is not spatially resolved and trying to align the PIV and LDV measurements was not possible. These results are not shown because the differences associated between the PIV and LDV measurements could be affected by these issues.

The LDV data had many statistical outliers which are non-physical. These points are fixed using the filtering technique suggested by Tinney (2005), which uses an iterative process that locates velocities that are larger than three standard deviations. Those points are replaced by an average of the data point immediately before and after the outlier.

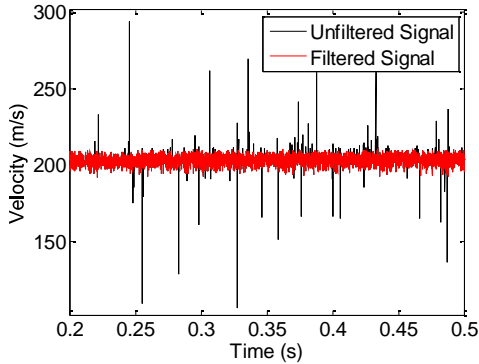


Fig. 6: Comparison of the unfiltered time series and the filtered data

4 Statistical Comparison of PIV and LDV Results

Before the temporal comparisons are made, it is important to show that the PIV and LDV measurements match in a statistical sense. The mean profiles, with the exception of the data point at the outer edge of the shear layer, are within the uncertainty as shown by the error bars in figure 7.

The RMS profiles also fall within the uncertainty as seen by the error bars in figure 8. Both measurements techniques show peak RMS values at the center of the shear layer ($r/D=0.5$) as expected. In figures 7 and 8 the abscissa is normalized using

$$\eta(z) = \frac{r - r_{0.5}}{z} \quad (1)$$

where $r_{0.5}$ is the radial location where the velocity is 50% of the freestream velocity. This normalization take advantage of the linear expansion of the jet, allowing the velocity

profiles at different axial locations to collapse on one another as seen in Tinney *et al.* (2008b) and Ukeiley *et al.* (2007). One should also note that the PIV has smaller fluctuations in the potential core than the LDV. This can be attributed to the fact that the PIV inherently performs spatial averages on the velocity causing lower levels of fluctuations. Also, the LDV had statistical outliers in the data, which could cause larger fluctuations if not completely filtered. The PIV measurements in the shear layer predict higher RMS values than the LDV, consistent with that of Wernet.

$\frac{z}{D}$	$(\frac{U}{U_\infty})_{PIV}$	$(\frac{U}{U_\infty})_{LDV}$	$(\frac{u'}{U_\infty})_{PIV}$	$(\frac{u'}{U_\infty})_{LDV}$
3	1.00	1.00	0.02	0.03
4	1.00	1.00	0.03	0.04
5	0.99	0.99	0.04	0.06
6	0.99	0.95	0.07	0.08

Table 1: Statistical comparison between LDV and PIV measurements along the centerline

$\frac{z}{D}$	$(\frac{U}{U_\infty})_{PIV}$	$(\frac{U}{U_\infty})_{LDV}$	$(\frac{u'}{U_\infty})_{PIV}$	$(\frac{u'}{U_\infty})_{LDV}$
3	0.58	0.58	0.17	0.15
4	0.56	0.57	0.17	0.15
5	0.55	0.55	0.18	0.15
6	0.53	0.54	0.17	0.15

Table 2: Statistical comparison between LDV and PIV measurements along the center of the shear layer

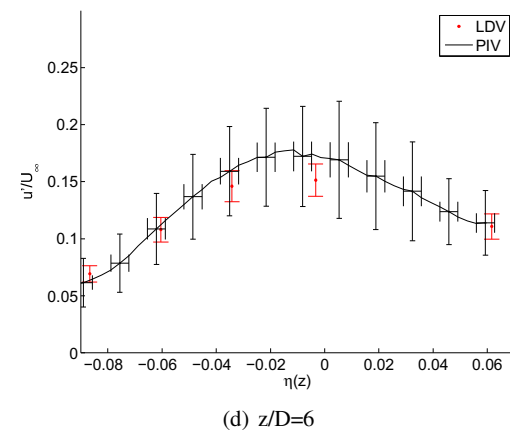
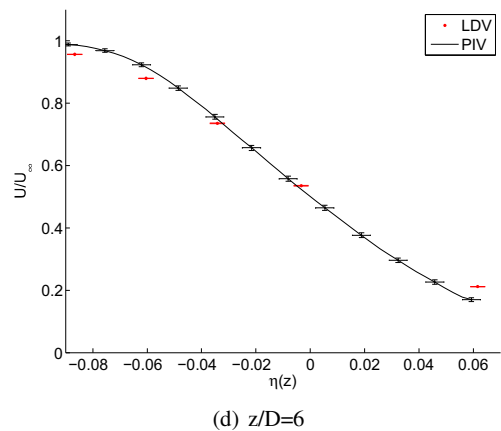
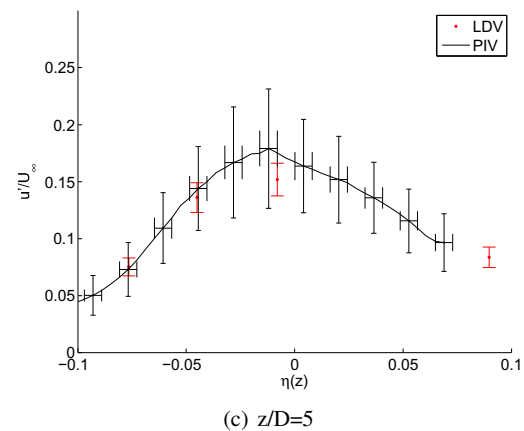
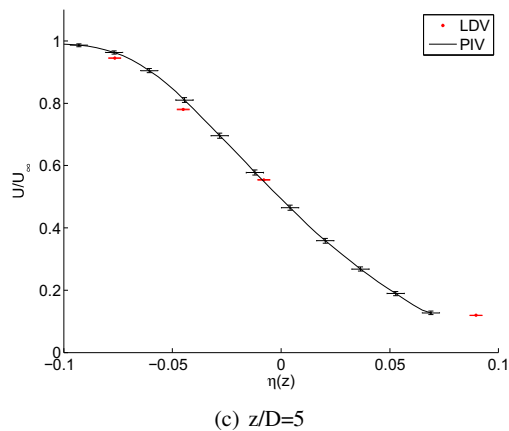
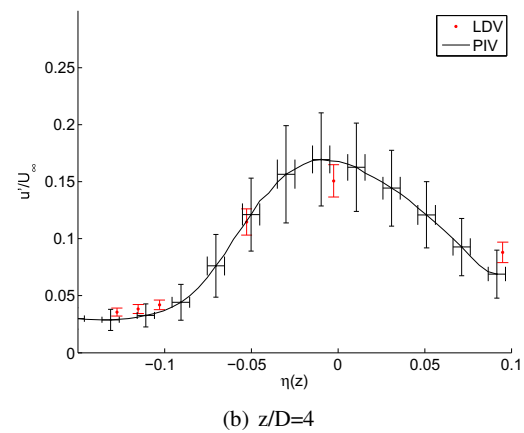
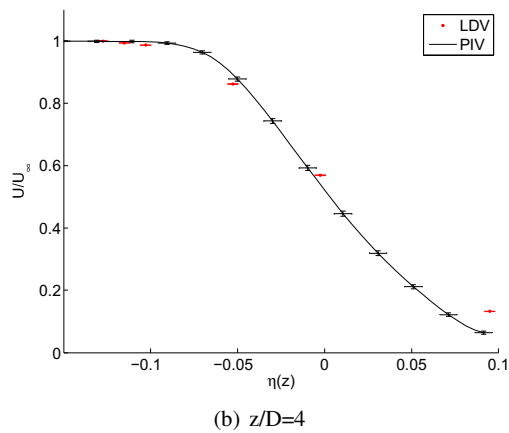
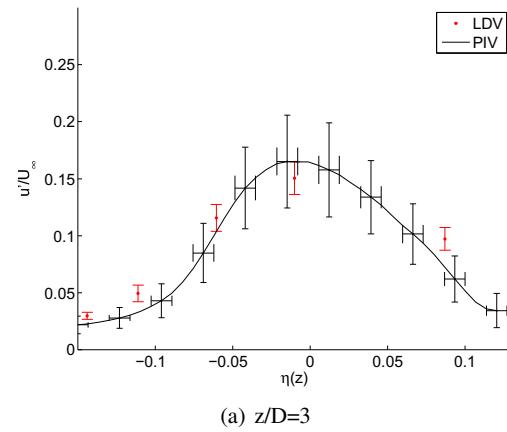
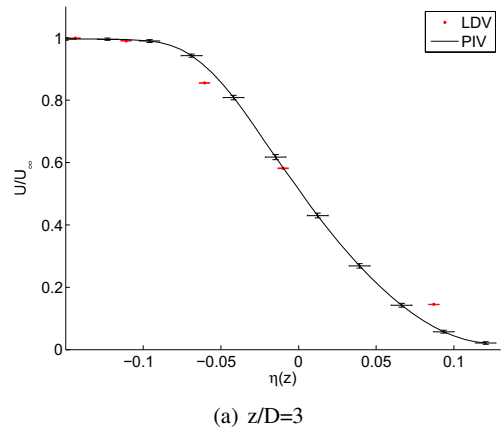


Fig. 7: Comparison of mean Mach number profiles

Fig. 8: Comparison of RMS Mach number profiles

5 Temporal Comparison of PIV and LDV Results

Knowing that the single point statistics of the LDV and PIV measurements match within the uncertainty, the spectral content of the measurements are compared along the centerline (figure 10) and the center of the shear layer (figure 11). A direct comparison of the measurements can not be made in the time domain because the measurements were not taken simultaneously as opposed to the simultaneous sampling performed by Mattern *et al.* (2012). The noisier spectra of the PIV is an artifact of two orders of magnitude fewer samples than the LDV. This was addressed using a 5-point moving average filter on the PIV data. The PIV and LDV spectra were block-averaged using a block size of 512 data points, giving 15 and $\mathcal{O}(1000)$ statistically independent blocks for the PIV and LDV, respectively.

Looking at the shape of the autospectra for both measurements, similar trends are seen, such as the increase in spectral content further downstream. This implies the jet is fluctuating more downstream as is expected. The shape also shows the jet is becoming broadband further downstream, showing the potential core is collapsing and the flow resembles homogenous turbulence. Both measurements also show the spectral content is high in energy and more broadband in the shear layer as compared to the potential core.

In the low frequency band (less than 1kHz) the PIV shows less spectral energy than the LDV measurements due to the inherent spatial averaging of the PIV system. Despite these differences, the peak frequencies measured by the LDV and PIV are within 10% of one another (see Table 3).

$\frac{z}{D}$	$f_{peakPIV}$ (Hz)	$f_{peakLDV}$ (Hz)
3	1850	1700
4	1750	1600
5	1650	1500
6	1600	1450

Table 3: Comparison of the dominant frequency

The spectra for the shear layer was compared with Wernet's work and similar trends are seen. The PIV is showing more energy than the LDV, which is consistent with the RMS plots in figure 8 and Wernet's work which shows the PIV has more energy than the hot-wire. According to Hearst *et al.* (2014), the spectra for LDV and hot-wire measurements match for three decades in the grid turbulence problem, giving a sufficient level of confidence for the results presented here.

6 Wave Number-Frequency

Based §5 it has been shown that the spatial and temporal resolution of the TRPIV yields the same results as the LDV within the measurement uncertainties. As a result one can

perform analyses that extract features coupled in time and space, such as wave number-frequency analysis. This aids in the identification of noise sources as shown by Tinney *et al.* (2008b), Freund (2001), and Fievet *et al.* (2013). These studies obtained using simulation and estimation techniques, show that most of the acoustic sources do not propagate to the far-field. The velocity field can be transformed into wave number-frequency space using the following

$$S(k_x, f) = \frac{1}{2\pi} \int \int u(x, t) e^{-i(k_x x + 2\pi f t)} dx dt \quad (2)$$

where $u(x, t)$ is the streamwise velocity, f is the frequency, and k is the wave number.

The transform is applied to the streamwise velocity, which according to Fievet *et al.* (2013) shows similar results to that of the full acoustic source terms. The results of the transformation are shown for data extracted at a line along the center of the shear layer for the PIV windows at $z/D = 2.0 - 3.5$, $z/D = 3.0 - 4.5$, $z/D = 4.0 - 5.5$, and $z/D = 5.0 - 6.5$ in figure 9. One can see that the strongest amplitudes fall along the line following the convection speed (U_c), which is approximately 60 % of the jet velocity U_j . Similar results are shown by previous simulation and estimation techniques. The number of spatial grid points in each TRPIV window resolves the wave number to roughly $0.5D$. This results in a step-like appearance in the wave number as seen in figure 9. One can also note that there are very few sources in the region of $|f/k| > a_\infty$, which according to Freund, are the sources that propagate to the far-field.

7 Summary and Conclusions

In this paper a jet flow field is investigated using two different techniques. Velocity measurements are obtained for a Mach 0.6 fully compressible, turbulent jet with a Reynolds number of 690,000. Both measurement techniques are optically based, acquiring the velocity in a non-intrusive manner. LDV measurements, relying on the Doppler shift, were acquired to assess the spectral content of the jet at various streamwise and radial locations. TRPIV measurements, which use correlations between a set of image pairs, were collected to investigate the time-evolution of the flow field. The TRPIV sampled at 10kHz allows for the velocity spectra to be evaluated, resolving a wide range of time scales. As previously stated, LDV and PIV measurements have been compared before in the statistical sense, however this is the first time high-speed TRPIV is compared with LDV using spectral approaches for a high-speed, compressible jet. The high sampling rates of each method allow for spectral comparisons to be made in a complex high-speed flow field. Exploiting the accurate temporal and spatial resolution of the TRPIV, wave number-frequency information is extracted to identify acoustic sources that propagate to the far-field.

The PIV and the LDV measurements show similar results to that of Wernet's hot-wire and TRPIV experiments. The autospectra for the PIV and the LDV exhibited similar trends and produce dominant frequencies within 10% of one another. The differences in the spectral content between the two measurements techniques is within the calculated uncertainty.

The work presented shows PIV and LDV measurements yield results within the uncertainty of one another statistically and spectrally for a high-speed jet flow. For the first time, these two techniques are compared to validate the reliability of the measurements in the spectral domain, for a high-speed jet flow. While each technique is optically based, the methods by which the velocity is calculated are completely independent. Therefore the results presented throughout this paper serve as validation for the individual approaches.

The wave number-frequency analysis produced from the TRPIV data show similar trends to that of past simulation and estimation work. The majority of the sources propagate at the convective speed and only a few sources propagate at supersonic speeds resulting in far-field noise.

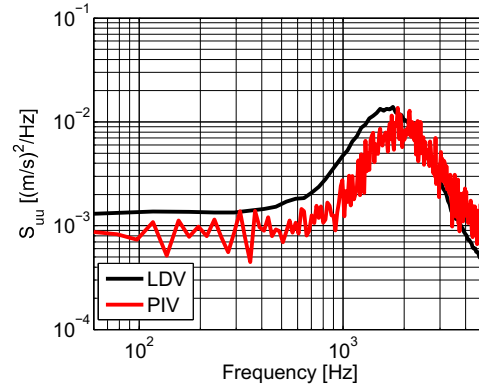
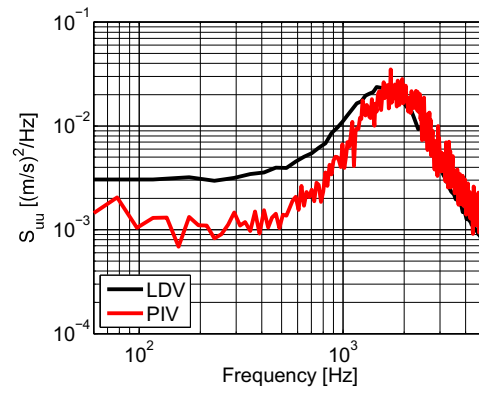
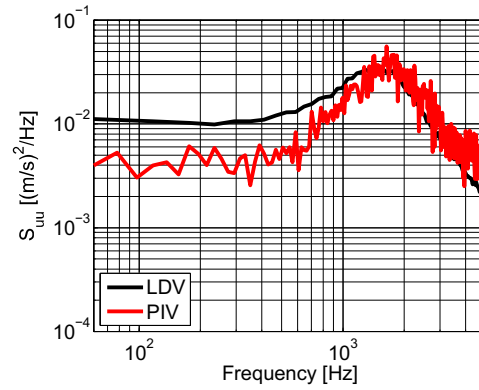
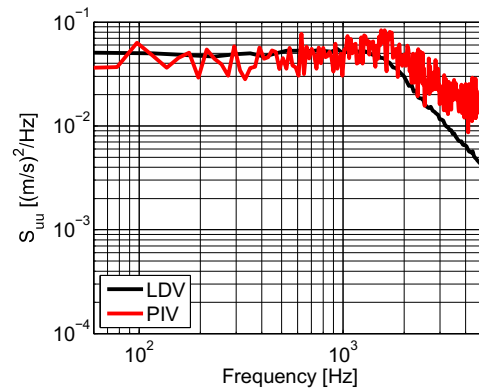
(a) $z/D=3$ (b) $z/D=4$ (c) $z/D=5$ (d) $z/D=6$

Fig. 10: Comparison of autospectra measured at the centerline

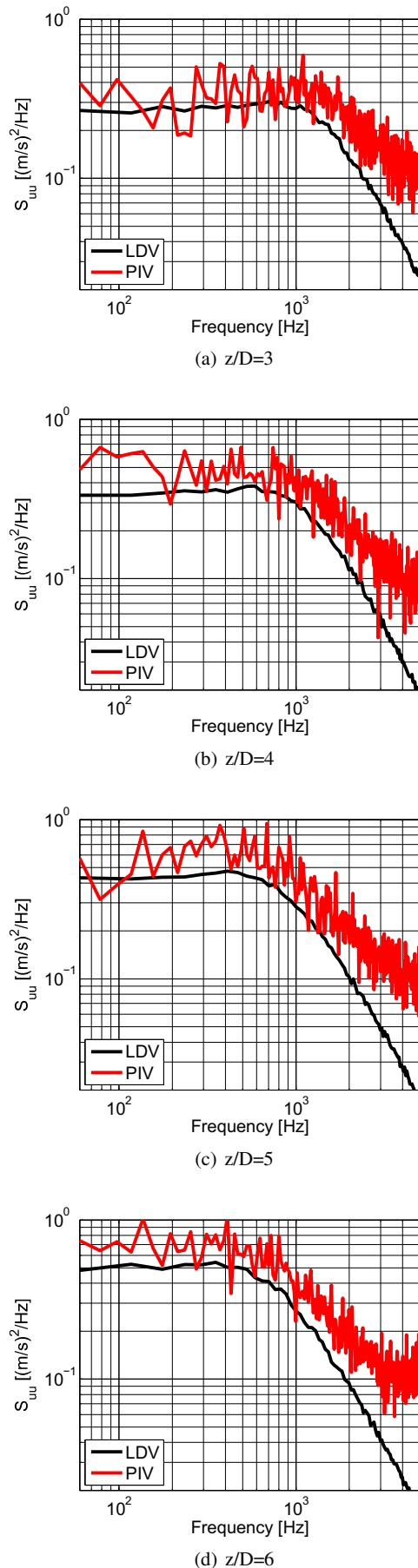


Fig. 11: Comparison of autospectra measured at the center of the shear layer

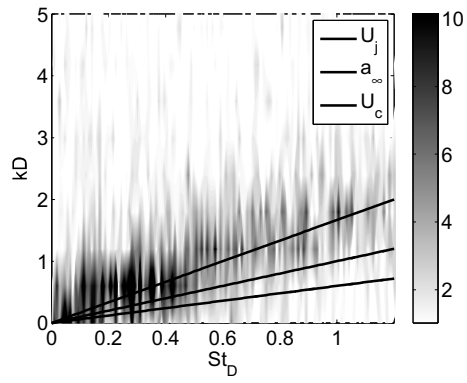
Acknowledgments

This work was supported by the National Science Foundation through the CNY-PR AGEP, the Air Force Office of Scientific Research, and the New York State Science, Technology, and Academic Research Office. The authors would also like to acknowledge Stanislav Kostka and Kerwin Low for collecting the TRPIV measurements. This work was supported in part by an AFOSR grant in conjunction with Clarkson University (FA9550-09-1-0051), with John Schmisser as program manager. The funding for this project was also provided by the Syracuse University STEM fellowship and the CNY-AGEP program. Lastly, the authors would like to acknowledge Spectral Energies, LLC. Phase I and II SBIR through AFRL.

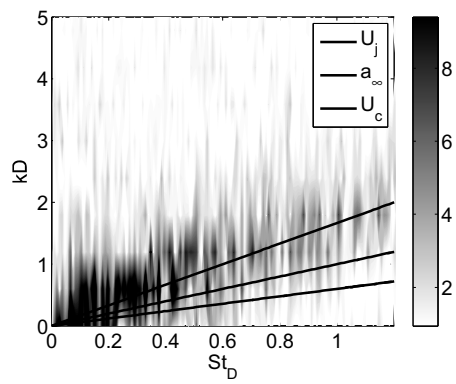
References

- (1995) Dantec Dynamics. Fiber Flow: Installation & User's guide
- (2006) Dantec Dynamics. BSA Flow Software: Version 4.10, Installation & User's Guide
- Adrian R, Yao C (1986) Power spectra of fluid velocities measured by laser doppler velocimetry. *Experiments in Fluids* 5(1):17–28
- Browne L, Antonia R, Chua L (1988) Calibration of X-probes for turbulent flow measurements. *Experiments in fluids* 7(3):201–208
- Bruun H, Principles HWA (1995) Signal analysis
- Buchhave P, George WKJ, Lumley JL (1979) The measurement of turbulence with the laser-doppler anemometer. *Annu Rev of Fluid Mech* 11:443–503
- Burattini P, Antonia R (2005) The effect of different X-wire calibration schemes on some turbulence statistics. *Experiments in fluids* 38(1):80–89
- Cantwell BJ (1975) A flying hot-wire study of the turbulent near wake of a circular cylinder at a Reynolds number of 140 000. PhD thesis, California Inst. Technology
- Cenedese A G and Doglia, Romano GP, Michele GD, Tanzini G (1994) LDA and PIV velocity measurements in free jets. *Exp Therm Fluid Sci* 9(2):125–134
- Coles D, Wadcock AJ (1979) Flying-hot-wire study of flow past an NACA 4412 airfoil at maximum lift. *AIAA J* 4(9):321–329
- Coles DR, Glauser MN (1998) Flying hot-wire measurements in an axisymmetric sudden expansion. *Exp Therm Fluid Sci* 18(2):150–167
- Comte-Bellot G (1976) Hot-wire anemometry. *Annu Rev of Fluid Mech* 8:209–231
- Durst F, Jovanovic J, Sender J (1995) LDA measurements in the near-wall region of a turbulent pipe flow. *Journal of Fluid Mechanics* 295:305–335
- Ewing D, Hussein H, George W (1995) Spatial resolution of parallel hot-wire probes for derivative measurements. *Experimental thermal and fluid science* 11(2):155–173
- Fiévet R, Tinney CE, Murray NE, Lyons G, Panickar P (2013) Acoustic source indicators using les in a fully expanded and heated supersonic jet. In: 19th AIAA/CEAS Aeroacoustics Conference (AIAA-2013-2193), Berlin, Germany
- Freund JB (2001) Noise sources in a low-reynolds-number turbulent jet at mach 0.9. *Journal of Fluid Mechanics* 438:277–305
- George WK, Lumley JL (1973) The laser-doppler velocimeter and its application to the measurement of turbulence. *Journal of Fluid Mechanics* 60(2):321–362

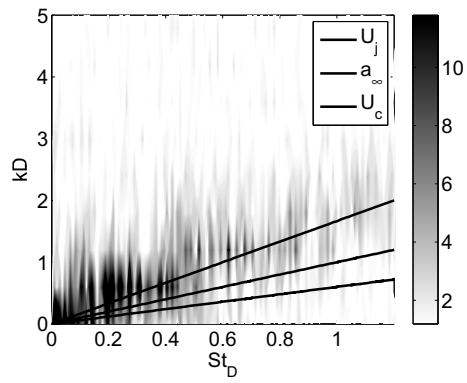
- Grosjean N, Graftieaux L, Michard M, Hubner W, Tropea C, Volkert J (1997) Combining LDA and PIV for turbulence measurements in unsteady swirling flows. *Meas Sci and Technol* 8:1523–1532
- Hall AM, Tinney CE, Glauser MN (2005) Investigating the ‘modified’ complementary technique using pressure-velocity correlations of an axisymmetric jet. In: 43rd AIAA Aerospace Sciences Meeting and Exhibit (AIAA-2005-0039)
- Hearst J, Shea P, Lavoie P, Glauser M (2014) Hot-wire versus LDA measurements for grid turbulence, private communication
- Hussein H, George W (1989) Measurements of small scale turbulence in an axisymmetric jet using moving hot-wires. In: 7th Symposium on Turbulent Shear Flows, pp 30.2.1–30.2
- Johnson F, Eckelmann H (1984) A variable angle method of calibration for x-probes applied to wall-bounded turbulent shear flow. *Experiments in fluids* 2(3):121–130
- Kerhervé F J Fitzpatrick, Kennedy J (2010) Determination of two-dimensional space-time correlations in jet flows using simultaneous PIV and LDV measurements. *Exp Therm Fluid Sci* 34:788–797
- Klewicki J, Falco R (1990) On accurately measuring statistics associated with small-scale structure in turbulent boundary layers using hot-wire probes. *J Fluid Mech* 219:119–142
- Lau JC, Morris PJ, Fisher MJ (1979) Measurements in subsonic and supersonic free jets using a laser velocimeter. *Experimental Thermal and Fluid Science* 93:1–27
- Lavoie P, Avallone G, De Gregorio F, Romano G, Antonia R (2007) Spatial resolution of PIV for the measurement of turbulence. *Experiments in fluids* 43(1):39–51
- Leder A, Brede M (2004) Comparison between 3D-LDA measurements and TR-PIV data in the separated flow of a circular cylinder. In: 12th International Symposium on Applications of Laser Techniques to Fluids Mechanics
- Low KR, Berger ZP, Kostka S, El Hadidi B, Gogineni S, Glauser MN (2013) Noise source identification and control in a Mach 0.6 turbulent jet with simultaneous time resolved PIV, pressure and acoustic measurements. *Experiments in Fluids* 54(4):1–17
- Mattern P, Sieber S, Dues M, Caglar S, Gabi M (2012) Simultaneous high speed stereo PIV and LDA measurements in the highly transient vortical wake in an axial fan. In: 16th International Symposium on Applications of Laser Techniques to Fluids Mechanics
- Murray N, Lyons G, Tinney C, Donald B, Baars W, Thurow B, Haynes H, Panickar P (2012) A laboratory framework for synchronous near/far-field acoustics and MHz PIV in high-temperature, shock-containing, jets. In: Proc. Internoise 2012 ASME NCAD meeting, pp 19–22
- Raffel M, Willert CE, Kompenhans J (1998) Particle Image Velocimetry: A Practical Guide; with 24 Tables. Springer
- Romano G, Antonia R, Zhou T (1999) Evaluation of LDA temporal and spatial velocity structure functions in a low Reynolds number turbulent channel flow. *Experiments in fluids* 27(4):368–377
- Shabbir A, Beuthert P, George W (1996) X-wire response in turbulent flows of high-intensity turbulence and low mean velocities. *Experimental thermal and fluid science* 12(1):52–56
- Tagawa M, Tsuji T, Nagano Y (1992) Evaluation of x-probe response to wire separation for wall turbulence measurements. *Experiments in fluids* 12(6):413–421
- Thompson B, Whitelaw J (1984) Flying hot-wire anemometry. *Exp Fluid* 2(1):47–55
- Tinney C, Glauser MN, Ukeiley L (2008a) Low-dimensional characteristics of a transonic jet. part 1. proper orthogonal decomposition. *J Fluid Mech* 612:107–141
- Tinney C, Ukeiley L, Glauser MN (2008b) Low-dimensional characteristics of a transonic jet. part 2. estimate and far-field prediction. *Journal of Fluid Mechanics* 615:53–92
- Tinney CE (2005) Low-dimensional techniques for sound source identification in high speed jets. PhD thesis, Syracuse University
- Tinney CE, Hall AM, Glauser MN, Ukeiley LS (2004) Designing and anechoic chamber for the experimental study of high speed heated jets. In: 42nd AIAA Aerospace Sciences Meeting and Exhibit (AIAA-2004-0010)
- Tutu NK, Chevray R (1975) Cross-wire anemometry in high intensity turbulence. *Journal of Fluid Mechanics* 71(4):785–800
- Ukeiley L, Tinney C, Mann R, Glauser M (2007) Spatial correlations in a transonic jet. *AIAA journal* 45(6):1357–1369
- Wattmuff J, Perry A, Chong M (1983) A flying hot-wire anemometry system. *Exp Fluids* 1(2):63–71
- Wernet MP (2007) Temporally resolved PIV for space-time correlations in both cold and hot jet flows. *Measurement Science and Technology* 18(5):1387
- Wyngaard J (1968) Measurement of small-scale turbulence structure with hot wires. *Journal of Physics E: Scientific Instruments* 1(11):1105
- Zhu Y, Antonia R (1996) The spatial resolution of hot-wire arrays for the measurement of small-scale turbulence. *Measurement Science and Technology* 7(10):1349



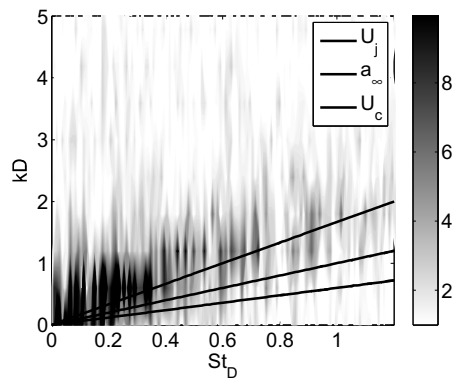
(a) $z/D=3$



(b) $z/D=4$



(c) $z/D=5$



(d) $z/D=6$

Fig. 9: Comparison of mean Mach number profiles

Bibliography

- [1] Berger, Z. P., *The Effects of Active Flow Control on High-Speed Jet Flow Physics and Noise*, Ph.D. thesis, Syracuse University, 2014.
- [2] Adrian, R. J., “Twenty years of particle image velocimetry,” *Experiments in Fluids*, Vol. 39, No. 2, 2005, pp. 159–169.
- [3] Westerweel, J., Elsinga, G. E., and Adrian, R. J., “Particle image velocimetry for complex and turbulent flows,” *Annual Review of Fluid Mechanics*, Vol. 45, 2013, pp. 409–436.
- [4] Versteeg, H. K. and Malalasekera, W., *An Introduction to Computational Fluid Dynamics: The Finite Element Method*, Pearson Education Limited, 2007.
- [5] Hall, D. L. and Llinas, J., “An Introduction to Multisensor Data Fusion,” *Proceedings of the IEEE*, Vol. 85, No. 1, 1997, pp. 6–23.
- [6] Steinberg, A. N., Bowman, C. L., and White, F. E., “Revisions to the JDL Data Fusion Model,” *American inst of aeronautics and astronautics*, 1998.
- [7] Wald, L., “Some Terms of Reference in Data Fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37, No. 3, 1999, pp. 1190–1193.
- [8] Synnergren, J., Gamalielsson, J., and Olsson, B., “Mapping of the JDL data fusion model to bioinformatics,” *IEEE International Conference on Systems, Man and Cybernetics, 2007. ISIC.*, 2007, pp. 1506–1511.
- [9] Abdelgawad, A. and Bayoumi, M., *Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks*, Springer Science, New York, 2012.

- [10] Giacobe, N. A., "Application of the JDL data fusion process model for cyber security," *Proceedings of the Society of Photo-Optical Instrumentation Engineers*, Vol. 7710, 2010.
- [11] Dayal, U., Castellanos, M., Simitsis, A., and Wilkinson, K., "Data Integration Flows for Business intelligence," *EDBT*, 2009.
- [12] Herrera, P. E., Kaufmann, H., Secue, J., Quiros, R., and Fabregat, G., "Improving data fusion in personal positioning systems for outdoor environments," *Information Fusion*, Vol. 14, 2013, pp. 45–56.
- [13] Fay, D., Waxman, A., Aguilar, M., Ireland, D., Racamato, J., Ross, W., Streilein, W., and Braun, M., "Fusion of Multi-Sensor Imagery for Night Vision: Color Visualization, Target Learning and Search," *Information Fusion*, Vol. 1, 2000.
- [14] Burzinski, T., "Business Intelligence: Today's Decisions, Yesterday's Data," 2004, www.devx.com/opinion/article/20304.
- [15] Sasiadek, J., Wang, Q., and Zeremba, M., "Fuzzy adaptive Kalman filtering for INS/GPS data fusion," *International Symposium on Intelligent Control*, 2000.
- [16] Abdel-Aleem, M., Chibelushi, C. C., and Moniri, M., "Multisensor Data Fusion for the Simultaneous Location and Condition Assessment of Underground Water Pipelines," *International Conference on Networking, Sensing and Control*, Vol. 66, Delft, the Netherlands, 11-13 April 2011.
- [17] Ultra Electronics Limited <<http://www.ultra-ccs.com/business/bordersecurity>>.
- [18] Yocky, David, A., "Image Merging and Data Fusion by Means of the Discrete Two-Dimensional Wavelet Transform," *Journal of the Optical Society of America A.*, Vol. 12, No. 9, 1995, pp. 1834–1841.
- [19] Varshney, Pramod, K., "Multisensor data fusion," *Electronics and Communication Engineering Journal*, 1997, pp. 245–253.

- [20] Chen, H.-M., Lee, S., Rao, R. M., Slamani, M.-A., and Varshney, P. K., “Imaging for concealed weapon detection: a tutorial overview of development in imaging sensors and processing,” *Signal Processing Magazine, IEEE*, Vol. 22, No. 2, 2005, pp. 52–61.
- [21] Burt, P. J., *The pyramid as a structure for efficient computation*, Springer, 1984.
- [22] Pajares, G. and Manuel de la Cruz, J., “A wavelet-based image fusion tutorial,” *Pattern Recognition*, Vol. 37, 2004, pp. 1855–1872.
- [23] Zhang, Z. and Blum, R. S., “A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application,” *Proceedings of the IEEE*, Vol. 87, No. 8, 1999, pp. 1315–1326.
- [24] Pinier, J. T., Ausseur, J. M., Glauser, M. N., and Higuchi, H., “Proportional closed-loop feedback control of flow separation,” *AIAA*, Vol. 45, 2007.
- [25] Shea, P. R., Berger, Z. P., Berry, M. G., Glauser, M. N., and Gogineni, S., “Low-Dimensional Modeling of a Mach 0.6 Axisymmetric Jet,” *52nd AIAA ASM*, National Harbor, MD, 13-17 January 2014, AIAA-2014-0245.
- [26] Shea, P. and Glauser, M. N., “Experimental Investigation of a Three-Dimensional Turret Wake with Active Flow Control,” 2012, AIAA-2012-0745.
- [27] Adrian, R. J., “Particle-imaging techniques for experimental fluid mechanics,” *Annual review of fluid mechanics*, Vol. 23, No. 1, 1991, pp. 261–304.
- [28] Raffel, M., Willert, C. E., and Kompenhans, J., *Particle Image Velocimetry: A Practical Guide; with 24 Tables*, Springer, 1998.
- [29] Wei, Y. and Chen, N., “Square Wave Analysis,” *Journal of Mathematical Physics*, Vol. 39, No. 8, 1998.
- [30] Tinney, C., Glauser, M. N., and Ukeiley, L., “Low-dimensional characteristics of a transonic jet. Part 1. Proper orthogonal decomposition,” *Journal of Fluid Mechanics*, Vol. 612, 2008, pp. 107–141.

- [31] Lumley, J. L., “The Structures of Inhomogeneous Turbulent Flows,” *Atmospheric Turbulence and Radio Wave Propagation*, 1967.
- [32] Berkooz, G., Holmes, P., and Lumley, J. L., “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, Vol. 25, No. 1, 1993, pp. 539–575.
- [33] Sirovich, L. and Kirby, M., “Low-dimensional procedure for the characterization of human faces,” *Journal of Optical Society of America*, Vol. 102, 1987, pp. 519–524.
- [34] Camphouse, R. C., Myatt, J. H., Schmit, R. F., Glauser, M. N., Auesseur, J. M., Andinio, M. Y., and Wallace, R. D., “A Snapshot Decomposition Method for Reduced Order Modeling and Boundary Feedback Control,” 2008, AIAA-2008-4195.
- [35] Aradag, S., Siegel, S., Seidel, J., Cohen, K., and McLaughlin, T., “Filtered POD-based low-dimensional modeling of the 3D turbulent flow behind a circular cylinder,” *International Journal for Numerical Methods in Fluids*, Vol. 66, 2011.
- [36] Sirovich, L., “Turbulence and the Dynamic Coherent structures Part 1: Coherent Structures,” *Quarterly Applied Mathematics*, Vol. 45, 1987.
- [37] Everson, R. and Sirovich, L., “Karhunen–Loève procedure for gappy data,” *Journal of Optical Society of America*, Vol. 12, No. 8, 1995, pp. 1657–1664.
- [38] Cavar, D. and Meyer, K. E., “LES of turbulent jet in cross flow: Part 2–POD analysis and identification of coherent structures,” *International Journal of Heat and Fluid Flow*, Vol. 36, 2012, pp. 35–46.
- [39] Ruscher, C. J., Dannenhoffer III, J. F., Glauser, M. N., Kiel, B. V., and Sekar, B., “Vortex Shedding of Various Bluff Bodies in Cross Flow,” *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Atlanta, GA, July 30 - August 01 2012, AIAA-2012-4285.

- [40] Cazemier, W., Verstappen, R., and Veldman, A., “Proper orthogonal decomposition and low-dimensional models for driven cavity flows,” *Physics of fluids*, Vol. 10, 1998, pp. 1685–1699.
- [41] Bui-Thanh, R., Damodaran, M., and Willcox, K., “Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition,” *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1505–1516.
- [42] Willcox, K., “Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition,” *Computers & fluids*, Vol. 35, No. 2, 2006, pp. 208–226.
- [43] Ruscher, C. J., Dannenhoffer, J. F., and Glauser, M. N., “Application of Snapshot POD on a Varying Grid,” *51st AIAA ASM*, Grapevine, Texas, 7-10 January 2013, AIAA-2013-0256.
- [44] Murray, N. E. and Ukeiley, L. S., “An application of Gappy POD for subsonic cavity flow PIV data,” *Experiments in Fluids*, Vol. 42, 2007, pp. 79–91.
- [45] Raben, S. G., Charonko, J. J., and Vlachos, P. P., “Adaptive gappy proper orthogonal decomposition for particle image velocimetry data reconstruction,” *Measurement Science and Technology*, Vol. 23, No. 2, 2012, pp. 025303.
- [46] Adrian, R. J., “On the Role of Conditional Averages in Turbulence Theory,” *Turbulence in Liquids: Preceedings of the Fourth Biennial Symposium on Turbulence in Liquids*, Vol. 66, 1977, pp. 323–332.
- [47] Tropea, C., Yarin, A., and Foss, J., *Springer Handbook of Experimental Fluid Mechanics*, Springer-Verlag Berlin Heidelberg, 2007.
- [48] Bonnet, J.-P., Cole, D., Delville, J., Glauser, M. N., and Ukeiley, L., “Stochastic Estimation and Proper Orthogonal Decomposition: Complementary Techniques for Identifying Structure,” *Experiments in Fluids*, Vol. 17, No. 5, 1994, pp. 307–314.

- [49] Taylor, J. A. and Glauser, M. N., “Towards Practical Flow Sensing and Control via POD and LSE Based Low-Dimensional Tools,” *Journal of Fluids Engineering*, Vol. 126, No. 3, 2004, pp. 337–345.
- [50] Naguib, A., Wark, C., and Juckenhöfel, O., “Stochastic estimation and flow sources associated with surface pressure events in a turbulent boundary layer,” *Physics of Fluids*, Vol. 13, 2001, pp. 2611.
- [51] Murray, N. E. and Ukeiley, L. S., “Estimation of the flowfield from surface pressure measurements in an open cavity,” *AIAA journal*, Vol. 41, No. 5, 2003, pp. 969–972.
- [52] Tung, T. and Adrian, R. J., “Higher-order estimates of conditional eddies in isotropic turbulence,” *Physics of Fluids*, Vol. 23, 1980, pp. 1469–1470.
- [53] Tu, J. H., Griffin, J., Hart, A., Rowley, C. W., Cattafesta III, L. N., and Ukeiley, L. S., “Integration of non-time-resolved PIV and time-resolved velocity point sensors for dynamic estimation of velocity fields,” 2013.
- [54] Ukeiley, L. and Murray, N., “Velocity and surface pressure measurements in an open cavity,” *Experiments in Fluids*, Vol. 38, No. 5, 2005, pp. 656–671.
- [55] Durgesh, V. and Naughton, J., “Multi-time-delay LSE-POD complementary approach applied to unsteady high-Reynolds-number near wake flow,” *Experiments in fluids*, Vol. 49, No. 3, 2010, pp. 571–583.
- [56] Ewing, D. and Citriniti, J. H., “Examination of a LSE/POD complementary technique using single and multi-time information in the axisymmetric shear layer,” *IUTAM Symposium on simulation and identification of organized structures in flows*, Springer, 1999, pp. 375–384.
- [57] Tinney, C., Ukeiley, L., and Glauser, M. N., “Low-dimensional characteristics of a transonic jet. Part 2. Estimate and far-field prediction,” *Journal of Fluid Mechanics*, Vol. 615, 2008, pp. 53–92.

- [58] Adrian, R., Jones, B., Chung, M., Hassan, Y., Nithianandan, C., and Tung, A.-C., “Approximation of turbulent conditional averages by stochastic estimation,” *Physics of Fluids A: Fluid Dynamics*, Vol. 1, 1989, pp. 992.
- [59] Gutmark, E. J., Veraille, S., Bonnet, J.-P., and Grinstein, F., “Linear Stochastic Estimation of a Swirling Jet,” *AIAA Journal*, Vol. 44, No. 3, 2006, pp. 457 – 468.
- [60] Tinney, C., Jordan, P., Hall, A. M., Delville, J., and Glauser, M. N., “A Time-Resolved Estimate of the Turbulence and Sound Source Mechanisms in a Subsonic jet flow,” *Journal of Turbulence*, Vol. 8, No. 7, 2007.
- [61] Hall, J. W. and Ewing, D., “Spectral linear stochastic estimation of the turbulent velocity in a square three-dimensional wall jet,” *Journal of Fluids Engineering*, Vol. 132, No. 5, 2010, pp. 051203.
- [62] Garcia-Sagrado, A. and Hynes, T., “Stochastic estimation of flow near the trailing edge of a NACA0012 airfoil,” *Experiments in fluids*, Vol. 51, No. 4, 2011, pp. 1057–1071.
- [63] Aubry, N., Holmes, P., Lumley, J. L., and Stone, E., “The dynamics of coherent structures in the wall region of a turbulent boundary layer,” *Journal of Fluid Mechanics*, Vol. 192, No. 1, 1988, pp. 115–173.
- [64] Rempfer, D., “Investigations of boundary layer transition via Galerkin projections on empirical eigenfunctions,” *Physics of Fluids*, Vol. 8, 1996, pp. 175.
- [65] Ukeiley, L., Cordier, L., Manceau, R., Delville, J., Glauser, M., and Bonnet, J.-P., “Examination of large-scale structures in a turbulent plane mixing layer. Part 2. Dynamical systems model,” *Journal of Fluid Mechanics*, Vol. 441, No. 1, 2001, pp. 67–108.
- [66] Ausseur, J. M. and Pinier, J. T., “Towards closed-loop feedback control of the flow over NACA-4412 airfoil,” *43rd AIAA ASM*, Reno, NV, 10-13 January 2005, AIAA-2005-343.

- [67] Ausseur, J. M., *Application of Low-Dimensional Techniques for Closed-Loop Control of Turbulent Flows*, Ph.D. thesis, Syracuse University, 2007.
- [68] Cordier, L., Noack, B. R., Tissot, G., Lehnasch, G., Delville, J., Balajewicz, M., Daviller, G., and Niven, R. K., “Identification strategies for model-based control,” *Experiments in fluids*, Vol. 54, No. 8, 2013, pp. 1–21.
- [69] Noack, B. R. and Eckelmann, H., “A low-dimensional Galerkin method for the three-dimensional flow around a circular cylinder,” *Physics of Fluids*, Vol. 6, 1994, pp. 124.
- [70] Rowley, C. W., Colonius, T., and Murray, R. M., “Model reduction for compressible flows using POD and Galerkin projection,” *Physica D: Nonlinear Phenomena*, Vol. 189, No. 1, 2004, pp. 115–129.
- [71] Kalman, R. E., “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.
- [72] Burgers, G., Van Leeuwen, P. J., and Evensen, G., “Analysis scheme in the ensemble Kalman filter.” *Monthly weather review*, Vol. 126, No. 6, 1998.
- [73] Panzieri, S., Pascucci, F., and Setola, R., “Interlaced extended kalman filter for real time navigation,” *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, 2005, pp. 2780–2785.
- [74] Hu, C., Chen, W., Chen, Y., and Liu, D., “Adaptive Kalman filtering for vehicle navigation,” *Positioning*, Vol. 1, 2009, pp. 0.
- [75] Farge, M., “Wavelet Transforms and their applications to turbulence,” *Annual Review of Fluid Mechanics*, Vol. 24, 1992, pp. 395–457.
- [76] Lewalle, J., Delville, J., and Bonnet, J.-P., “Decomposition of Mixing Layer Turbulence into Coherent Structures and Background Fluctuations,” *Flow, Turbulence and Combustion*, Vol. 64, 2000, pp. 301 – 328.
- [77] Meneveau, C., “Analysis of turbulence in the orthonormal wavelet representation,” *Journal of Fluid Mechanics*, Vol. 232, 1991, pp. 469–520.

- [78] Chainais, P., Abry, P., and Pinton, J.-F., “Intermittency and coherent structures in a swirling flow: A wavelet analysis of joint pressure and velocity measurements,” *Physics of Fluids*, Vol. 11, 1999, pp. 3524–3539.
- [79] Camussi, R. and Guj, G., “Orthonormal wavelet decomposition of turbulent flows: intermittency and coherent structures,” *Journal of Fluid Mechanics*, Vol. 348, No. 1, 1997, pp. 177–199.
- [80] Schobeiri, M. T., Read, K., and Lewalle, J., “Effect of Unsteady Wake Passing Frequency on Boundary Layer Transition, Experimental Investigation, and Wavelet Analysis,” *Journal of Fluids Engineering*, Vol. 125, 2003, pp. 251–266.
- [81] Farge, M., Schneider, K., Pellegrino, G., Wray, A. A., and Rogallo, R. S., “Coherent vortex extraction in three-dimensional homogeneous turbulence: Comparison between CVS-wavelet and POD-Fourier Decompositions,” *Physics of Fluids*, Vol. 15, No. 10, 2003, pp. 2886–2896.
- [82] Schneider, K., Farge, M., Pellegrino, G., and Rogers, M. M., “Coherent vortex simulation of three-dimensional turbulent mixing layers using orthogonal wavelets,” *Journal of Fluid Mechanics*, Vol. 534, 2005, pp. 39–66.
- [83] Haar, A., “Zur Theorie der orthogonalen Funktionensysteme [On the Theory of Orthogonal Function Systems],” *Mathematische Annalen*, Vol. 69, 1910, pp. 331–371, Translated by Georg Zimmermann.
- [84] Daubechies, I., “Orthogonal Bases of Compactly Supported Wavelets,” *Communications on Pure and Applied Mathematics*, Vol. 41, No. 7, 1988, pp. 909–996.
- [85] Ramac, L. C., Uner, M. K., Varshney, P. K., Alford, M. G., and Ferris Jr, D. D., “Morphological filters and wavelet-based image fusion for concealed weapons detection,” *Aerospace/Defense Sensing and Controls*, International Society for Optics and Photonics, 1998, pp. 110–119.
- [86] Nunez, J., Otazu, X., Fors, O., Prades, A., Pala, V., and Arbiol, R., “Multiresolution-based image fusion with additive wavelet decomposition,” *Geoscience and Remote Sensing, IEEE Transactions on*, Vol. 37, No. 3, 1999, pp. 1204–1211.

- [87] Lewis, A. S. and Knowles, G., “Image compression using the 2-D wavelet transform,” *Image Processing, IEEE Transactions on*, Vol. 1, No. 2, 1992, pp. 244–250.
- [88] Calderbank, A. R., Daubechies, I., Sweldens, W., and Yeo, B.-L., “Lossless image compression using integer to integer wavelet transforms,” *Image Processing, 1997. Proceedings., International Conference on*, Vol. 1, IEEE, 1997, pp. 596–599.
- [89] Chang, S. G., Yu, B., and Vetterli, M., “Adaptive wavelet thresholding for image denoising and compression,” *Image Processing, IEEE Transactions on*, Vol. 9, No. 9, 2000, pp. 1532–1546.
- [90] Kivanc Mihcak, M., Kozintsev, I., Ramchandran, K., and Moulin, P., “Low-complexity image denoising based on statistical modeling of wavelet coefficients,” *Signal Processing Letters, IEEE*, Vol. 6, No. 12, 1999, pp. 300–303.
- [91] Sevcenco, I. S., Hampton, P. J., and Agathoklis, P., “Seamless Stitching of Images Based on Haar Wavelet 2D Integration Method,” *Digital Signal Processing (DSP), 2011 17th International Conference on IEEE*, 2011, pp. 1–6.
- [92] Hampton, P. J., Agathoklis, P., and Bradley, C., “A New Wave-Front Reconstruction Method for Adaptive Optics Systems Using Wavelets,” *IEEE Journal of Selected Topics in Signal Processing*, Vol. 2, No. 5, 2008, pp. 781–792.
- [93] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes 3^d Edition: The Art of Scientific Computing*, Cambridge University Press, 2007.
- [94] Planquart, P., “Integration of CFD and Experimental Results at VKI in Low-Speed Aerodynamic Design,” *3rd International Symposium on Integrating CFD and Experiments in Aerodynamics*, 2007.
- [95] Zimmermann, R., Vendl, A., and Görtz, S., “Reduced order modeling of steady flows subject to aerodynamic constraints,” *AIAA Journal*, 2014.

- [96] Vendl, A. and Faßbender, H., “Proper orthogonal decomposition for steady aerodynamic applications,” *PAMM*, Vol. 10, No. 1, 2010, pp. 635–636.
- [97] Berger, Z. P., ”Personal Communication”, 2014.
- [98] Ruscher, C. J., *User Guide for Orange High Speed Data: Version 3.4.5.0-December 2013*, 2013.
- [99] Low, K. R., Berger, Z. P., Kostka, S., El Hadidi, B., Gogineni, S., and Glauser, M. N., “Noise Source Identification and Control in a Mach 0.6 Turbulent Jet with Simultaneous Time Resolved PIV, Pressure and Acoustic Measurements,” *Experiments in Fluids*, Vol. 54, No. 4, 2013, pp. 1–17.
- [100] Berger, Z. P., Berry, M. G., Shea, P. R., Glauser, M. N., Jiang, N., Gogineni, S., Eurika, K., Noack, B. R., and Spohn, A., “Analysis of High Speed Jet Flow Physics with Time-Resolved PIV,” *52nd AIAA ASM*, National Harbor, MD, 13-17 January 2014, AIAA-2014-1226.
- [101] Berger, Z. P., Low, K. R., Kostka, S., Gogineni, S., and Glauser, M. N., “Investigation of an Axisymmetric Transonic Jet with High Resolution Time Resolved PIV,” *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Atlanta, GA, 30 July - 01 August 2012, AIAA-2012-3822.
- [102] Berry, M. G., ”Personal Communication”.
- [103] Hall, A. M., Tinney, C. E., and Glauser, M. N., “Investigating the ‘modified’ Complementary Technique Using Pressure-Velocity Correlations of an Axisymmetric Jet,” *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005, AIAA-2005-0039.
- [104] Ruscher, C. J., Berger, Z. P., Shea, P. R., Hall, A. M., Tinney, C. E., Berry, M. G., Glauser, M. N., Gogineni, S., and Kiel, B. V., “Comparison of High Speed Optical Measurements in a Jet Flow,” In process toward submittal to *Experiments in Fluids*.
- [105] Low, K. R., *Towards Closed Loop Control of a High-Speed Jet for Noise Reduction*, Ph.D. thesis, Syracuse University, 2014.

- [106] Berger, Z. P., Low, K. R., Berry, M. G., Glauser, M. N., Kostka, S., Gogineni, S., Cordier, L., and Noack, B. R., “Reduced Order Models for High Speed Jet with Time-Resolved PIV,” *51st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Grapevine, TX, 07 - 10 January 2013, AIAA-2013-0011.
- [107] Berger, Z. P., Berry, M. G., Shea, P. R., Glauser, M. N., Noack, B. R., and Gogineni, S., “Active Flow Control For High Speed Jets Using Advanced Modeling Copled with PIV,” *International Symposium on Turbulence and Shear Flow Phenomena*, Poitier, France, 28 - 30 August 2013.
- [108] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill, 2001.
- [109] Abbott, I. H., Von Doenhoff, A. E., and Stivers Jr, L., “Summary of airfoil data,” Tech. rep., 1945.
- [110] Jacobs, E. N., Ward, K. E., and Pinkerton, R. M., “The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel,” Tech. rep., DTIC Document, 1933.
- [111] Theodorsen, T., *Theory of wing sections of arbitrary shape*, US Government Printing Office, 1932.
- [112] Bertin, J. J. and Smith, M. L., “Aerodynamics for engineers,” 1998.
- [113] Robinson, A. and Laurmann, J. A., *Wing Theory*, Cambridge University Press, 2013.
- [114] Abbott, I. H. and Von Doenhoff, A. E., *Theory of wing sections: including a summary of airfoil data*, Courier Dover Publications, 2012.
- [115] Nichols, R. H. and Buning, P. G., *Version 2.1t Aug 2008 User’s Manual for OVERFLOW 2.1*, 2008.
- [116] Thompson, J. F., Soni, B. K., and Weatherill, N. P., *Handbook of grid generation*, CRC press, 1998.

- [117] Suhs, N., Rogers, S., and Dietz, W., “PEGASUS 5: An Automated Pre-Processor for Overset-Grid CFD,” *AIAA Journal*, Vol. 41, 2003, pp. 1037–1045.
- [118] Thompson, J. F., Warsi, Z., and Mastin, C. W., *Numerical Grid Generation: Foundations and Applications*, North Holland, 1985.
- [119] Menter, F., “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications,” *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.
- [120] Smagorinsky, J., “General circulation experiments with the primitive equations: I. The basic experiment*,” *Monthly weather review*, Vol. 91, No. 3, 1963, pp. 99–164.
- [121] Spalart, P. R., “Detached-Eddy Simulation,” *Annual Review of Fluid Mechanics*, Vol. 41, 2009, pp. 181–202.
- [122] Spalart, P. R., Deck, S., Shur, M., Squires, K., Strelets, M. K., and Travin, A., “A new version of detached-eddy simulation, resistant to ambiguous grid densities,” *Theoretical and computational fluid dynamics*, Vol. 20, No. 3, 2006, pp. 181–195.
- [123] Ramac, L. C. and Varshney, P. K., “A wavelet domain diversity method for transmission of images over wireless channels,” *Selected Areas in Communications, IEEE Journal on*, Vol. 18, No. 6, 2000, pp. 891–898.
- [124] Terino, J. G., “Cities spread to once-remote airports,” *Environmental Science & Technology*, Vol. 10, No. 9, 1976, pp. 872–875.
- [125] Fields, J. M., “Effect of personal and situational variables on noise annoyance in residential areas,” *The Journal of the Acoustical Society of America*, Vol. 93, 1993, pp. 2753.
- [126] Espey, M. and Lopez, H., “The impact of airport noise and proximity on residential property values,” *Growth and Change*, Vol. 31, No. 3, 2000, pp. 408–419.
- [127] Viswanathan, K. and Pilon, A. R., “Aeroacoustics,” *Aerospace America: Year in Review 2011*, Dec. 2011.

- [128] Trost, R. P. and Shaw, G. B., “Statistical analysis of hearing loss among navy personnel,” *Military medicine*, Vol. 172, No. 4, 2007, pp. 426–430.
- [129] Lighthill, M. J., “On Sound Generated Aerodynamically I. General Theory,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 1951.
- [130] Ffowcs Williams, J. and Hawkings, D., “Theory Relating to the Noise of Rotating Machinery,” *Journal of Sound and Vibration*, Vol. 10, No. 1, 1969, pp. 10–21.
- [131] Crow, S. C. and Champagne, F. H., “Orderly Structure in Jet Turbulence,” *Journal of Fluid Mechanics*, Vol. 48, No. 3, 1971, pp. 547–591.
- [132] Brown, G. L. and Roshko, A., “On Density Effects and Large Structure in Turbulent Mixing Layers,” *Journal of Fluid Mechanics*, Vol. 64, 1974, pp. 775–816.
- [133] Crighton, D., “Basic Principle of Aerodynamic Noise Generation,” *Progress in Aerospace Sciences*, Vol. 16, No. 1, 1975, pp. 31–96.
- [134] Glauser, M. N., Leib, S. J., and George, W. K., “Coherent Structures in Axisymmetric Turbulent Jet Mixing Layer,” *Turbulent Shear Flows 5*, 1987, pp. 134–145.
- [135] Saiyed, N. H., Mikkelsen, K. L., and Bridges, J. E., “Acoustics and thrust of separate-flow exhaust nozzles with mixing devices for high-bypass-ratio engines,” *6th AIAA/CEAS Aeroacoustics Conference and Exhibit(21st AIAA Aeroacoustic Conference)*, Lahaiiaa, HI, 12-14 June 2000, AIAA-2000-1961.
- [136] Mengle, V. G., “Jet noise characteristics of chevrons in internally mixed nozzles,” *11th AIAA/CEAS Aeroacoustics Conference*, Monterey, CA, 23-25 May 2005, AIAA-2005-2934.
- [137] Alkislar, M. B., Krothapalli, A., and Butler, G., “The effect of streamwise vortices on the aeroacoustics of a Mach 0.9 jet,” *Journal of Fluid Mechanics*, Vol. 578, 2007, pp. 139–169.

- [138] Sinha, A., Serrani, A., and Samimy, M., “Development of Empirical Estimators for Feedback Control of High-Speed Axisymmetric Jets,” *AIAA journal*, Vol. 49, No. 9, 2011, pp. 1971–1987.
- [139] Low, K., El Hadiddi, B., Andino, M. Y., Berdanier, R., and Glauser, M. N., “Investigation of Different Active Flow Control Strategies for High Speed Jets Using Synthetic Jet Actuators,” *AIAA 40th Fluids Dynamics Conference and Exhibit*, 2011.
- [140] Cavalieri, A. V., Jordan, P., Colonius, T., and Gervais, Y., “Using large eddy simulation to explore sound-source mechanisms in jets,” *Journal of Fluid Mechanics*, Vol. 704, 2012, pp. 388–420.
- [141] Colonius, T., Lele, S. K., and Moin, P., “Sound Generation in a Mixing Layer,” *Journal of Fluid Mechanics*, Vol. 330, 1997, pp. 375–409.
- [142] Freund, J. B., “Noise Sources in a Low-Reynolds-Number Turbulent jet at Mach 0.9,” *Journal of Fluid Mechanics*, Vol. 438, 2001, pp. 277–305.
- [143] Self, R., “Jet noise prediction using the Lighthill acoustic analogy,” *Journal of Sound and Vibration*, Vol. 275, 2003, pp. 757–768.
- [144] Bodony, D. J. and Lele, S. K., “On Using Large-Eddy Simulation for the Prediction of Noise from Cold and Heated Turbulent Jets,” *Physics of Fluids*, Vol. 17, 2005.
- [145] Tucker, P., “The LES model’s Role in Jet Noise,” *Progress in Aerospace Sciences*, Vol. 44, 2008, pp. 427–436.
- [146] Bodony, D. J. and Lele, S. K., “Low-frequency sound sources in high-speed turbulent jets,” *Journal of Fluid Mechanics*, Vol. 617, 2008, pp. 231–253.
- [147] Cavalieri, A. V., Daviller, G., Comte, P., Jordan, P., Tadmor, G., and Gervais, Y., “Using large eddy simulation to explore sound-source mechanisms in jets,” *Journal of sound and Vibration*, Vol. 330, 2011, pp. 4098–4113.
- [148] Tennekes, H. and Lumley, J., *A First Course in Turbulence*, MIT Press, 1972.

- [149] Citriniti, J. and George, W., “Reconstruction of the global velocity field in the axisymmetric mixing layer utilizing the proper orthogonal decomposition,” *Journal of Fluid Mechanics*, Vol. 418, 2000, pp. 137–166.
- [150] Bodony, D. J. and Lele, S. K., “Current Status of Jet Noise Prediction Using Large-Eddy Simulation,” *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 364–380.
- [151] Tinney, C. E., Hall, A. M., Glauser, M. N., and Ukeiley, L. S., “Designing and Anechoic Chamber for the Experimental Study of High Speed Heated Jets,” 2004, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV AIAA-2004-0010.
- [152] Kan, P., Lewalle, J., and Gogineni, S. P., “Mach-number dependence of acoustic source properties in high speed jets—Part II: event-based description,” 52nd AIAA ASM, National Harbor, MD, 13-17 January 2014, AIAA-2014-1229.
- [153] Lewalle, J., Kan, P., and Gogineni, S. P., “Mach-number dependence of acoustic source properties in high speed jets—Part I: ensemble statistics of active regions,” 52nd AIAA ASM, National Harbor, MD, 13-17 January 2014, AIAA-2014-1228.
- [154] Shea, P. R., ”Personal Communication”.
- [155] Lewalle, J., ”Personal Communication”.
- [156] Ukeiley, L., Tinney, C., Mann, R., and Glauser, M., “Spatial correlations in a transonic jet,” *AIAA journal*, Vol. 45, No. 6, 2007, pp. 1357–1369.
- [157] Anderson, J. D., *Modern Compressible Flow*, McGraw-Hill, 3rd ed., 2003.

Christopher John Ruscher

Office: 263 Link Hall
Syracuse, NY 13244

Home: 233 Robert Drive
Syracuse, NY 13210
(585)-436-5032

Permanent: 6 Cheshire Lane
Rochester, NY 14624

E-mail: cjrusche@syr.edu
cjrusche@gmail.com

Citizenship: United States of America

Date of CV: 24 February 2014

Date of Birth: 03 December 1986

Research Interests: Data Fusion for Fluid Dynamics, Data analysis, Computational Fluid Dynamics, High-Speed Jet Flows

Education:

2014 **Ph.D.** *In Progress (Anticipated Graduation: May 2014)*
Institution: Syracuse University; Syracuse, NY
Degree: Mechanical and Aerospace Engineering
Advisor: John F. Dannenhoffer, III
Co-Advisor: Mark N. Glauser
Ph.D. GPA: 4.0/4.0

2011 **M.S.** **Institution:** Syracuse University; Syracuse, NY
Degree: Mechanical and Aerospace Engineering
Advisor: John F. Dannenhoffer, III
Co-Advisor: Mark N. Glauser
M.S. GPA: 3.95/4.0

2009 **B.S.** **Institution:** Syracuse University; Syracuse, NY
Degree: Aerospace Engineering
Minor: Math
Minor: Engineering Management
Advisor: John F. Dannenhoffer, III
B.S. GPA: 3.92/4.0

Work Experience:

08/2011 - Present **Graduate Research Assistant**
Syracuse University; Syracuse, NY

- Designed a data processing tool for fluid dynamic data sets
- Used the idea of data fusion to improve the understanding of fluid dynamic problems

- 05/2012 - 08/2012 **AFRL Summer Research Fellowship:**
Air Force Research Laboratory; WPAFB, OH
- Ran Simulation to predict the boundary conditions for a new combustion Rig.
 - Helped process combustion data using Proper Orthogonal Decomposition, Wavelets, and Correlations.
- 06/2011-08/2011 **AFRL Summer Research Fellowship:**
Air Force Research Laboratory; WPAFB, OH
- Run Simulation of Bluff bodies in a cross flow
 - Design and implement a proper orthogonal decomposition code to analyze the simulation data
 - Compare the modal energy of the simulation data to experimental data
- 08/2010 - 06/2011 **Graduate Research Assistant**
Syracuse University; Syracuse, NY
- Designed an ejector using CFD
 - Ejector was to be used to decrease static pressure in a combustion rig used to model a jet afterburner
 - The ejector was to be used to study combustion at low pressures
- 01/2010 - 07/2010 **Graduate Assistant**
Syracuse University; Syracuse, NY
- Designed a Simple Website for a professor
 - Helped organized a poster competition for engineering graduate students at Syracuse University
- 06/2009 - 08/2010 **Graduate Research Assistant**
Syracuse University; Syracuse, NY
- Worked with experimental engineers to create a CFD model of a spherical Laser Turret with a flat window
 - The CFD model will be used to guide the experiment by running different control schemes
 - Used a wind tunnel to obtain boundary conditions
- 06/2008 - 08/2008 **Summer Intern with the Last Stage Bucket Team**
General Electric; Schenectady, NY
- Wrote many codes using MatLab that improved the efficiency of the other engineers. Most of these codes were used to figure out why some of the buckets were forming cracks. These codes were primarily used to display where the cracking occurred.

06/2007 - 08/2007 Summer Intern with the Sealing Group for Steam Turbine*General Electric; Schenectady, NY*

- Wrote a Visual Basic program to predict the stress in labyrinth packing rings. Utilized old documents to assist in understanding how stress was calculated in the past, as a result, new calculations were written to accurately calculate the stress in the packing rings. The program was finally written to automate the new calculations.
- Categorized different packing rings. Utilized old drawings and found trends that allowed me to convert the old drawings into multiple types of rings. Updated the newer drawing by adding the new type of packing ring to the drawing.

06/2006 - 08/2006 Summer Intern with the Heat Transfer and Ventilation Group*General Electric; Schenectady, NY*

- Wrote Visual Basic programs to improve the efficiency of engineering processes. The program interacted with other programs, performed calculations, and stored results in Excel.
- Analyzed the calculation of gas properties for a mixture of gases. Wrote a report on the findings.

Honors and Awards:

2011 Outstanding Mechanical and Aerospace Engineering Graduate Student Award
2009 William Peil Inventors Award

Teaching Experience:

Spring 2012

Aerodynamics (AEE 342)

Teaching Assistant: Syracuse University

- Helped students with homework and projects.
- Helped the professor with grading

Fall 2011

Flight Performance and Dynamics (AEE 427)

Teaching Assistant: Syracuse University

- Designed and ran labs for senior engineers using the Syracuse University flight simulator
- Lectured about the lab and what data will be calculated from the experiment

Fall 2009

Mechanical and Aerospace Engineering Laboratory (MAE 315)

Teaching Assistant: Syracuse University

- Helped Students run labs for the mechanical and aerospace junior lab class
- Was responsible for a digital signal processing lab

- Fall 06 - Spring 09 **Academic Excellence Workshop (ECS 105, ECS 201, and ECS 202)**
 Facilitator: Syracuse University
- Worked with small (5-8) groups of students to re-enforce concepts learned in Calculus I (MAT 295), Statics (ECS 221), and Dynamics (ECS 222)

Professional Activities:

2005 - Present	American Institute of Aeronautics and Astronautics	Student Member
2010 - Present	American Physical Society	Student Member

Publications:

Journal Articles:

To be Submitted:

- [1] Ruscher, C.J., Berger, Z.P., Shea, P.R., Hall, A.M., Tinney, C.E., Berry, M.G., Glauser, M.N., Gogineni, S., Kiel, B.V., “Comparison of High Speed Optical Measurements in a Jet Flow”, *Experiments in Fluids*.

Invited Talks:

- [1] Ruscher, C.J., Shea, P.R., Berger, Z.P., Berry, M.G., Dannenhoffer, J.F., Glauser, M.N., “Validation of Estimation Techniques using Time Resolved PIV on a High Speed Jet Flow”, Syracuse University MAE Seminar, 31 January 2014 Syracuse, NY
- [2] Shea, P.R., Glauser, M.N. Ruscher, C.J., Berger, Z.P., Dannenhoffer, J.F., Berry, M.G., Kiel, B.V., Gogineni, S., Noack, B.R., “Validation of Estimation Methods via Time Resolved PIV in High Speed Compressible Jets”, 52nd AIAA ASM, 13-17 January 2014 National Harbor, MD

Conference Proceedings (Peer Reviewed):

- [1] Ruscher, C.J., Dannehoff, J.F., Glauser, M.N., Belovich, V.M., and Sekar, B., “Modeling Flow Through a Perforated Plate for a New Combustion Rig”, 21st AIAA Computational Fluid Dynamics Conference., 24-27 June 2013, San Diego, CA (AIAA 2013-3092)
- [2] Ruscher, C.J., Dannenhoffer, J.F., and Glauser, M.N., “Application of Snapshot POD on a Varying Grid”, 51st AIAA ASM, 7-10 January 2013, Grapevine, TX (AIAA-2013-0256)
- [3] Ruscher, C.J., Dannehoff, J.F., Glauser, M.N., Kiel, B.V., and Sekar, B., “Vortex Shedding of Various Bluff Bodies in Cross Flow”, 48th AIAA Joint Propulsion Conference., July 29 August 1, 2012, Atlanta, GA (AIAA-2012-4285)

Conference Proceedings (Non-Refereed):

- [1] Ruscher, C.J., Dannehoff, J.F., and Glauser, M.N., "Data Fusion for Fluid Dynamic Data", 66th APS Division of Fluid Dynamics Mtg., 24-26 November 2013, Pittsburgh, PA
- [2] Ruscher, C.J., Dannehoff, J.F., Glauser, M.N., Belovich, Vincent M., and Sekar, Balu, "Characterization of the Boundary Conditions at the Test Section Inlet for a Combustion Rig", 65th APS Division of Fluid Dynamics Mtg., 18-20 November 2012, San Diego, CA
- [3] Ruscher, C.J., Dannehoff, J.F., and Glauser, M.N., "Snapshot POD for a Varying Grid", 2012 Thousand Island Fluid Dynamics Meeting, 27-29 May 2012, Gannanoque, Ontario, Canada.
- [4] Ruscher, C.J., Dannehoff, J.F., Glauser, M.N., Kiel, B.V., Sekar, B., Kapaku, R., "Vortex Shedding of Various Bluff Bodies in a Cross Flow and Flame Luminosity" 64th APS Division of Fluid Dynamics Mtg., 20-22 November 2011, Baltimore, MD
- [5] Ruscher, C.J., Dannehoff, J.F., Glauser, M.N., "Design of an Ejector for the AFRL HIT Rig," 2011 Thousand Island Fluid Dynamics Meeting, 15-17 April 2011, Gannanoque, Ontario, Canada
- [6] Shea, P.R., Ruscher, C.J., Wallace, R.D., Glauser, M.N., Dannehoff J.F., "Characterization of a Three-Dimensional Turret Wake for Active Flow Control Part II: Experimental Study," 63rd APS Division of Fluid Dynamics Mtg., 21-23 November 2010, Long Beach, CA.
- [7] Ruscher, C.J., Shea, P.R., Wallace, R.D., Glauser, M.N., Dannehoff, J.F., "Characterization of a Three-Dimensional Turret Wake for Active Flow Control Part I: Simulation," 63rd APS Division of Fluid Dynamics Mtg., 21-23 November 2010, Long Beach, CA.

Student Conferences:

- [1] Ruscher, C.J., "CFD Study of Flow over a Turret," AIAA Region I-NE Student Conference, 23-24 April 2010, Boston, MA.

Lab Experience:

2010 - Present Flight Simulator
 2009 - Present Subsonic Wind Tunnel

Computer Skills:

Operating Systems: Windows XP to Windows 7 Linux Ubuntu
Engineering Software: Pro Engineering Wildfire, Solid Works, Gambit, Fluent
 Engineering Equation Solver, OVERFLOW, SUGGAR
Data Processing: Matlab, Gfortran, Visual Basics Maple, LaTeX,
 Microsoft Office, Tecplot360, Fieldview

Undergraduate Team Project:

Spring 2009

Senior Design Project

- Worked in teams of three to build an aircraft
- Aircraft had to carry as much payload as possible across the carrier dome with one charge of the battery
- My team was able to score the most points

Spring 2006

CAD Model Clock Project

- Used Pro Engineering to design and build a clock
- Worked in teams of three to build the parts in CAD and assemble them final project

Fall 2005

Mars Rover Design Project

- Worked in groups to design, build, and program a Lego rover
- Rover had to pick up golf balls and deploy a rod on an obstacle course