DYNAMICAL MODELING OF THE FLOW OVER FLAPPING WING BY APPLYING PROPER ORTHOGONAL DECOMPOSITION AND SYSTEM IDENTIFICATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OĞUZ DURMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

SEPTEMBER 2011

Approval of the thesis:

DYNAMICAL MODELING OF THE FLOW OVER FLAPPING WING BY APPLYING PROPER ORTHOGONAL DECOMPOSITION AND SYSTEM IDENTIFICATION

submitted by **OĞUZ DURMAZ** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Ozan Tekinalp¸
Head of Department, **Aerospace Engineering** _______________

Assoc. Dr. Dilek Funda Kurtuluş
Supervisor, **Aerospace Engineering Department, METU** _______________

Assoc. Prof. Dr. Coşku Kasnakoğlu
Co-Supervisor, **Electrical & Electronics Engineering, TOBB ETU** _______________

Examining Committee Members:

Prof. Dr. Serkan Özgen
Aerospace Engineering, METU _________________________

Assoc. Prof. Dr. Dilek Funda Kurtuluş
Aerospace Engineering, METU _________________________

Asst. Prof. Dr. Oğuz Uzol
Aerospace Engineering, METU _________________________

Asst. Prof. Dr. Ali Türker Kutay
Aerospace Engineering, METU _________________________

Assoc. Prof. Dr. Coşku Kasnakoğlu
Electrical & Electronics Engineering, TOBB ETU _________________________

Date: 06.09.2011

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Oğuz DURMAZ

Signature :

ABSTRACT

DYNAMICAL MODELING OF THE FLOW OVER FLAPPING WING BY APPLYING PROPER ORTHOGONAL DECOMPOSITION AND SYSTEM IDENTIFICATION

Durmaz, Oğuz
M.Sc., Department of Aerospace Engineering
Supervisor: Assoc. Prof. Dilek Funda Kurtulus
Co-supervisor: Assoc. Prof. Coşku Kasnakoğlu

September 2011, 96 pages

In this study the dynamical modeling of the unsteady flow over a flapping wing is considered. The technique is based on collecting instantaneous velocity field data of the flow using Particle Image Velocimetry (PIV), applying image processing to these snapshots to locate the airfoil, filling the airfoil and its surface with proper velocity data, applying Proper Orthogonal Decomposition (POD) to these post-processed images to compute the POD modes and time coefficients, and finally fitting a discrete time state space dynamical model to the trajectories of the time coefficients using subspace system identification (N4SID). The procedure is applied using MATLAB for the data obtained from NACA 0012, SD 7003, elliptic airfoil and flat plate, and the results show that the dynamical model obtained can represent the flow dynamics with acceptable accuracy.

Keywords: Flapping wing, dynamical modeling, Proper Orthogonal Decomposition (POD), System Identification (SI), Particle Image Velocimetry (PIV)

ÖZ

UYGUN DİKGEN AYRIŞIMI VE SİSTEM TANILAMA YÖNTEMLERİ KULLANILARAK BİR ÇIRPAN KANAT ÜZERİNDEKİ AKIŞIN DİNAMİK MODELLEMESİ

Durmaz, Oğuz
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Dilek Funda Kurtuluş
Ortak Tez Yöneticisi: Doç. Dr. Coşku Kasnakoğlu

Eylül 2011, 96 sayfa

Bu tezde çırpan kanat çevresindeki değişken akışın dinamik modellemesi tamamlanmıştır. Bu yöntemde, öncelikle parçacık görüntülemeli akış ölçüm tekniği (PIV) kullanılarak anlık hız görüntüleri elde edilir. Daha sonra görüntü işleme tekniği kullanılarak anlık görüntülere kanat profili yerleştirilir. Bu işlenmiş anlık görüntülere uygun dikgen ayrışımı (POD) yöntemi uygulanarak POD modları ve zaman katsayıları bulunur. Son olarak alt işlem sistem tanılama (N4SID) yöntemi kullanılarak, daha önce bulunan zaman katsayılarının izdüşümleri kullanılarak ayrık zamanlı durum uzayı dinamik modellemesi yapılır. Görüntüsel akış ölçüm tekniği kullanılarak NACA 0012, SD 7003, eliptik ve düz plaka kanat profillerinden elde edilen veriler için MATLAB programı kullanılarak süreç tamamlanmış ve görülmüştür ki, elde edilen dinamik model, akışın dinamiğini kabul edilebilir bir doğrulukla göstermektedir.

Anahtar Kelimeler: Çırpan kanat, dinamik modelleme, Uygun Dikgen Ayrışımı (POD), Sistem Tanılama (SI), Görüntüsel Akış Ölçümü (PIV)

"To my parents"

ACKNOWLEDGEMENTS

I would like to thank to my advisor Assoc. Prof. Dr. Dilek Funda Kurtulus from METU and co-advisor Assoc. Prof. Dr. Coşku Kasnakoğlu from TOBB University for their guidance, advice, criticism, encouragements and insight throughout the research.

The technical assistance of Mr. Huseyin Deniz Karaca, Mr. Hasan Hızlı, Mr. Alpay Demircan and Miss. Gökçen Deniz Özen are gratefully acknowledged.

This study was supported by the Scientific & Technological Research Council of Turkey (TUBITAK) under project 109E233 and by the European Commission (EC) under project PIRG-2008-GA-239536.

TABLE OF CONTENTS

ABSTRACT iv
ÖZ v
ACKNOWLEDGEMENTS viii
TABLE OF CONTENTS ix
LIST OF FIGURES xi
LIST OF TABLES xv
CHAPTERS
1. INTRODUCTION 1
2. LITERATURE SURVEY 4
2.1 Flapping Wing Studies 4
2.2 Particle Image Velocimetry Studies 7
2.3 Proper Orthogonal Decomposition Studies 8
2.4 System Identification Studies 10
3. PARTICLE IMAGE VELOCIMETRY 12
3.1 Illumination 12
3.2 Seeding 13
3.3 Image Acquisition 13
3.4 Data Verification and Analysis 14
3.5 Pitching Motion 14
3.6 Experimental Procedure 18
4. IMAGE PROCESSING 20

5. PROPER ORTHOGONAL DECOMPOSITION 24
6. SYSTEM IDENTIFICATION 28
7. RESULTS 33
7.1 Results for NACA 0012 Airfoil 33
7.2 Results for SD 7003 Airfoil 41
7.3 Results for Elliptic Airfoil 49
7.4 Results for Flat Plate 56
8. CONCLUSION 64
REFERENCES 66
APPENDIX A 74
A.1.MATLAB Codes 74

LIST OF FIGURES

FIGURES

Figure 3.1: Schematic illustration of PIV setup. 15

Figure 3.2: Experimental PIV setup. 16

Figure 3.3: Schematic illustration of Traverse Mechanism 16

Figure 3.4: Instantaneous PIV Images of pitching NACA 0012 airfoil 18

Figure 3.5: Instantaneous PIV Images of pitching SD 7003 airfoil. 19

Figure 3.6: Instantaneous PIV Images of pitching elliptic airfoil. 19

Figure 3.7: Instantaneous PIV Images of pitching flat plate. 19

Figure 4.1: PIV Image processing 21

Figure 4.1.a: Grey scale original PIV image 21

Figure 4.1.b: Adaptive filter applied PIV image 21

Figure 4.1.c: Image after applying disk-shaped and linear structuring elements 21

Figure 4.1.d: Image after small elements are removed 21

Figure 6.1: The loop of System Identification 29

Figure 7.1: X Velocity Components of the PIV Snapshots during the second period for the pitching motion of NACA 0012 airfoil. 34

Figure 7.2: Y Velocity Components of PIV Snapshots during the second period for the pitching motion of NACA 0012 airfoil 34

Figure 7.3: POD Modes in X Direction for the pitching motion of NACA 0012 airfoil 35

Figure 7.4: POD Modes in Y Direction for the pitching motion of NACA 0012 airfoil 36

Figure 7.5: Energy Level vs Number of POD Modes for the pitching motion of NACA 0012 airfoil 37
Figure 7.6: X Velocities Reconstructed with POD Time Coefficients for the pitching motion of NACA 0012 airfoil 38
Figure 7.7: Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of NACA 0012 airfoil 38
Figure 7.8: Channel Output for the pitching motion of NACA 0012 airfoil 39
Figure 7.9: X Velocities Reconstructed with System Identification Time Coefficients for the pitching motion of NACA 0012 airfoil.. 40
Figure 7.10: Y Velocities Reconstructed with System Identification Time Coefficients for the pitching motion of NACA 0012 airfoil.. 41
Figure 7.11: X Velocity Components of the PIV Snapshots during the second period for the pitching motion of SD 7003 airfoil 42
Figure 7.12: Y Velocity Components of the PIV Snapshots during the second period for the pitching motion of SD 7003 airfoil 42
Figure 7.13: POD Modes in X Direction for the pitching motion of SD 7003 airfoil.... 43
Figure 7.14: POD Modes in Y Direction for the pitching motion of SD 7003 airfoil.... 44
Figure 7.15: Energy Level vs Number of POD Modes for the pitching motion of SD 7003 airfoil 45
Figure 7.16: X Velocities Reconstructed with POD Time Coefficients for the pitching motion of SD 7003 airfoil 46
Figure 7.17: Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of SD 7003 airfoil 46
Figure 7.18: Channel output for the pitching motion of SD 7003 airfoil 47
Figure 7.19: X Velocities Reconstructed with SI Time Coefficients for the pitching motion of SD 7003 airfoil 48
Figure 7.20: Y Velocities Reconstructed with SI Time Coefficients for the pitching motion of SD 7003 airfoil 48

Figure 7.21: X Velocity Components of the PIV Snapshots during the second period for the pitching motion of elliptic airfoil 49
Figure 7.22: Y Velocity Components of the PIV Snapshots during the second period for the pitching motion of elliptic airfoil 50
Figure 7.23: POD Modes in X Direction for the pitching motion of elliptic airfoil 51
Figure 7.24: POD Modes in Y Direction for the pitching motion of elliptic airfoil 51
Figure 7.25: Energy Level vs Number of POD Modes for the pitching motion of elliptic airfoil 52
Figure 7.26: X Velocities Reconstructed with POD Time Coefficients for the pitching motion of elliptic airfoil 53
Figure 7.27: Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of elliptic airfoil 53
Figure 7.28: Channel output for the pitching motion of elliptic airfoil 54
Figure 7.29: X Velocities Reconstructed with SI Time Coefficients for the pitching motion of elliptic airfoil 55
Figure 7.30: Y Velocities Reconstructed with SI Time Coefficients for the pitching motion of elliptic airfoil 55
Figure 7.31: X Velocity Components of the PIV Snapshots during the second period for the pitching motion of flat plate 56
Figure 7.32: Y Velocity Components of the PIV Snapshots during the second period for the pitching motion of flat plate 57
Figure 7.33: POD Modes in X Direction for the pitching motion of flat plate 58
Figure 7.34: POD Modes in Y Direction for the pitching motion of flat plate 58
Figure 7.35: Energy Level vs Number of POD Modes for the pitching motion of flat plate 59
Figure 7.36: X Velocities Reconstructed with POD Time Coefficients for the pitching motion of flat plate 60

Figure 7.37: Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of flat plate..60
Figure 7.38: Channel output for the pitching motion of flat plate.................................61
Figure 7.39: X Velocities Reconstructed with SI Time Coefficients for the pitching motion of flat plate..62
Figure 7.40: Y Velocities Reconstructed with SI Time Coefficients for the pitching motion of flat plate..62

LIST OF TABLES

TABLES

Table 3.1: Flow properties and PIV parameters 17

CHAPTER I

INTRODUCTION

In nature, flight with flapping wing is one of the most complicated types of transportation. Different kinds of insects and birds use flapping wings for flying [1]. Scientists and biologists were inspired from birds and insects for mechanical flight. They believed that, in order to create lift and propulsion, flapping wings were required. They have tried to describe the kinematics of flapping wing motion. According to the studies on birds and insects, they have found that, flapping wing motion is related with weight, wingspan, power requirements and flapping frequency [2].

Using flapping wings to achieve flight is a rich topic of active research which has recently received significant attention in literature, especially in the area of Micro Aerial Vehicles (MAVs). Using flapping wings for flight has important advantages over conventional rotary-driven aircraft such as higher maneuverability, increased efficiency, more lift, and reduced noise [3–4]. Understanding and modeling the aerodynamics of the flow of air over flapping wings is important since it can help improve these advantages. However, the aerodynamics of flapping motion is a complex nonlinear system because of the unsteady interaction between the vortex topologies and it is therefore difficult to model accurately [5].

As it is the case for any flow process, physical meaning of the flow over flapping wings can be understood and analyzed mathematically by describing the flow in terms of velocity vectors and representing it with dynamical models. Navier-Stokes partial differential equations (PDEs) are the most commonly used set of equations that are capable of representing fluid flows very accurately; however, due to their complexity they are very difficult to analyze and most of the time obtaining an analytical solution is not possible [6]. For this reason techniques such as Proper Orthogonal Decomposition (POD) are used to obtain a simpler representation of the velocity field and provide a convenient means to analyze the flow behaviour. POD is a technique to decompose a flow velocity field into spatial modes (POD modes) and time-dependent amplitudes (time coefficients). The POD method extracts deterministic functions associated with large-scale energetic structures in a flow [7].

Since the POD technique requires instantaneous velocity information (i.e. snapshots) to decompose the flow, Particle Image Velocimetry (PIV) measurement technique can be used to obtain data for the unsteady velocity field. Once the POD modes and time coefficients are obtained, techniques such as Galerkin Projection (GP) or System Identification (SI) can be used to derive a system of ODE’s to represent the dynamics of these time coefficients.

In this work, the dynamical modeling of the unsteady flow over a flapping wing is constructed. The modeling technique is based on collecting instantaneous velocity field data of the flow using Particle Image Velocimetry (PIV), applying image processing to these snapshots to locate the airfoil, filling the airfoil and its surface with proper velocity data, applying Proper Orthogonal Decomposition (POD) to these post-processed images to compute the POD modes and time coefficients, and finally fitting a discrete time state space dynamical model to the trajectories of the time coefficients using subspace system identification (N4SID). The procedure is implemented using MATLAB for the data obtained from four different types of airfoils namely, NACA 0012, SD 7003, elliptic and flat plate.

The present thesis consists of eight chapters. First chapter includes the introduction part. Second chapter includes the literature review of studies on flapping wings, particle image velocimetry (PIV), image processing, proper orthogonal decomposition, dynamical modeling and system identification (SI). In the third chapter, Particle Image Velocimetry technique is introduced. Experimental procedure of PIV is explained. In the fourth chapter, how to apply image processing on the original PIV image is explained. The fifth chapter includes details of proper orthogonal decomposition (POD) method. The sixth chapter is about how to construct a dynamical model by using System Identification. In the seventh chapter, the results of four different flapping wing studies and their dynamical model are given and discussed. The final chapter includes the conclusion of the present study.

CHAPTER II

LITERATURE SURVEY

The present chapter includes the literature review of studies on flapping wings, Particle Image Velocimetry (PIV), Proper Orthogonal Decomposition, dynamical modeling and System Identification (SI).

2.1 Flapping Wing Studies

Flapping wing studies started a century ago with the independent works of Knoller [8] and Betz [9]. They observed that flapping motion of the airfoil induces effective velocity. It causes an effective angle of attack during the motion therefore effective angle of attack creates an aerodynamic force on the airfoil. These aerodynamic forces can be decomposed into lift and drag components. The capability of the thrust generation of flapping wings is called Knoller-Betz effect [10]. The numerical studies on flapping airfoils started with Birnbaum [11]. He used linearized potential flow theory to investigate flapping airfoils numerically. The thrust production was first explained by von Kàrmàn and Burgers [12]. They related the vortex shedding from the trailing edge as a momentum deficit or surplus in the wake of the airfoil.

Pitching and plunging airfoil studies were started with the investigation of dynamic stall of the helicopter blades [13]. Using flapping wings to achieve flight is a topic of active research which has recently received significant attention in literature, especially in the area of Micro Aerial Vehicles (MAVs).

Using flapping wings for flight has important advantages over conventional rotary-driven aircraft such as higher maneuverability, increased efficiency, more lift, and reduced noise [3–4]. Efforts in this direction include the work by Kurtulus et al. [14], who used artificial neural networks (ANN) to model the input–output responses of this aerodynamics, and showed that these models successfully capture the most significant features of unsteady flapping motion. Deng et al. [15] developed a model which captures the main dynamical features of a Micromechanical Flying Insect (MFI) capable of sustaining self-governing flight through the use of linear estimation methods. Kurtulus et al. [16] constructed numerical models by using direct numerical simulations (DNS) and experimental models by using Particle Image Velocimetry (PIV) and Laser Sheet Visualizations in order to investigate the aerodynamics and vortex formation mechanism of the flow at different stages of unsteady flapping motion.

Zbikowski [17] introduced a conceptual structure for the aerodynamic modeling of an insect-like flapping wing in hover for MAVs and proposed two analytic approaches.

Fearing et al. [4] developed a thorax structure using four bar frames combined with an extensible fan-fold wing to provide adequate wing stroke and rotation. They considered the kinematic and power requirements for a micro-robotic flying device using beating wings, and presents an initial design of a thorax for the device.

Bai et al. [18] developed a new type of bionic flapping which is called the pitching down flapping. It is based on their aerodynamic mechanisms of pitching up flapping. Pitching down flapping reduce the time averaged power requirements when compared with pitching up flapping.

Hua et al. [19] investigated the time averaged aerodynamic performances of two flexible membrane wings in flapping flight. They tried a flexible nylon wing, a very flexible latex wing and a rigid wing. They found that, flapping wing provides aerodynamic benefits when the flapping flight is in unsteady state regime with advance ratio (i.e., the ratio of forward flight speed to wingtip velocity) of the flapping flight being smaller than one. Moreover, they found that, the rigid wing gave better lift production performance when compared with the flexible membrane wings for flapping flight. The latex wing, which is the most flexible wing when compared with other wings, is the best in thrust generation performance for flapping flight.

Mazaheri et al. [20] tried to find the effect of chordwise flexibility on the aerodynamics of flapping wings in hovering flight. They investigated how the twisting stiffness of wing affects generated thrust force and the power required at different flapping frequencies. Flapping wing system and experimental setup were designed in order to find the unsteady aerodynamic and inertial forces, power usage and angular speed of the flapping wing motion. They found that, the elastic deformation and inertial flapping forces affect the dynamical behavior of the wing.

Nguyen et al. [21] analyzed two dimensional aerodynamic models of flapping wing motions. They investigated different types of wing rotation and different positions of rotation axis in order to explain the force peak at the end of each half stroke. They found that, flapping mechanism with delayed rotation around quarter-chord axis increases efficiency and it can be used in robotic applications.

Amiralaei et al. [22] studied on the effects of unsteady parameters on the aerodynamics of a low Reynolds number pitching airfoil. They examined the influence of unsteady parameters (amplitude of oscillation, reduced frequency and Reynolds number) on the aerodynamic performance of the model. They found that, these parameters affected the aerodynamic performance (maximum lift coefficients, hysteresis loops, number of the generated vortices within the harmonic motion) of the system greatly.

Günaydınoğlu and Kurtulus [23-26] studied flapping kinematics, the effect of Reynolds number, reduced frequency and airfoil geometry on unsteady aerodynamics of a flapping airfoil in forward flight. They investigated pure plunge motion and pitch-plunge motion of the airfoil. They observed that, in the Reynolds number range of 10,000 to 60,000 dynamic stall of the airfoil is the primary mechanism of lift production and the airfoil geometry and reduced frequency affect the strength and duration of the leading edge vortex.

Tsai et al. [27] made design and aerodynamic analysis of a flapping wing micro air vehicle. They calculated the average lift force of the flapping wing in order to make a light flapping wing MAV. The flapping wing was analyzed at different frequencies and angle of attack and the pressure distribution was investigated.

Liang et al. [28] demonstated high-order accurate simulations of unsteady flow past plunging and pitching airfoil by using high-order spectral difference method. This method shows that, the vortex shedding pattern of an airfoil in an oscillating plunge motion becomes asymmetric at high frequency. This algorithm can be also used in high order accurate simulations of unsteady flow past flapping micro air vehicles.

In early studies, it was known that the pitching motions at high frequency can generate thrust on the airfoil. Sarkar et al. [29] tried to find the influence of various parameters on thrust generation from a harmonically pitching airfoil. They obtained the effect of location of the pitching axis on the propulsive characteristics of the airfoil.

Tuncer et al. [30] studied the optimization of the propulsive efficiency of a flapping airfoil by solving parallel Navier-Stokes equations. They found out that the propulsive efficiency of single flapping airfoil increases by preventing the formation of the large scale Leading Edge Vortices (LEVs).

2.2 Particle Image Velocimetry Studies

Since the POD technique requires instantaneous velocity information (i.e. snapshots) to decompose the flow, Particle Image Velocimetry (PIV) measurement technique can be used to obtain data for the unsteady velocity field. Details of the PIV technique like PIV algorithms, optical considerations, tracer particles, illuminating lasers, recording hardware, errors in PIV measurements and PIV vector processing can be found in [31,32]. Studies regarding PIV include Scarano et al. [33] who proposed an improved algorithm based on cross-correlation for the interrogation of PIV images by predicting the displacement of interrogation areas by means of an iterative procedure. Daichin et al. [34] investigated the influence of a water surface on the structure of the trailing wake of a NACA 0012 airfoil using PIV measurements of the flow at different ride heights between the airfoil and the surface. Works that are most significant and related to the direction in this thesis are probably the use of PIV measurements as the set of snapshots required to obtain the POD modes. These include Druault et al. [35] who reconstructed the 3D in-cylinder mean flow field from PIV data by using POD and Guibert et al. [36] who used POD for time interpolation from PIV data found in in-cylinder engine flow.

2.3 Proper Orthogonal Decomposition Studies

Proper Orthogonal Decomposition is a method of model reduction that attempts to find a basis set that maximizes the expectation of the energy. It is also known Karhunen–Loève decomposition or principal component analysis [6]. The proper orthogonal decomposition (POD) technique has gained wide popularity in recent years in applications related to data analysis and reduced-order modeling [37]. The basis can be obtained by first constructing the covariance matrix from an experimental measurement or a computer simulation. This approach is called "Method of Snapshots" which was presented by Sirovich [38]. After the basis set is obtained, the reduced order model can be found by the projection of the full model onto the eigenvectors which correspond to the largest eigenvalues. POD retains the greatest amount of kinetic energy for a specified number of basis vectors [39].

One can find many studies regarding fluid flows using POD methods including the work by Noack et al. [40], who studied low-dimensional models for the transient and post-transient cylinder wake and developed reduced flow methods by Galerkin

Projection. Model reduction for compressible isentropic flows using POD and Galerkin Projection was achieved by Rowley et al. [6].

Wilcox et al. [41] studied balanced model reduction via proper orthogonal decomposition method in order to provide accurate low-order representations by state-space systems. Tran et al. [42] applied proper orthogonal decomposition to the modeling and control of a complex flow process, namely the Rayleigh-Bénard convection phenomena. By using POD, reduced order modeling for unsteady transonic flows around an airfoil is completed by Bourguet et al. [43] where POD analysis was used to identify von Karman instability and buffeting which are two main unsteady phenomena induced by compressibility effects that describe the physics of flow.

Liberge et al. [44] studied reduced order modeling for flow around an oscillating cylinder. They selected proper orthogonal decomposition method because it is very efficient in the domain of fluid mechanics. Navier-Stokes equations were projected on the POD modes for the global velocity field and a non linear low order dynamical system was constructed by using a multiphase method similar to the fictitious domain method.

Reduced-order models are not only attractive for real-time control computation but also crucial for detailed stability and bifurcation analysis. Ravindran [45] designed the reduced order optimal controller for flow separation which is based on POD and Galerkin projection. POD was applied on the detailed finite element simulation results to extract the most energetic eigenmodes. Actuation was performed on a small part of the boundary with this reduced order model. It was obtained that the tangential blowing is more efficient in mitigating flow separation and reducing wake spread.

2.4 System Identification Studies

Once the POD modes and time coefficients are obtained, techniques such as Galerkin Projection (GP) or System Identification (SI) can be used to derive a system of ODE’s to represent the dynamics of these time coefficients. Kung [46] identified the most commonly known subspace method algorithm that a discrete-time state space model is calculated from a block Hankel matrix with Markov parameters. This theory relied on Markov parameters therefore it was difficult to construct a model for unstable systems.

Moonen et al. [47,48] described an alternative direct identification method for deterministic systems. This method also computed state space model from a block Henkel matrix from the input-output data. Their system matrices are computed from a set of linear equations that the state vector sequence is calculated as an interface between “past” and “future”.

Van Overschee et al. [49] computed the algorithm of data-driven identification schemes for pure stochastic identification. State sequence was obtained from the projection of input-output data in both of these studies. It is useful to predict the future because this projection includes all the information in the past. Thus, the state space matrices could be found from this state sequence.

There is a difference between classical identification and N4SID method that, in classical identification, first system matrices are obtained then by using Kalman filter, state sequences are calculated. However, in N4SID method, first state sequence is calculated then by using least squares, system matrices are constructed. [49]

Rowley et al. [6] obtained models for compressible isentropic flows by using Galerkin projection and Gerhard et al. [50] obtained low dimensional Galerkin models to control vortex shedding. As for using system identification, one can find studies such as the one by Khalil et al. [51] where the system matrices were obtained from POD modes using a least squares based linear method on frequency data. Perret et al. [52] proposed a new method to identify low order dynamical systems exploiting the fact that all such systems can be written in polynomial form and using statistical correlation between the projection coefficients onto the POD modes and their time derivatives.

Sjövall et al. [53] studied on subsystem state-space modes to analysis the dynamic behavior of built-up structures. They identified subsystem models by using adopting contemporary system identification methods. It was obtained that the order of non-trivial model determination was an important step in system identification process.

Van Overschee et al. [54] derived two new subspace state space system identification (N4SID) algorithms in order to identify mixed deterministic-stochastic systems. State sequences were determined by these algorithms through the projection of input and output data. These state sequences were shown to be outputs of non-steady state Kalman filter banks. Thus, state space system matrices were found.

This study contributes the literature that, dynamical modeling of the unsteady flow over a flapping wing is completed. Instantaneous velocity field data is collected from PIV measurements. Image processing is applied on the snapshots in order to locate the airfoil and to fill airfoil and its surface with proper velocity data. Then, POD method is used to compute POD modes and time coefficients. Finally, a discrete time state space dynamical model is fit to the trajectories of the time coefficients using subspace system identification (N4SID). The results show that the dynamical model obtained can represent the flow dynamics with acceptable accuracy.

CHAPTER III

PARTICLE IMAGE VELOCIMETRY

The measurement of the flow field properties such as the velocity and pressure fields is one of the most challenging and time consuming problems in experimental fluid mechanics. Local velocity can be measured by using Hot-Wire and Laser Doppler Velocimetry (LDV). However, the measurement of the entire flow field is very difficult by using these measurement techniques. These methods are very limited to measure an entire velocity field simultaneously [55]. Particle Image Velocimetry (PIV) is a non-intrusive flow measurement technique that allows observing the instantaneous velocity [16]. Two dimensional streamline patterns can be visualized by using PIV measurement technique in unsteady flows [32]. There are four requisites of PIV measurements, namely, illumination, seeding, image acquisition, data verification and analysis.

3.1 Illumination

In order to provide flow illumination, lasers are used. One of the most used laser in PIV measurements is an Nd:YAG laser. Nd:YAG (neodymium-doped yttrium aluminium garnet) is a type of crystal that is used for solid state lasers [56]. Nd:YAG lasers emit light with a wavelength 532 nm. Generally, there are two lasers inside of the box. Single Nd:YAG lasers can operate in the dual pulse mode that the duration of two pulse can be 0.1-200 µs. In dual pulse mode, the energy delivered is about %70 of the energy delivered in the single pulse mode. Losses in the harmonic generator cause the decrease of efficiency in dual pulse mode [55]. The laser power can be 120-200 mJ/pulse for Nd:YAG laser. For high-speed lasers, the power is nearly 30 mJ/pulse [57]. In this study, the power of the Nd:YAG laser is 120 mJ/pulse. The time separation is 10000 μs.

3.2 Seeding

One of the important components in PIV measurements is seeding. In the PIV system, different particle seeds in the flow field are used to reflect the beam of Nd:YAG laser, which flashes in discrete time with a definite frequency to the CCD camera. If the measurement is performed in airflow, fog and smoke generators, atomizing glycerin and water/oil mixtures can be used for seeding [57]. If it is performed in water, fluorescent dyes, hollow glass spheres and silver-coated hollow glass spheres can be used. In the present case silver coated hollow sphere particles with 10 μm average diameter are used for the visualization of the vector field.

The concentration of seeding is important because flow field regions may be left out due to poor seeding [55]. The concentration of the seeding inside the water tank is 4.75×10^{-5} g/cm^3. Silver coated hollow spheres can reflect the light better then hollow glass spheres. However, they are heavier than hollow glass spheres. Specific gravity of silver-coated hollow glass spheres is bigger than one. Therefore, gravity is effecting out of plane velocity but, the error is minimum since laser sheet is horizontal in this case. For reacting flows, TiO_2, Al_2O_3 particles are used. The characteristics of these particles are that they resist high temperature changes and do not melt easily. Moreover, the particle should follow the flow but it should not alter the flow [56].

3.3 Image Acquisition

A FlowSense 2M/E CCD camera is placed under the water tank to obtain instantaneous velocity field around the airfoil at a section. The light detected by the

CCD camera in PIV measurements is that which has been scattered 90 degree with respect to the laser light [36]. The focal length of the CCD camera is 60 mm and the frame rate is 5 Hz. Therefore, the system can take 50 double images in a period of motion (T=10 s). The resolution of the snapshots is 1600 x 1200 pixels. In order to collect velocity data, it is necessary to calibrate the system before the experiment to convert pixel to meters. The camera and Nd:YAG laser operate in synchronized fashion to capture the reflection from the particles.

3.4 Data Verification and Analysis

The captured double images are processed by Dantec Studio program employing adaptive cross correlation method. This is used to construct the velocity vectors by finding differences between sequential snapshots [58]. In order to extract the velocity information from the PIV images of the particles, interrogation analysis is needed. PIV images are analyzed by dividing the images into small interrogation regions [59]. Smaller interrogation window size is preferable because it can give better spatial resolution of the PIV measurements. However, if the windows size is too small, it can cause inaccurate velocity vectors. Interrogation is done by adaptive cross correlation method [60]. The resolution of interrogation area is 32 x 32 pixels. The output of the process is the velocity vector fields, i.e. the x and y velocities of the snapshots, which then imported into MATLAB.

3.5 Pitching Motion

The experimental setup includes a plexiglass water tank of 40 cm x 40 cm x 80 cm length. The schematic illustration of the PIV setup can be seen in Figure 3.1. A photo of PIV experimental setup can be seen in Figure 3.2. Inside the tank the pitching motion of a wing takes place, which is performed by a traverse system placed on top of the water tank. In the experiments, four different type of airfoils are used, namely, NACA 0012, SD 7003, elliptic airfoil and flat plate. The center of rotation is located at center of pressure, which is the quarter chord of the airfoil from the leading edge

[16]. Two-step motors constitute traverse system. The first motor allows the translational motion and the second one is associated with the rotation of the wing. The useful rotational motion is of 360°. The pitching motion is performed in zero free-stream velocity [16]. The pitching motion of the airfoil is provided by a motor, which is mounted on the traverse system. The schematic illustration of the traverse mechanism can be seen in Figure 3.3

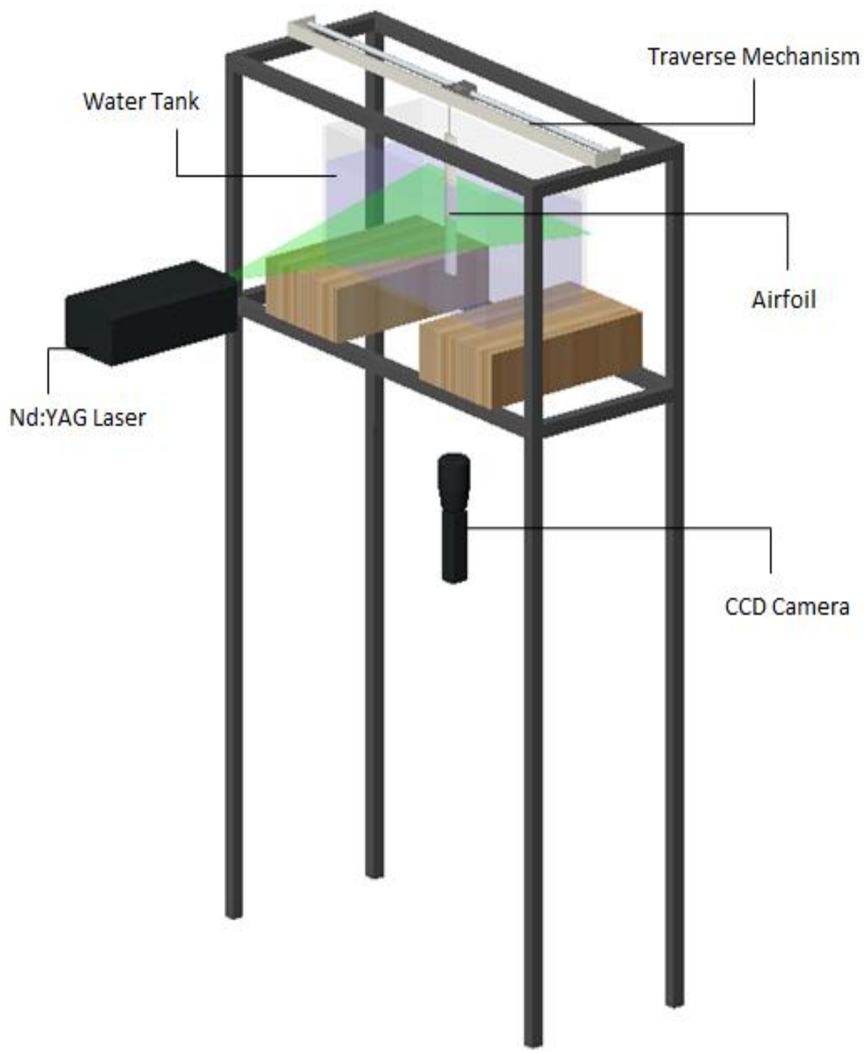


Figure 3.1 Schematic illustration of PIV setup

Figure 3.2 PIV Experimental Setup

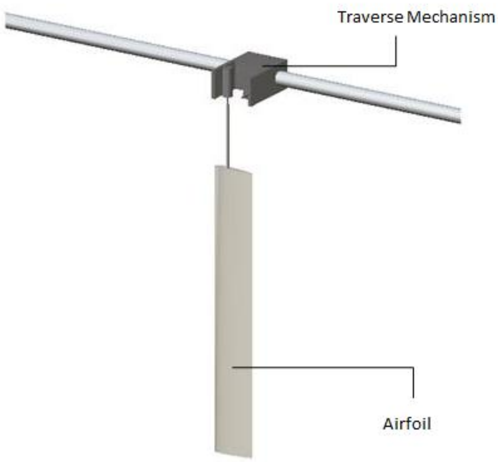


Figure 3.3 Schematic illustration of Traverse Mechanism

Period of the pitching motion is 10 s and maximum angular velocity is 0.63 rad/s. The motor is connected to the quarter chord of the airfoil; therefore, angular displacement of quarter chord is zero. The signals provide the movement of the motors, which are generated by a MATLAB/Simulink program. Flow properties and PIV parameters are tabulated in Table 3.1.

Table 3.1 Flow properties and PIV parameters

Airfoil	Type	NACA 0012, SD 7003, Elliptic and Flat plate
	Chord	c = 0.06 m
Flow	Fluid	Water
	Temperature [°C]	21
Motion	Period [s]	10
	Max. Angular Velocity [rad/s]	0.63
	Equation of Angular Velocity	$2\pi f \alpha_0 \cos(2\pi f t)$
Seeding	Type	Silver Coated Hollow Glass Spheres
	Diameter [µm]	10
	Concentration [g/cm^3]	0.0000475
Laser	Type	Nd:YAG
Recording	Camera Type	CCD FlowSense 2M/E
	Number of Camera	1
	Lens Focal Length [mm]	60
	Frame Rate [Hz]	5
	$\Delta t/T$	0.02
	Resolution [pixels]	1600 x 1200
	Exposure Delay Time [µs]	10000
Interrogation	Method	Double Frame & Adaptive Cross Correlation
	Resolution	32 x 32 pixels with 50% overlap

3.6 Experimental Procedure

PIV measurements are very sensitive, so they should be done very carefully. A successful experiment depends on lots of parameters. First, the environment of experimental setup should be dark because the light of laser should be seen effectively. Therefore, CCD camera takes clear snapshots. In addition, when the laser emits green light at that time, to avoid reflections from the back of the water tank, black paperboards are put around the water tank. These paperboards have been successful in preventing reflections caused by the glass of the water tank. Before the experiment, the goal is to get snapshots 4 times wider than the chord of the airfoil. For this reason, the distance between the experimental setup and the camera was chosen to be as high as possible. Thick woods were put under the experimental setup therefore; the camera's field of view was increased. Moreover, in order to get exact snapshots, CCD camera is put perpendicular to the floor of the tank. Another important issue to consider before the experiment is that the water tank must be stagnant. Otherwise, wrong movements and vectors could be observed. In order to avoid this, experiments is done after 5 minutes to fix airfoil to traverse mechanism. This process is repeated for each case. In the first experiment, the images were quite blurred. The reason is that, the density of silver-coated hollow sphere particles was too high. Hence, the experiment was done immediately after the particles were put into the water tank. At that time the snapshots seemed to be clear. The images can be seen in Figure 3.4, 3.5, 3.6 and 3.7.

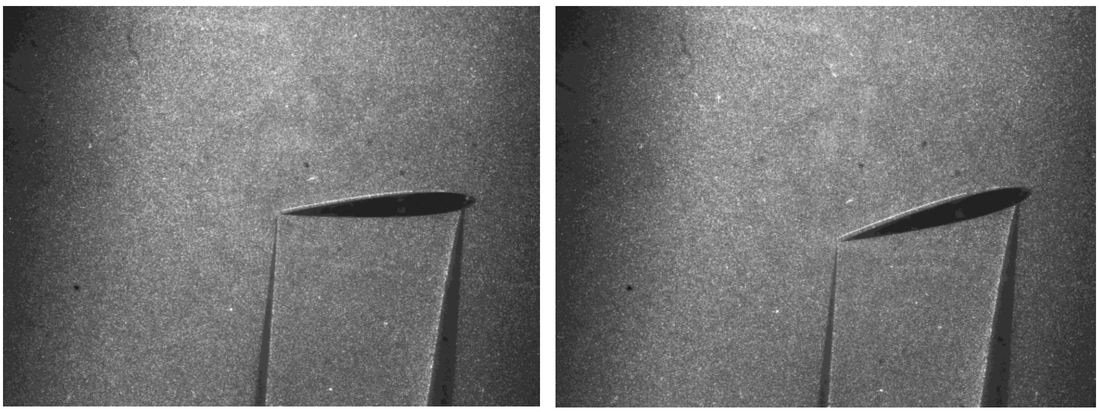


Figure 3.4 Instantaneous PIV Images of pitching NACA 0012 airfoil

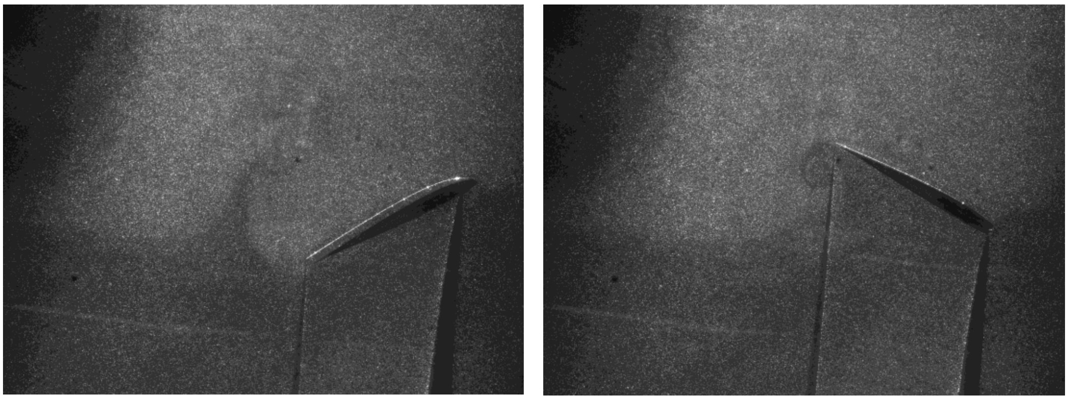


Figure 3.5 Instantaneous PIV Images of pitching SD 7003 airfoil

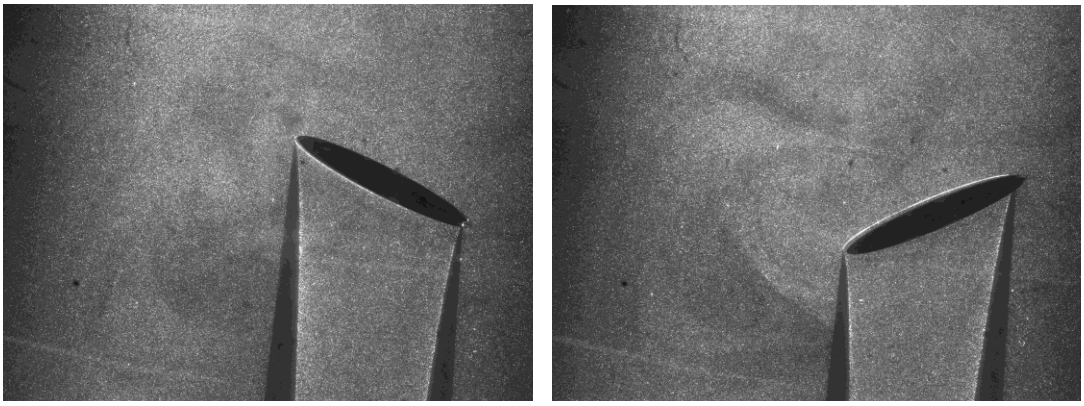


Figure 3.6 Instantaneous PIV Images of pitching elliptic airfoil

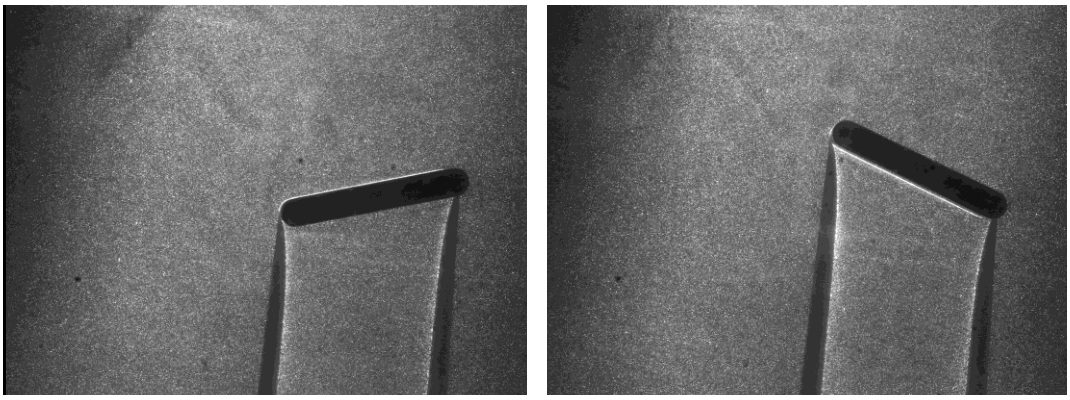


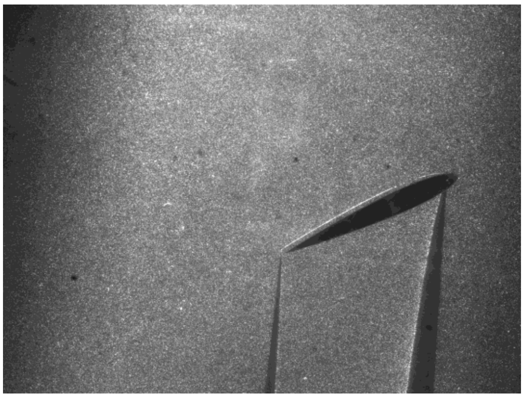
Figure 3.7 Instantaneous PIV Images of pitching flat plate

CHAPTER IV

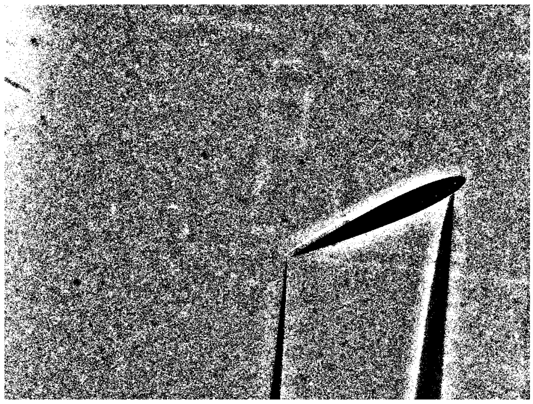
IMAGE PROCESSING

To use the PIV data for POD and SI, a preprocessing of the snapshots is needed in order to determine the geometry of the airfoil; this must be done at each time instant since the airfoil is in motion. For this purpose, an image-processing algorithm in MATLAB was developed for masking the airfoil.

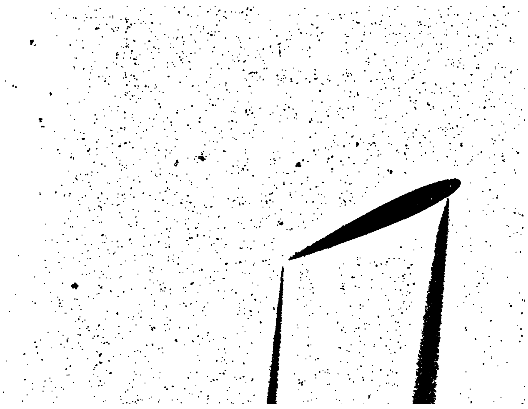
By using MATLAB Image Processing Toolbox, techniques like image enhancement, noise elimination, morphological operations and edge detection was applied in proper sequence to each PIV snapshot, to determine the geometric properties of the airfoil such as the leading edge, trailing edge, quarter chord location, coordinates of its surface and angle of attack. Original PIV images are grey scale. One of these images can be seen in Figure 4.1.a.



a) Grey scale original PIV image



b) Adaptive filter applied PIV image



c) Image after applying disk-shaped and linear structuring elements

d) Image after small noises are removed

Figure 4.1 PIV Image processing

The first application on the original images is to use adaptive filter, which takes the average of grey scale image. If intensity of the pixel is bigger than the average threshold value, this filter changes the colour of this pixel black. If intensity of the pixel is less than the average threshold value, at this time adaptive filter changes the colour of this pixel white. The adaptive filter applied on the original PIV image can be seen in Figure 4.1.b.

After applying the adaptive filter, a disk-shaped structuring element is created on the image within a radius of 2 pixels. This technique is called “radial decomposition using periodic lines” [61,62]. Moreover, the linear structuring element is created and applied on the image. By using these filters small noises on the image can be eliminated. The image after applying these structuring elements can be seen in Figure 4.1. c.

After applying second filter, the connected components in the image, which are fewer than 25000 pixels, are removed. “bwareaopen” command is used for this application which is a function of MATLAB. First, the connected components are determined. Then the area of each component is computed and finally small objects are removed. (Figure 4.1.d)

After eliminating all noises inside the image, the image is ready for determining the geometric properties of the airfoil such as the leading edge, trailing edge, quarter chord location, coordinates of its surface and angle of attack. To obtain them, “bwperm” command is used on MATLAB.

Once the geometry of the airfoil is obtained by the image processing algorithm described above, velocity information is added inside the airfoil and its surface. This is necessary because in the flow field the streamlines cannot penetrate the airfoil since it is a solid body. In addition, the POD method is easier to use if the velocity information is available for all points within the snapshots. The airfoil performs a sinusoidal pitching motion, where the instantaneous angle of attack α in radians is given by;

$$\alpha(t) = \alpha_0 sin\,(2\pi f t) \tag{4.1}$$

where $\alpha_0 = \pi/6$, $f = 0.1\,Hz$ is the frequency of the motion, and t is the time variable. The angular velocity ω in rad/s with respect to the quarter chord of the airfoil is given by the expression

$$\omega(t) = \frac{d\alpha(t)}{dt} = 2\pi f \alpha_0\, cos(2\pi f t) \tag{4.2}$$

Since the x-y coordinates of the airfoil's surface and the angular velocity ω is known for each snapshot, the velocity vectors $\boldsymbol{U}(x,y)$ inside the moving airfoil and on its surface can be calculated by

$$\boldsymbol{U}(x,y) = \boldsymbol{r}(x,y) \times \omega\boldsymbol{k} \tag{4.3}$$

Where $\boldsymbol{r}$ is the position vector relative to the quarter chord and $\boldsymbol{k}$ is the unit vector in z-direction.

CHAPTER V

PROPER ORTHOGONAL DECOMPOSITION

Proper Orthogonal Decomposition is an effective method for data analysis that, high-dimensional processes can be described by low-dimensional approximations by using this method. Since the high order models are too complex to be used in analysis, simulation or design, model reduction is needed. During model reduction, it is important that the properties of the original model should be preserved such as stability and passivity. POD often provides a reduced order model that is also stable [63]. A flow velocity field can be decomposed into spatial modes and time-dependent amplitudes by using POD; this method is also known as *the Karhunen-Loeve Decomposition* [7]. In order to identify the characteristics of experimental and numerical data, POD has proven to be an effective method. The characteristics of the physical process can be found in the large quantities of data by using POD [60].

When POD is applied on a flow, the energy of the flow is investigated. It means that, the mean square of the fluctuating value of the flow variable is investigated. The result of this application is a set of POD modes. These POD modes represent an average spatial description of structures which includes most of the energy and which are dominant in large-scale structure [60].

There are some limitations of the application of POD although it is useful to extract the dominant flow structures. If the flow is less deterministic, POD expansion will represent the flow less efficiently. For example, a low Reynolds number, laminar flow can be represented by ten POD modes. However, a high Reynolds number and a turbulent flow may be required several hundred POD modes to capture the kinetic energy of the flow [60].

To apply Proper Orthogonal Decomposition, first some snapshots of the flow are collected from an experimental or a numerical system. For our case, unsteady velocity measurements around a NACA 0012 airfoil are collected by using PIV technique as described in the previous section. After collecting the data, POD technique produces a set of basis functions, which optimally represent the spatial distribution of the snapshots collected. Each POD basis function captures a certain percentage of the energy of the flow field [59,64]. One can choose a certain number (N) of POD modes to capture a sufficient amount of the flow energy and then represent the flow field as a finite dimensional approximation as [7]

$$U(x,y,t) \approx \sum_{i=1}^{N} a_i(t)\phi_i(x,y) \tag{5.1}$$

where a_i, $i = 1,2,\ldots,N$ are called *time coefficients* and ϕ_i are the *POD modes* of the ensemble. To obtain the POD modes ϕ_i of the flow field, one first obtains a set of instantaneous velocities (snapshots) U_i where $U_i(x,y) = U(x,y,t_i)$ and t_i is the i^{th} time instant where measurements are taken. The average of the ensemble of snapshots can then be defined as

$$\bar{U}(x,y) = \langle U \rangle = \frac{1}{M}\sum_{i=1}^{M} U_i(\,x,y\,) \tag{5.2}$$

where M is the number of snapshots. Then a new set of snapshots (V_i) are obtained by subtracting this mean value from each velocity measurement

$$V_i(x,y) = U_i(x,y) - \bar{U}(x,y) \tag{5.3}$$

Next, the $M \times M$ spatial correlation matrix C of the ensemble is constructed as follows

$$C_{ij} = \frac{1}{M} \int \int V_i(x,y) V_j(x,y) dx\, dy\,, \qquad i = 1, \dots, M, \qquad j = 1, \dots, M \tag{5.4}$$

where the 2D integration is carried out over the entire flow domain. The POD modes ϕ_i of the flow field can be obtained by solving the eigenvalue equation

$$C\phi_i = \lambda_i \phi_i \tag{5.5}$$

That is, the POD modes ϕ_i are the eigenvectors of the correlation matrix C. The eigenvalue λ_i represent the amount of energy of the flow field captured by the ith POD mode ϕ_i. Based on this energy information, one can decide on the number of POD modes (N) to include in the approximation Eq. (5.1). It can be shown that the POD modes obtained using this procedure are orthonormal, i.e. they satisfy

$$\frac{1}{M} \int \int \phi_i(x,y) \phi_j(x,y) dx\, dy = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{5.6}$$

Once the POD modes are obtained, the time coefficients a_i of the flow field can be obtained by projecting the snapshots onto the POD modes

$$a_i(t_j) = \frac{1}{M} \int \int V(x,y,t_j) \phi_i(x,y) dx\, dy\,, \qquad i = 1, \dots, N, \qquad j = 1, \dots, M \tag{5.7}$$

The details of the POD procedure can be found in Sirovich [38] and Holmes et al. [65].

In this work, the Proper Orthogonal Decomposition (POD) method just described is applied to the post-processed PIV snapshots of the pitching airfoil in order to obtain an expansion of the form given in Eq. (5.1). For this purpose PIV data for 5 periods of the pitching motion is used, which consists of 250 snapshots. After the POD modes of unsteady velocity measurements are obtained, time coefficients are calculated by taking projection of snapshots on the POD modes as in Eq. (5.7)

CHAPTER VI

SYSTEM IDENTIFICATION

Once the POD modes are obtained and the flow is expanded as in Eq. (5.1), it can be observed that the time variation is dictated only by the coefficients a_i because the modes Φ_i depend only on the spatial variables and not the time variable [7]. Hence to model the flow dynamics it is sufficient to fit a proper model to the trajectories of the time coefficients $a_i(t)$. For this purpose, a state space model should be derived [7].

System identification can be defined as a technique to identify a dynamic model by using a data set, which is measured, by some mathematical tools and algorithms. In the system identification procedure, there are three steps to construct a model. Experimental data is collected from PIV measurement technique. The input signals can be selected from this data set. Then, a set of candidate model is selected which type of model can be applicable for the process. This is the most critical point in the system identification procedure. During the model selection, not only the model should fit the data, but also it should reflect the physical considerations. Otherwise, model with some unknown parameters cannot be constructed from basic physical laws. By the guidance of the data, the best model in the set is obtained. This is called identification method. The quality of models is evaluated by the comparison between the performances of models when the measured data is attempted to reproduce.

After these steps are done, the model validation is applied in order to test the model whether the model is valid for its purpose or not. Model validation evaluates the relationship between the model and observed data, prior knowledge and its intended use. If the model is not good enough, it can be rejected by the result of model validation. If the model does not pass the model validation test, steps of system identification procedure should be revised. Selected model may have deficiencies for some reasons. The numerical procedure may be failed to obtain the best model with reference to criterion. The criterion might be chosen badly. The model set might not good enough to define the system. The data set might not contain enough information to select a good model. The loop of system identification can be seen in Figure 6.1.

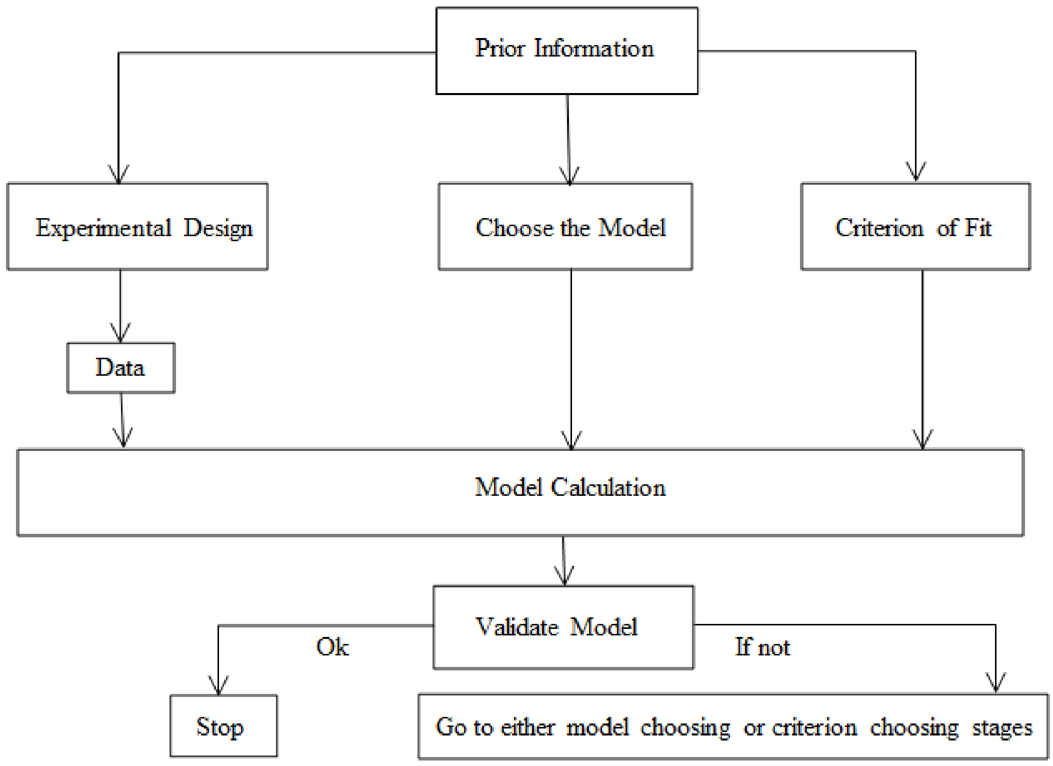


Figure 6.1 The loop of System Identification [67]

State space model can be described by the relationship between the input, noise and output signals which is written as a system of first-order differential equations or difference equations by using a state vector. It can be combined into a first order vector matrix difference equation or a differential equation [66]. The mathematical representation of the systems of equations can be simplified by using the vector matrix notation. After Kalman's work on prediction, state space models became a popular approach to describe linear dynamical systems [67].

The state space concept provides to design control systems with respect to given performance indexes. Design in the state space can be performed for a class of inputs, instead of a specific input function such as the impulse function, step function, or sinusoidal function. Moreover, state space methods provide to include initial conditions in the design. In conventional design methods, this feature is not possible [66].

In state space analysis, three types of variables are involved in the modeling of dynamic systems, namely, input variables, output variables and state variables. The state variables of a dynamic system are the variables making up the smallest set of variables that determine the state of the dynamic system. These variables need not be physically measurable or observable quantities. They do not represent physical quantities therefore choosing state variables is independent. This is an advantage of the state space method [66].

A state-space model can be derived as follows [7]:

$$\xi(t + T_s) = A\xi(t) + B\gamma(t) \tag{6.1}$$

$$y(t) = C\xi(t) + D\gamma(t) \tag{6.2}$$

where $\xi \in \mathbb{R}^n$ is the state vector, $n \in \mathbb{N}$ is the degree of the system, $\gamma \in \mathbb{R}$ is control input and $\gamma \in \mathbb{R}^n$ is the output signal. Since the flow snapshots are obtained at discrete time intervals separated by a sampling period of $T_s \in \mathbb{R}$ seconds, the model above is a discrete-time state space model. The matrices A, B, C and D determine the dynamical system and are to be obtained by constructing a model from Eq. (6.1) and Eq. (6.2) using system identification techniques [7]. For this purpose, input-output data must be collected for the system, which is usually done by applying different input signals (e.g. triangle waves, square waves) and recording final outputs, which are the time coefficients of the expansion, i.e.

$$y(t) = a(t) = [a_1(t)\ a_2(t)\ \dots\ a_N(t)]^T \tag{6.3}$$

where N is the number of POD modes used in the expansion. Subspace system identification method can compute the A, B, C, and D matrices in Eq. (6.1) and Eq. (6.2) by using the input-output data. The extended observability matrix is estimated from:

$$Q_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} \tag{6.4}$$

This matrix used for the system from input–output data by direct least-squares-like projection steps [7]. Especially, it is possible to show that an expression of the form [7]:

$$Y_r(t_k) = Q_r\,\xi(t_k) + S_r\Gamma_r(t_k) + V(t) \tag{6.5}$$

can be obtained from Eq. (6.1) and Eq. (6.2), where

$$Y_r(t_k) = \begin{bmatrix} Y(t_k) \\ Y(t_{k+1}) \\ \vdots \\ Y(t_{k+r-1}) \end{bmatrix} \tag{6.6}$$

$$\Gamma_r(t_k) = \begin{bmatrix} \gamma(t_k) \\ \gamma(t_{k+1}) \\ \vdots \\ \gamma(t_{k+r-1}) \end{bmatrix} \tag{6.7}$$

$$S_r = \begin{bmatrix} D & 0 & \cdots & 0 & 0 \\ CB & D & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{r-2}B & CA^{r-3}B & \cdots & CB & D \end{bmatrix} \tag{6.8}$$

and $V(t)$ occurs due to noise. The extended observability matrix Q_r can be calculated from Eq. (6.5) by correlating both sides of the equality with quantities that eliminate the term $S_r\Gamma(t_k)$ and make the noise influence from $V(t)$ disappear asymptotically [7]. Once the extended observability matrix is found, A and C matrices can be determined by using the first block row of Q_r and the shift property, respectively. Then, B and D can be calculated by using linear least squares on the following expression [7]:

$$y(t_k) = C(zI - A)^{-1}B\gamma(t_k) + D\gamma(t_k) \tag{6.9}$$

where Eq. (6.9) is a representation of the system described by Eq. (6.1) and Eq. (6.2) in terms of the time-shift operator z [7]. This last step may be skipped in case of an autonomous model (since B and D will be absent), which is the case in this study. One can find additional information about system identification in Ljung [67], Van Overschee [68] and Larimore [69].

In this work, MATLAB System Identification Toolbox is used to construct a dynamical system for the time coefficients, which were obtained in the previous step by projecting the snapshots on the POD modes as in Eq. (5.7).

CHAPTER VII

RESULTS

In this chapter, the results for a dynamical model of the pitching motion of the NACA 0012, SD 7003, elliptic airfoils and flat plate are presented.

7.1 Results for NACA 0012 Airfoil

The first task is to complete experiments and collect velocity data using the PIV setup as described in Chapter 3. Two of the images of NACA 0012 airfoil obtained from the PIV measurements are shown in Figure 3.4. The laser illuminated seeding particles can be seen clearly which appear white in the figure, and the airfoil in motion, which seems black. Then, velocity vectors are obtained from the PIV images for 5 periods using Dantec Dynamic Studio program. Velocity information is extracted from double images by using Adaptive Cross Correlation method. The resolution of interrogation area is 32x32 pixels with %50 overlap.

Next, the instantaneous location of the airfoil is determined using the image processing techniques described in Chapter 4. Figure 7.1 and Figure 7.2 show the instantaneous velocity field over NACA 0012 airfoil in the X and Y directions with the appropriate airfoil location superimposed on the image. The images were taken from the second period of motion.

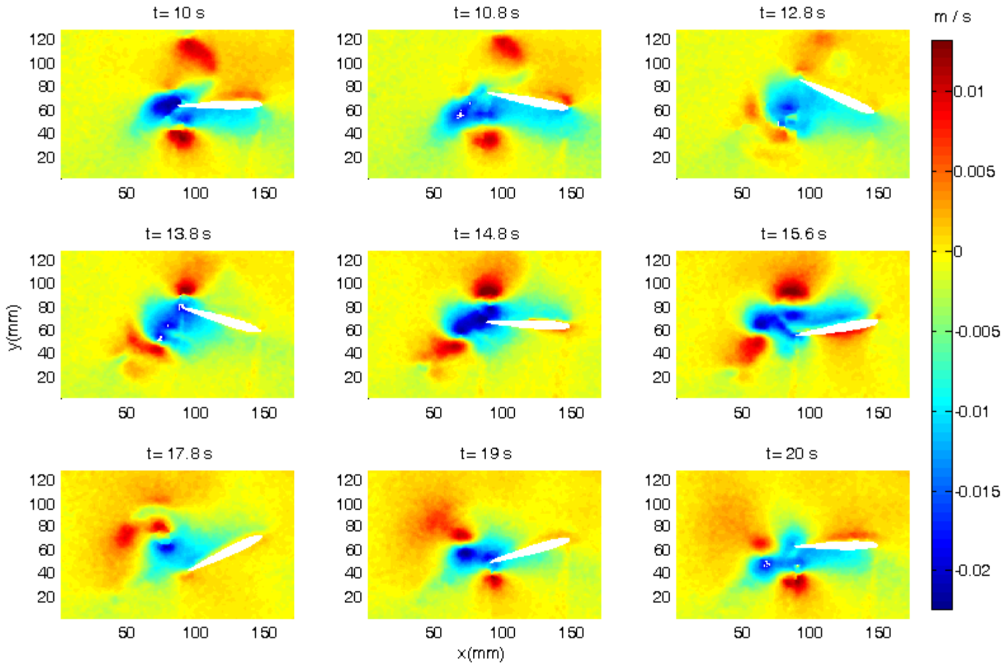


Figure 7.1 X Velocity Components of the PIV Snapshots during the second period for the pitching motion of NACA 0012 airfoil

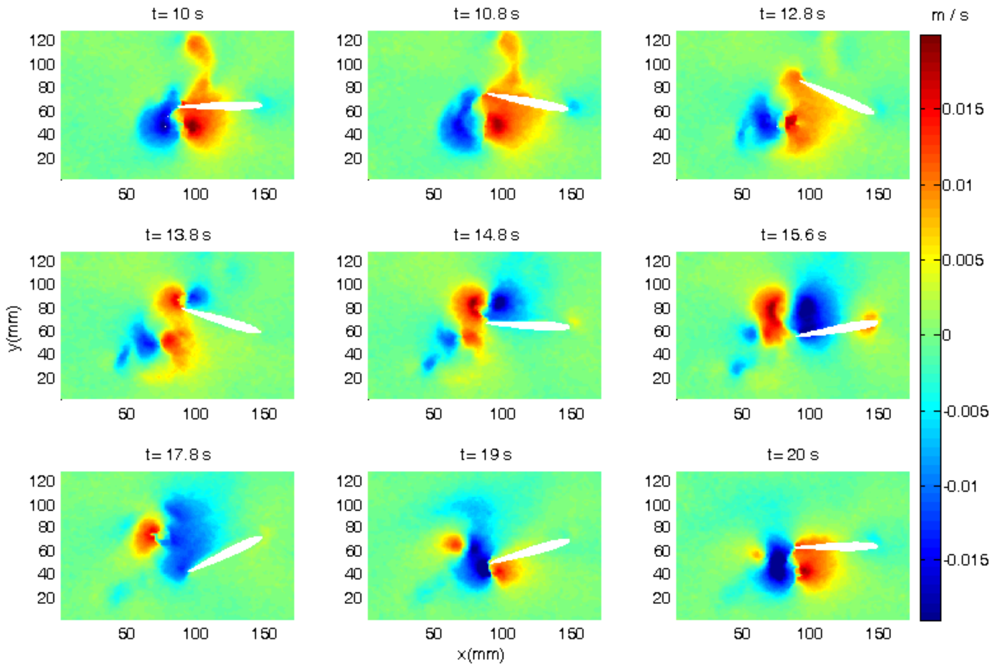


Figure 7.2 Y Velocity Components of PIV Snapshots during the second period for the pitching motion of NACA 0012 airfoil

In order to apply Proper Orthogonal Decomposition method to these snapshots, the velocity information inside the airfoil and its surface is filled in using Eq. (4.3) and the mean value of the flow is removed. The X and Y components of the first nine POD modes resulting from applying POD to the flow snapshots can be seen in Figure 7.3 and Figure 7.4.

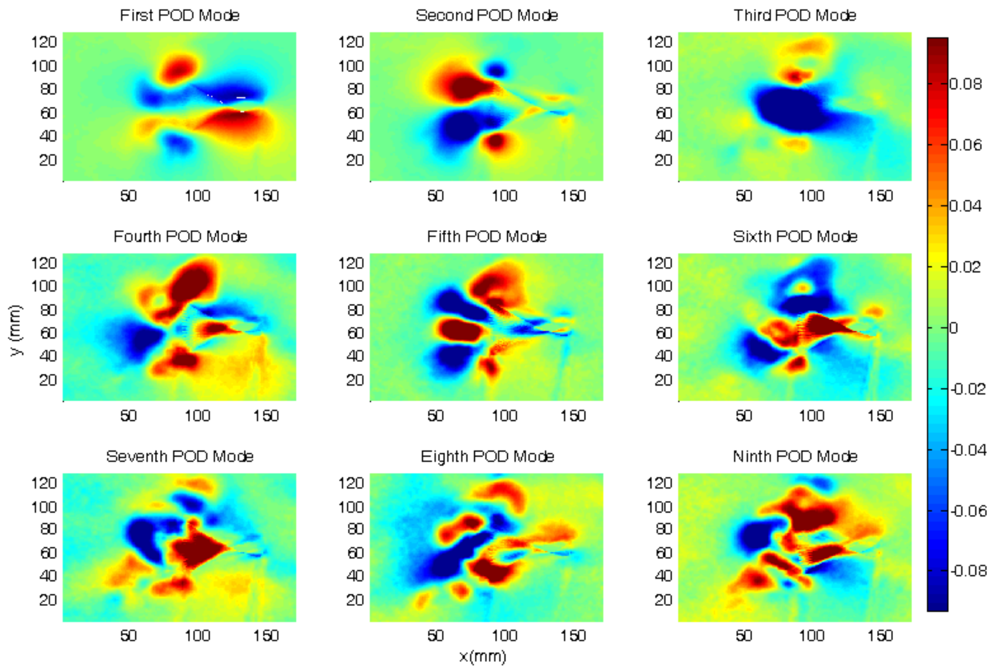


Figure 7.3 POD Modes in X Direction for the pitching motion of NACA 0012 airfoil

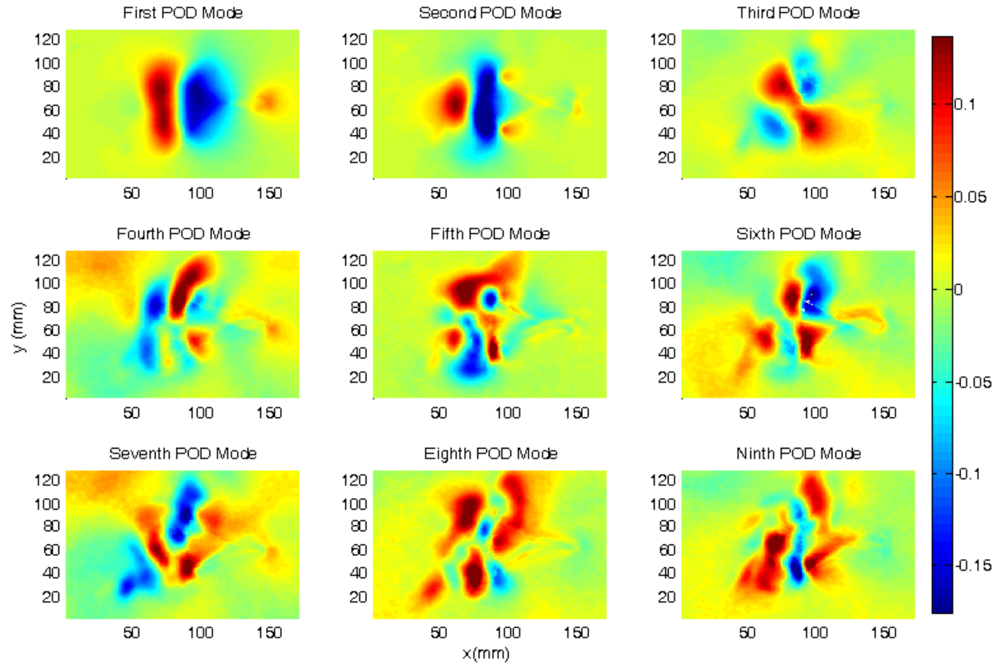


Figure 7.4 POD Modes in Y Direction for the pitching motion of NACA 0012 airfoil

In this work, 100 modes out of 250 modes have been chosen, which correspond roughly to 98.93% of the flow energy. Energy level vs. number of modes can be seen in Figure 7.5.

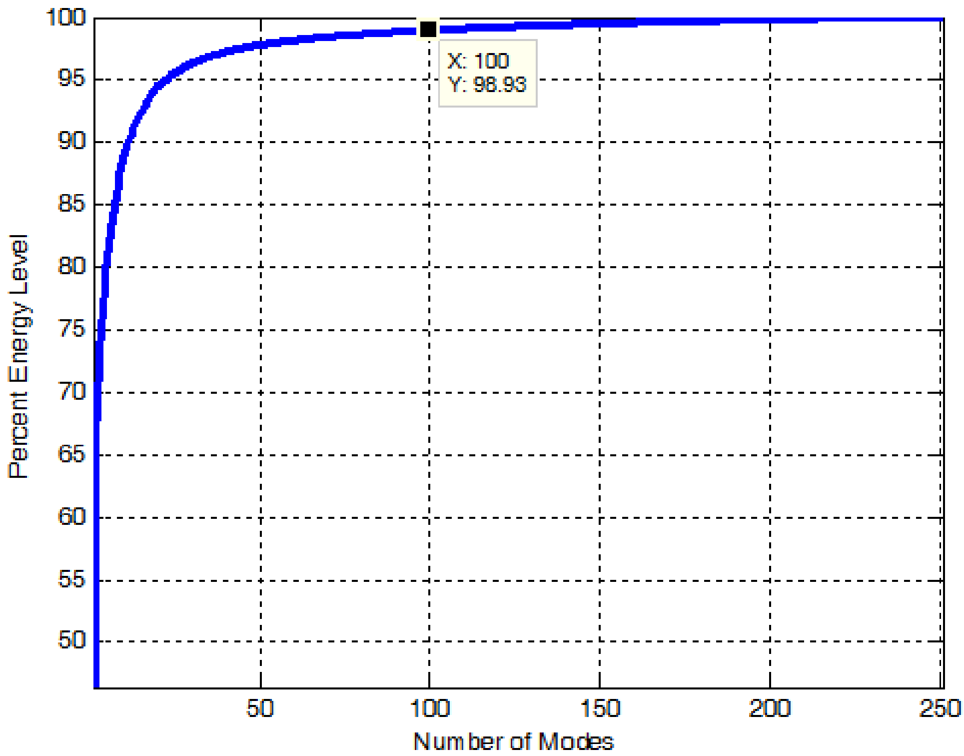


Figure 7.5 Energy Level vs Number of POD Modes for the pitching motion of NACA 0012 airfoil

In order to verify that the selected number of modes can represent the flow accurately, reconstruction of the flow is obtained by using Eq. (5.1), where the time coefficients a_i are obtained by projecting the flow snapshots onto the POD modes as expressed in Eq. (5.7). The X and Y components of the reconstructed flow velocity are shown in Figure 7.6 and Figure 7.7, respectively.

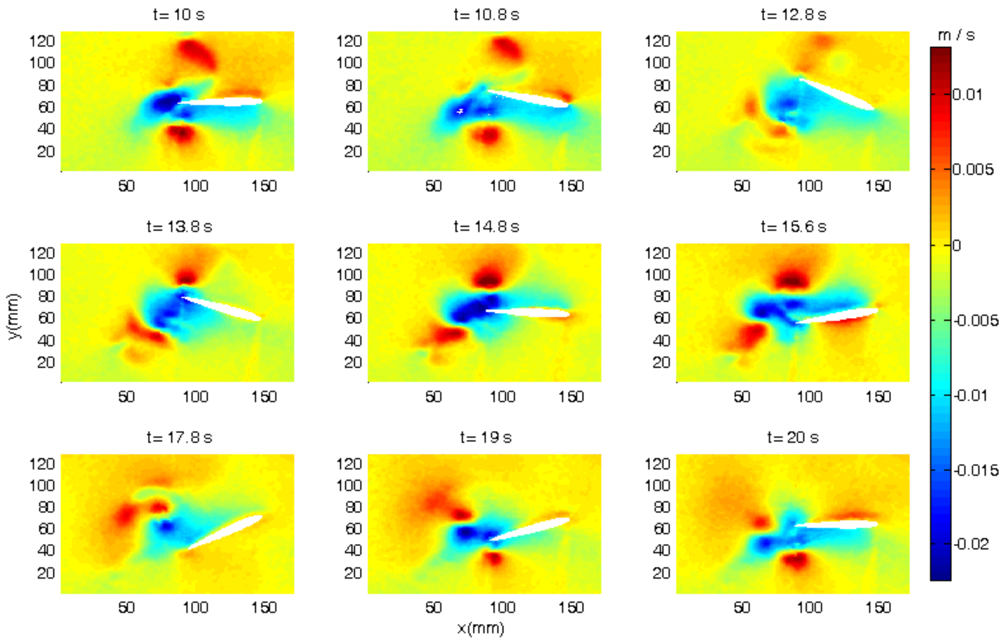


Figure 7.6 X Velocities Reconstructed with POD Time Coefficients for the pitching motion of NACA 0012 airfoil

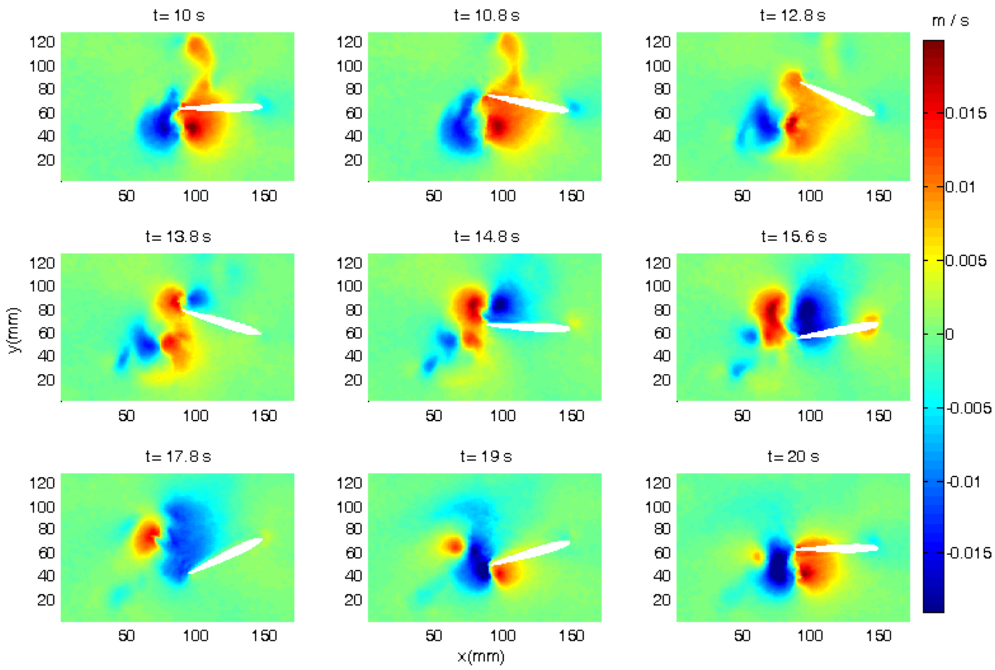


Figure 7.7 Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of NACA 0012 airfoil

It can be observed that the reconstructed flow velocity is quite similar to snapshots obtained from PIV shown in Figure 7.1 and Figure 7.2. The next step is to fit a dynamical model to the time coefficient data using N4SID system identification technique. This procedure is carried out using MATLAB System Identification Toolbox and a discrete time state space dynamical model of the form Eq. (6.1) and Eq. (6.2) is obtained [7]. The order of this model turned out to be 40.

In order to evaluate how well this model approximates the POD time coefficients, this model was run for the length of five periods (50 seconds) and compare the output of the model with the POD time coefficients. This comparison is presented in Figure 7.8.

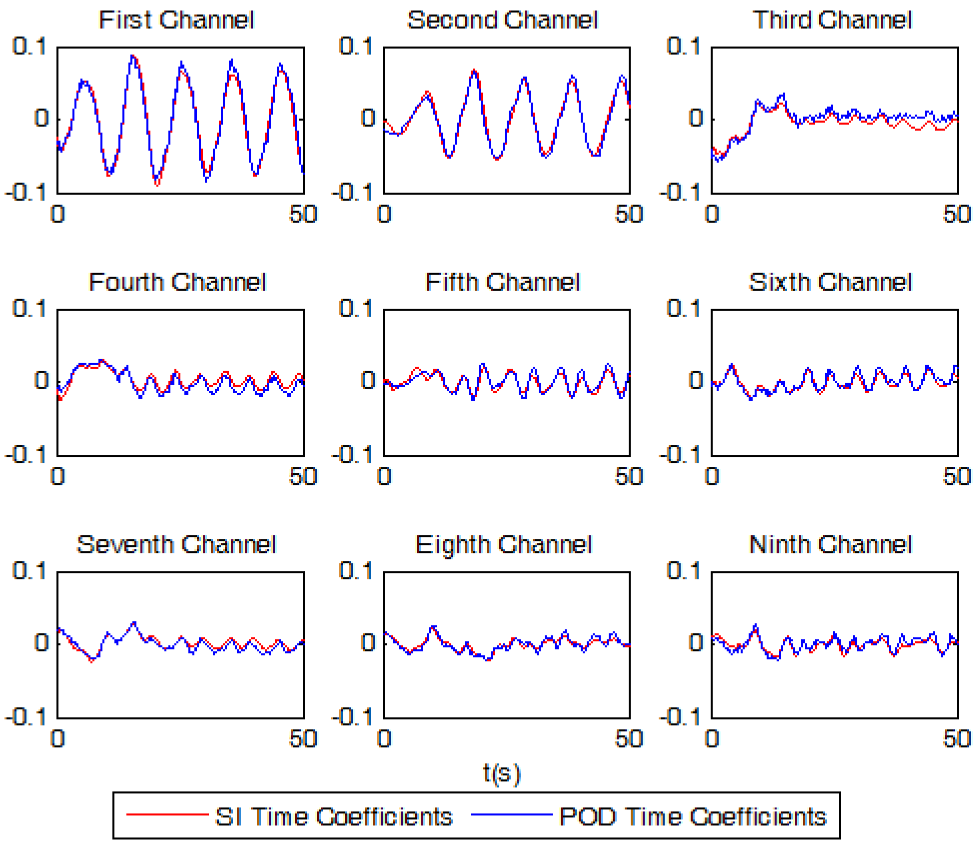


Figure 7.8 Channel Output for the pitching motion of NACA 0012 airfoil

It can be seen that the model output is acceptably close to the POD time coefficients. Hence, it can be stated that the dynamical model is successful in capturing the time variation of the flow. As a final test, a reconstruction of the flow is performed by using the dynamical model's output as the time coefficients (a_i's) in Eq. (5.1), the results of which are shown in Figure 7.9 and Figure 7.10.

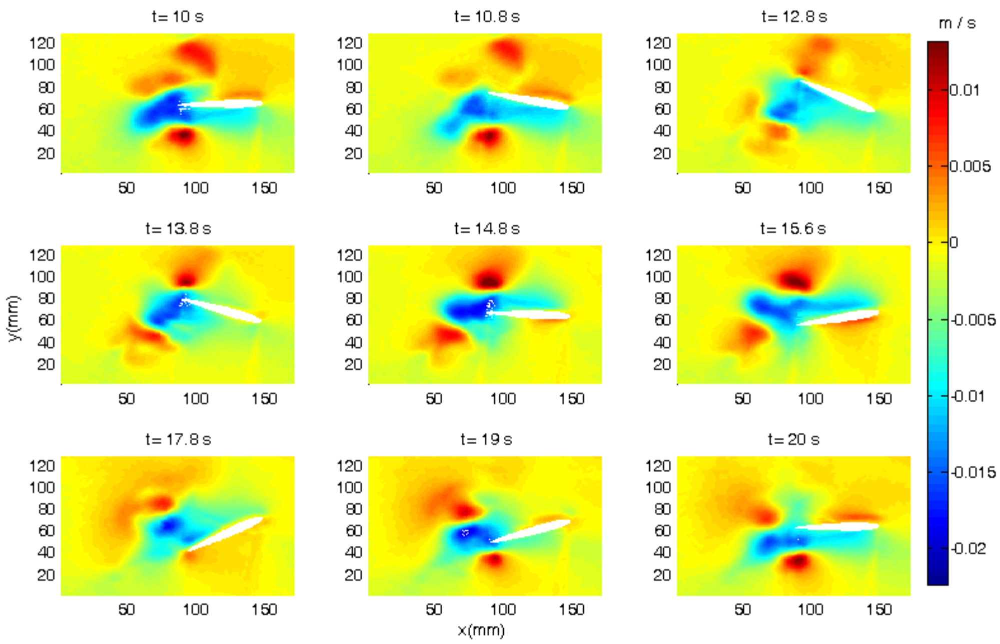


Figure 7.9 X Velocities Reconstructed with System Identification Time Coefficients for the pitching motion of NACA 0012 airfoil

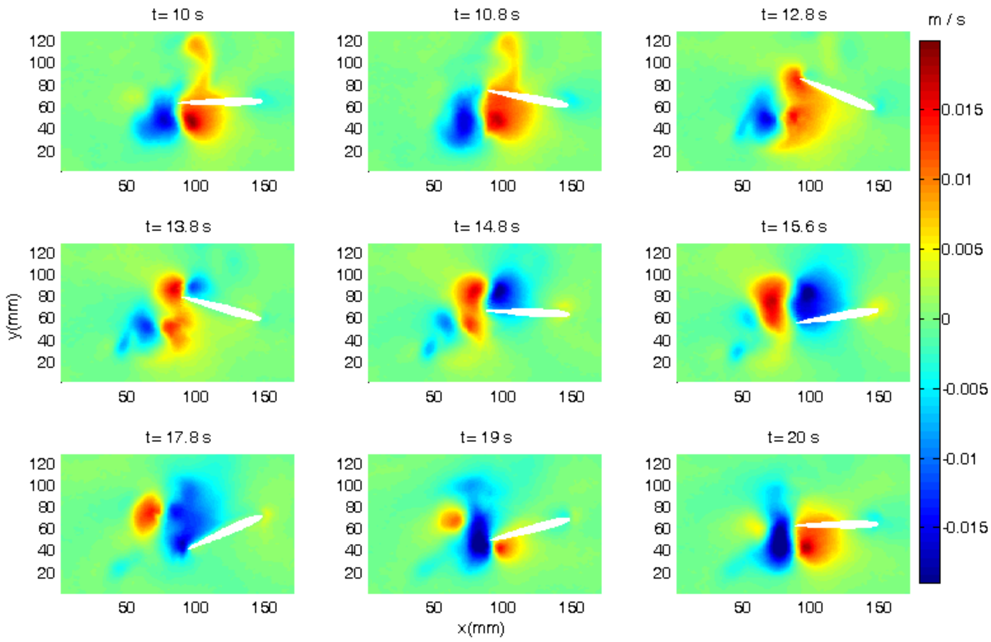


Figure 7.10 Y Velocities Reconstructed with System Identification Time Coefficients for the pitching motion of NACA 0012 airfoil

Comparing these figures with the original PIV snapshots in Figure 7.1 and Figure 7.2, one can see that the results are sufficiently close to each other. Therefore the model obtained using the procedure described in the paper can represent the flow with acceptable accuracy and can be used for various analysis and design tasks.

7.2 Results for SD 7003 Airfoil

The applied procedure for the pitching motion of SD 7003 airfoil is same with the pitching motion of NACA 0012 airfoil. Two of the images of SD 7003 airfoil obtained from the PIV measurements are shown in Figure 3.5. Figure 7.11 and Figure 7.12 show the instantaneous velocity field over SD 7003 airfoil in the X and Y directions with the appropriate airfoil location superimposed on the image. The images were taken from the second period of motion.

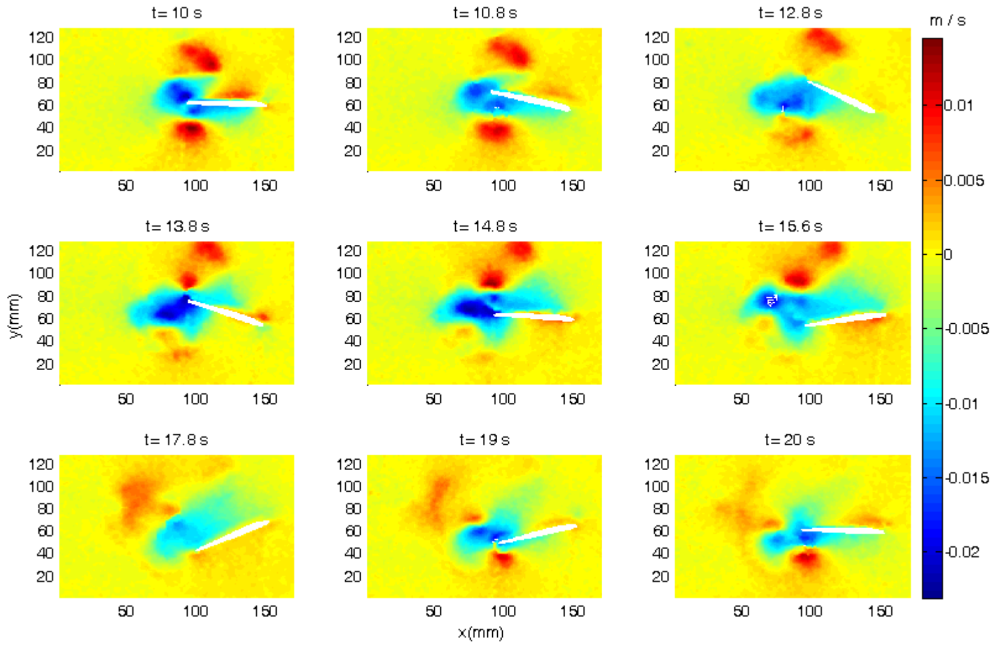


Figure 7.11 X Velocity Components of the PIV Snapshots during the second period for the pitching motion of SD 7003 airfoil

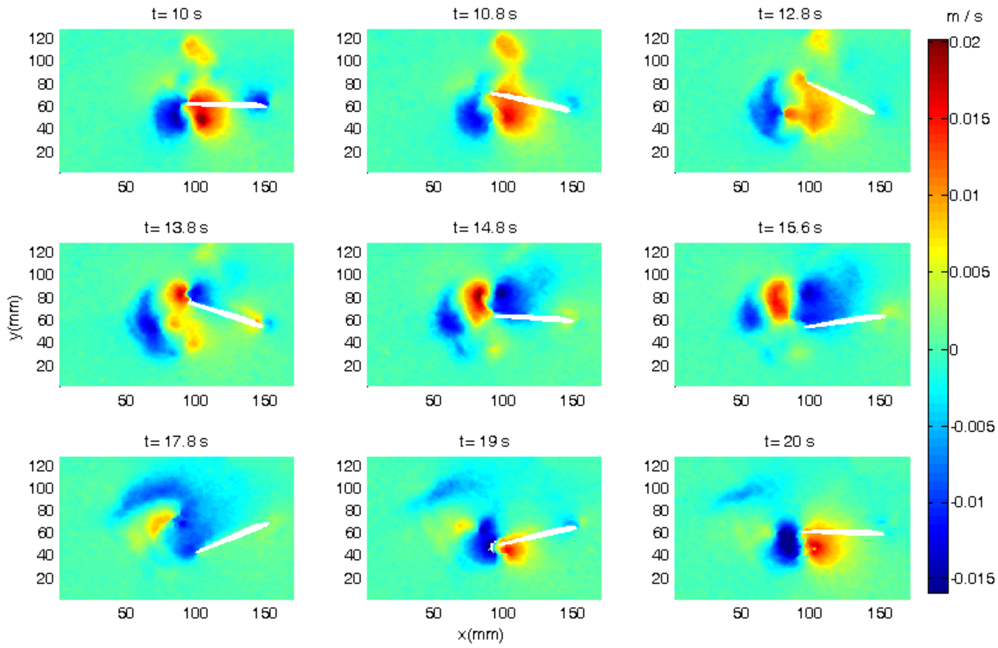


Figure 7.12 Y Velocity Components of the PIV Snapshots during the second period for the pitching motion of SD 7003 airfoil

After the velocity information inside the airfoil and its surface is filled in using Eq. (4.3) and the mean value of the flow is removed, POD modes of SD 7003 airfoil are found. The X and Y components of the first nine POD modes for SD 7003 airfoil resulting from applying POD to the flow snapshots are shown in Figure 7.13 and Figure 7.14.

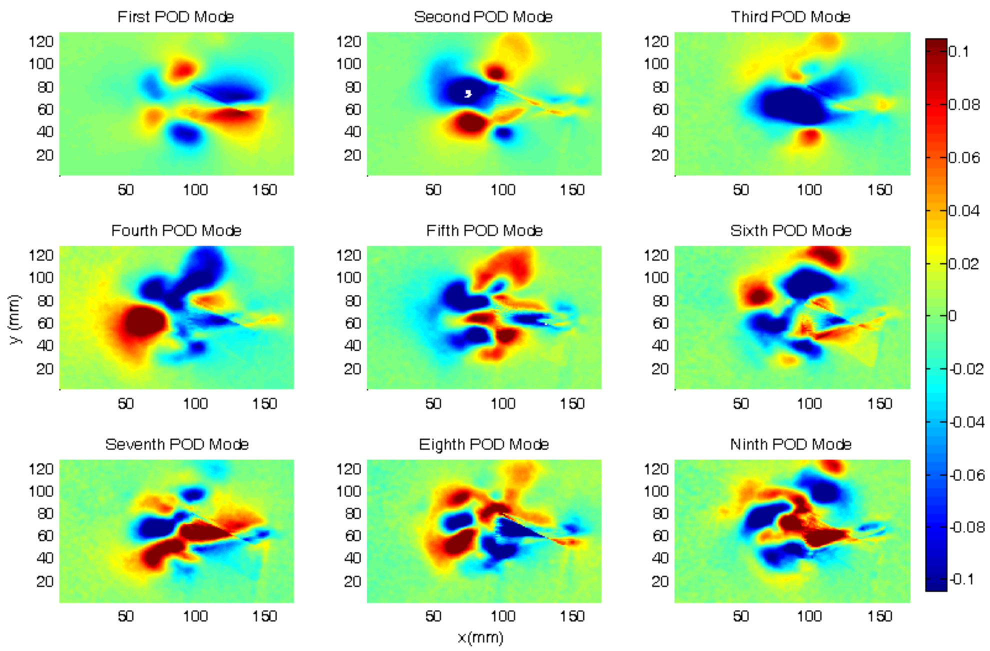


Figure 7.13 POD Modes in X Direction for the pitching motion of SD 7003 airfoil

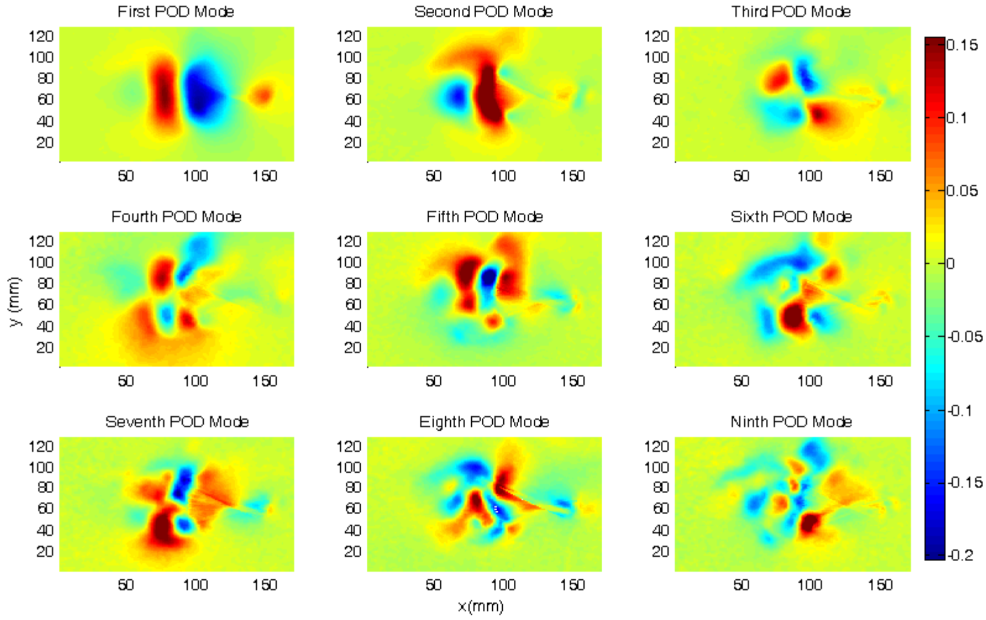


Figure 7.14 POD Modes in Y Direction for the pitching motion of SD 7003 airfoil

100 modes out of 250 modes have been chosen, which correspond roughly to 98.67% of the flow energy. Energy level vs. number of modes for SD 7003 airfoil can be seen in Figure 7.15.

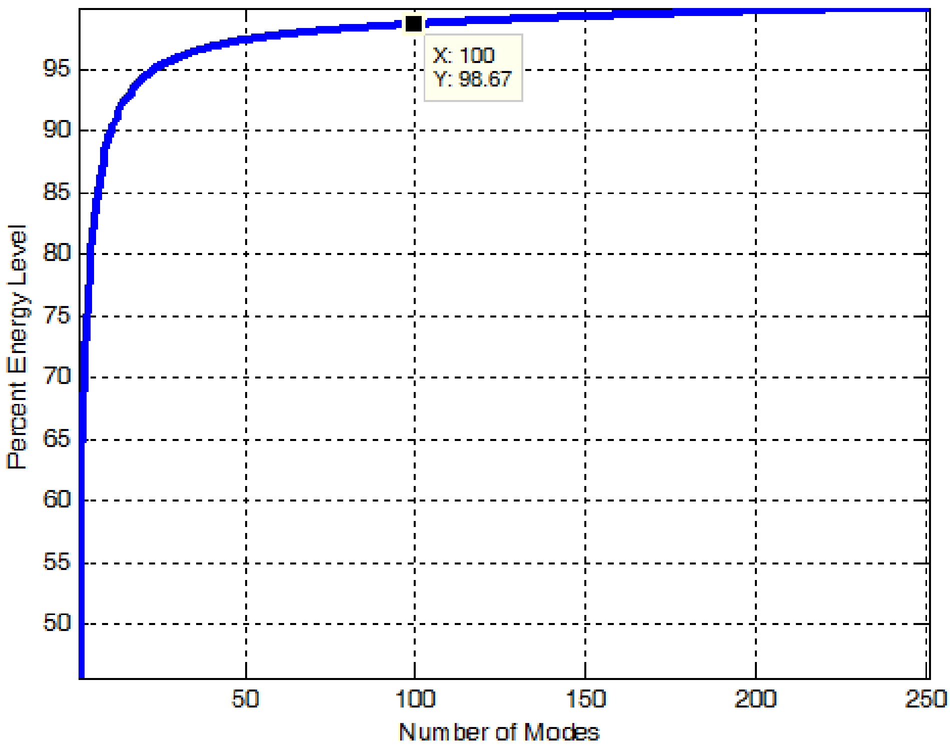


Figure 7.15 Energy Level vs Number of POD Modes for the pitching motion of SD 7003 airfoil

To verify that the selected number of modes can represent the flow accurately, reconstruction of the flow is obtained by using Eq. (5.1), where the time coefficients a_i are obtained by projecting the flow snapshots onto the POD modes as expressed in Eq. (5.7). The X and Y components of the reconstructed flow velocity for SD 7003 airfoil are shown in Figure 7.16 and Figure 7.17.

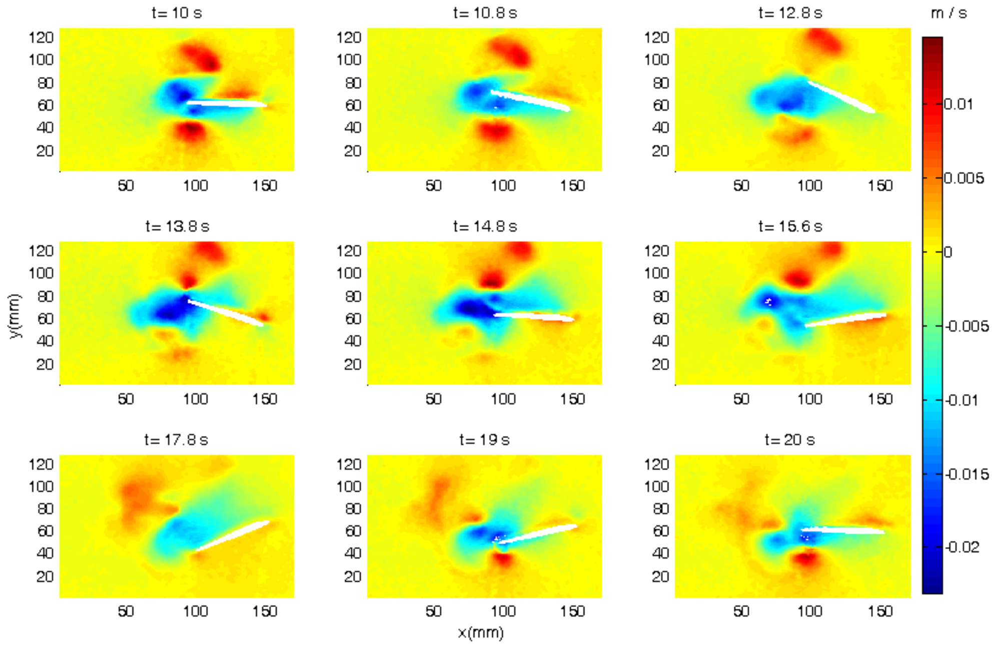


Figure 7.16 X Velocities Reconstructed with POD Time Coefficients for the pitching motion of SD 7003 airfoil

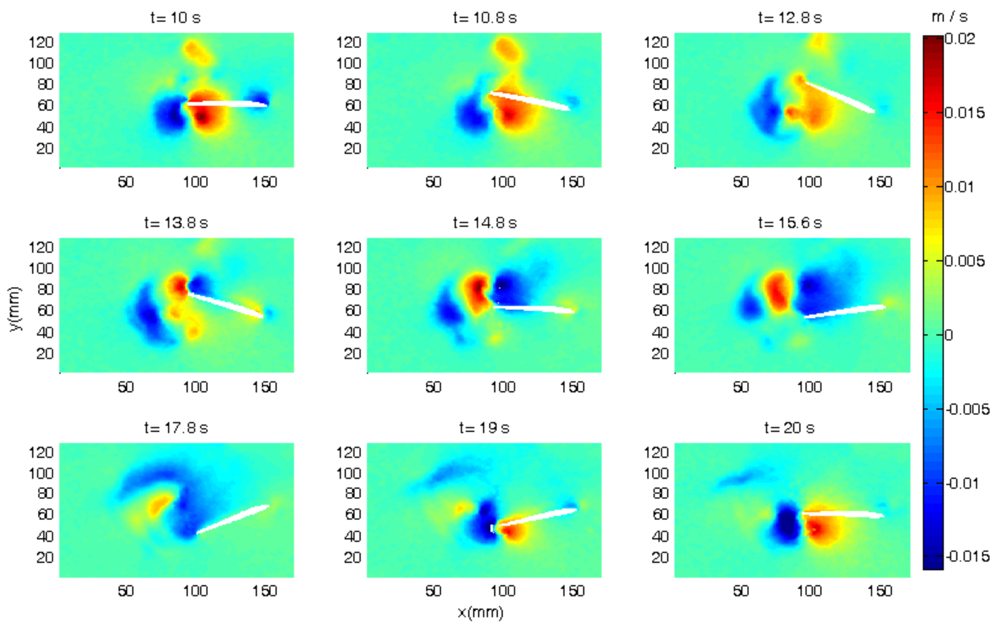


Figure 7.17 Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of SD 7003 airfoil

It can be seen that the reconstructed flow velocity for the pitching motion of SD 7003 airfoil is quite similar to snapshots obtained from PIV shown in Figure 7.11 and Figure 7.12. Then, a discrete time state space dynamical model is constructed by using N4SID system identification technique for SD 7003 airfoil. The order of this model is 40. The comparison between the outputs of the model with the POD time coefficients can be seen in Figure 7.18. It can be seen that the model output is sufficiently close to the POD time coefficients for SD 7003 airfoil. Thus, it can be said that the dynamical model is successful in capturing the time variation of the flow. Finally, a reconstruction of the flow is performed by using the dynamical model's output as the time coefficients (a_i's) in Eq. (5.1), the results of which are shown in Figure 7.19 and Figure 7.20.

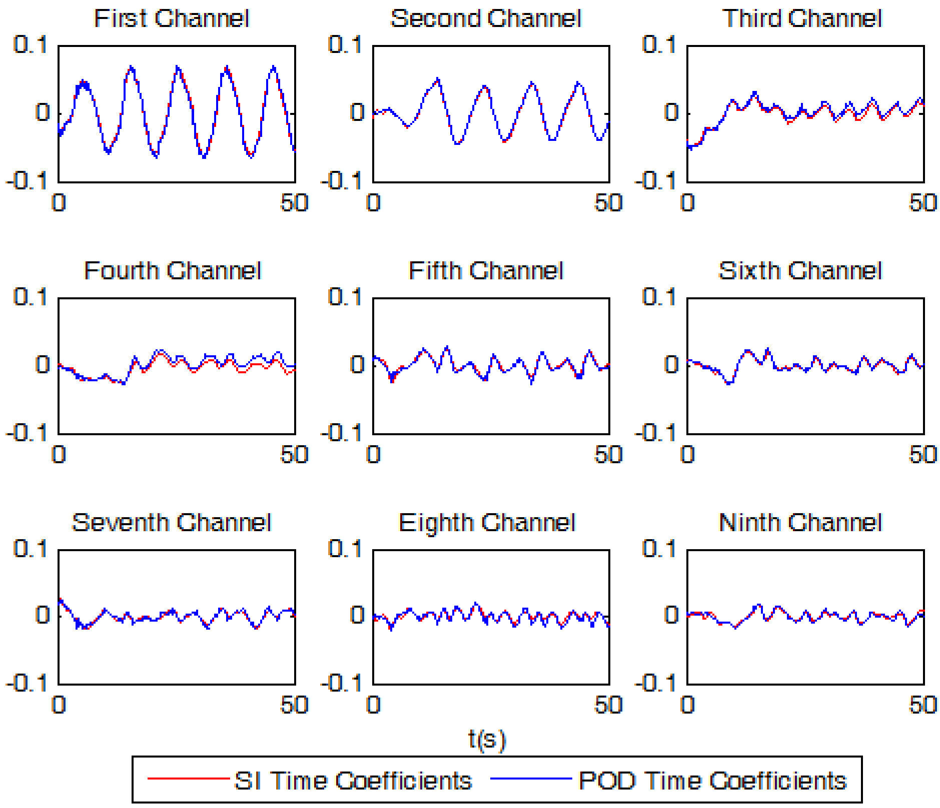


Figure 7.18 Channel Output for the pitching motion of SD 7003 airfoil

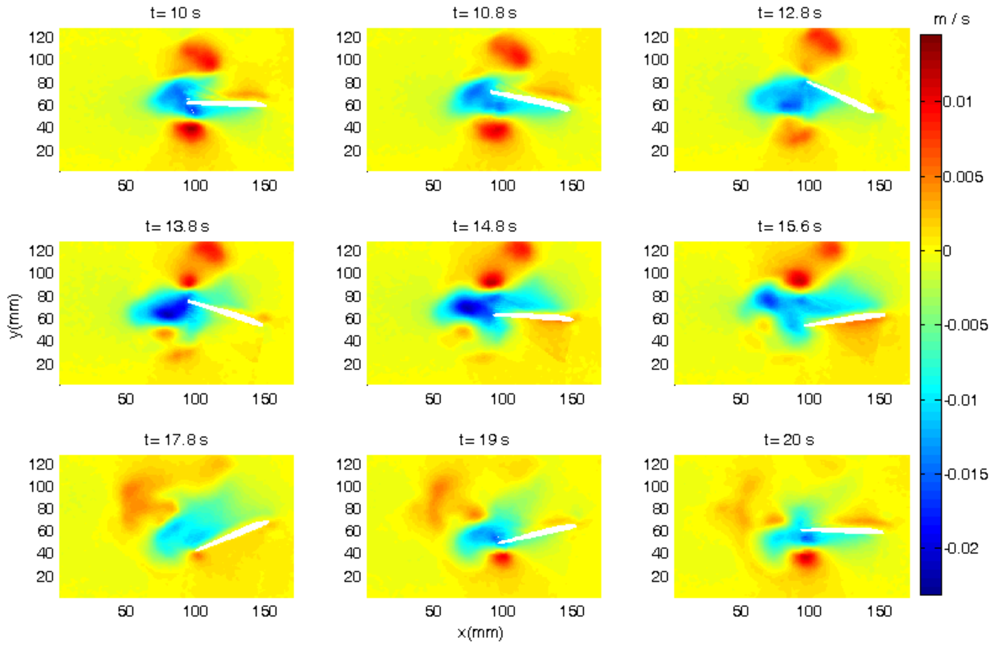


Figure 7.19 X Velocities Reconstructed with SI Time Coefficients for the pitching motion of SD 7003 airfoil

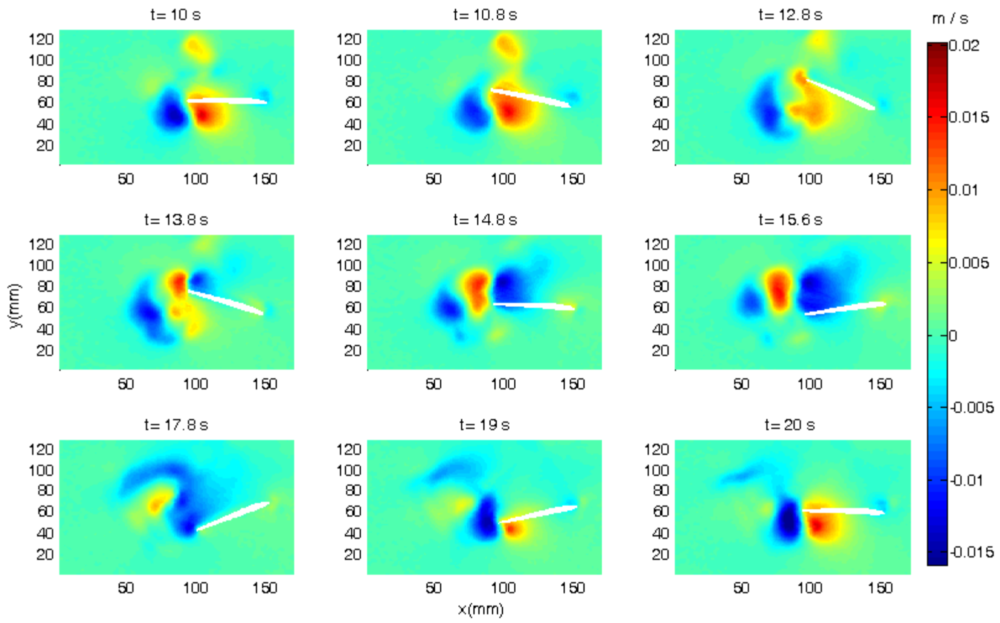


Figure 7.20 Y Velocities Reconstructed with SI Time Coefficients for the pitching motion of SD 7003 airfoil

If Figure 7.19 and Figure 7.20 are compared with the original PIV snapshots in Figure 7.11 and Figure 7.12, it can be seen that the results are very close to each other. Thus the model obtained can represent the pitching motion of SD 7003 with acceptable accuracy. This model can be used for various analysis and design tasks.

7.3 Results for Elliptic Airfoil

The applied procedure for the pitching motion of elliptic airfoil is same with the pitching motion of NACA 0012 airfoil. Two of the images of SD 7003 airfoil obtained from the PIV measurements are shown in Figure 3.6. The instantaneous velocity field over SD 7003 airfoil in the X and Y directions with the appropriate airfoil location superimposed on the image can be seen in Figure 7.21 and Figure 7.22. The images were taken from the second period of motion.

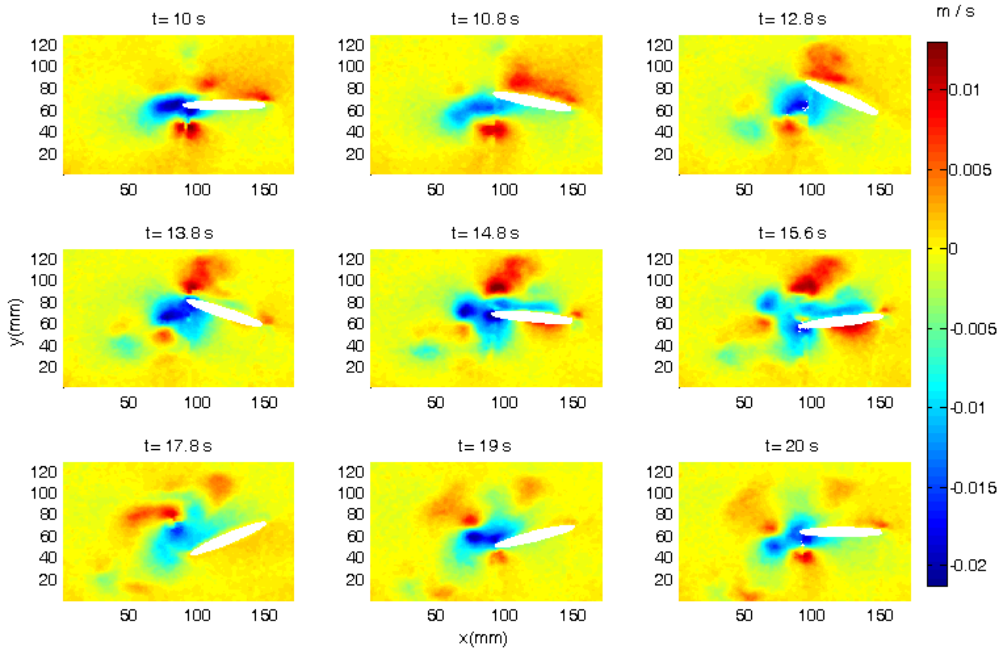


Figure 7.21 X Velocity Components of the PIV Snapshots during the second period for the pitching motion of elliptic airfoil

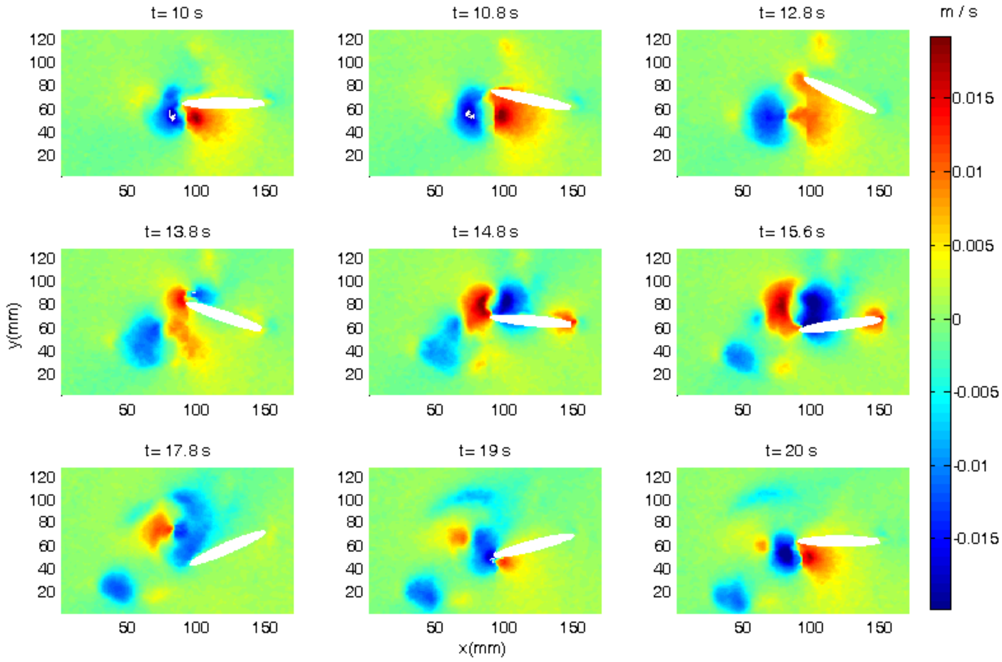


Figure 7.22 Y Velocity Components of the PIV Snapshots during the second period for the pitching motion of elliptic airfoil

POD modes of the pitching motion of elliptic airfoil is found after the velocity information is added inside the airfoil and its surface by using Eq. (4.3) and the mean value of the flow is removed. The X and Y components of the first nine POD modes for the pitching motion of elliptic airfoil resulting from applying POD to the flow snapshots are shown in Figure 7.23 and Figure 7.24.

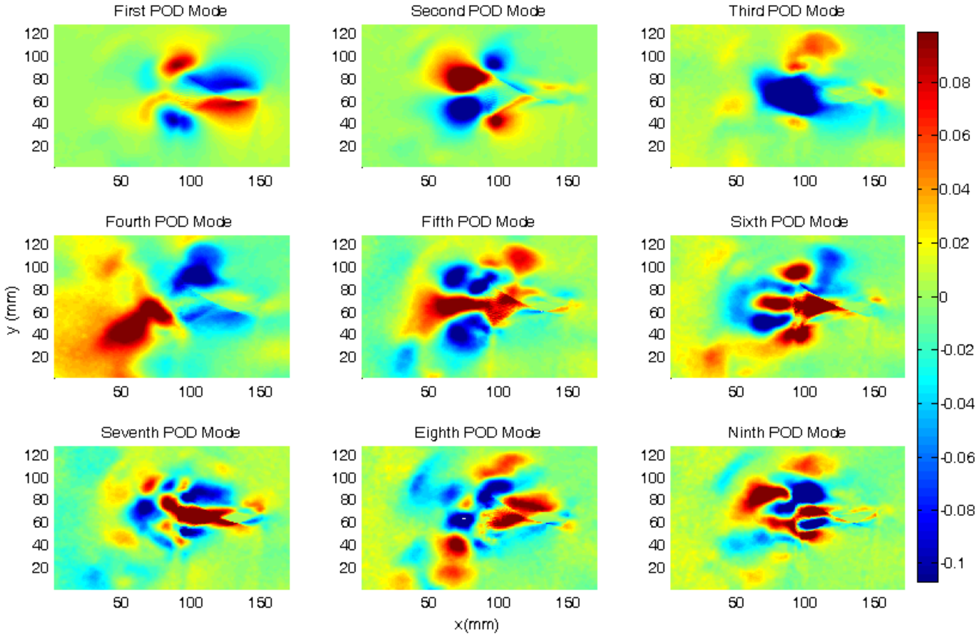


Figure 7.23 POD Modes in X Direction for the pitching motion of elliptic airfoil

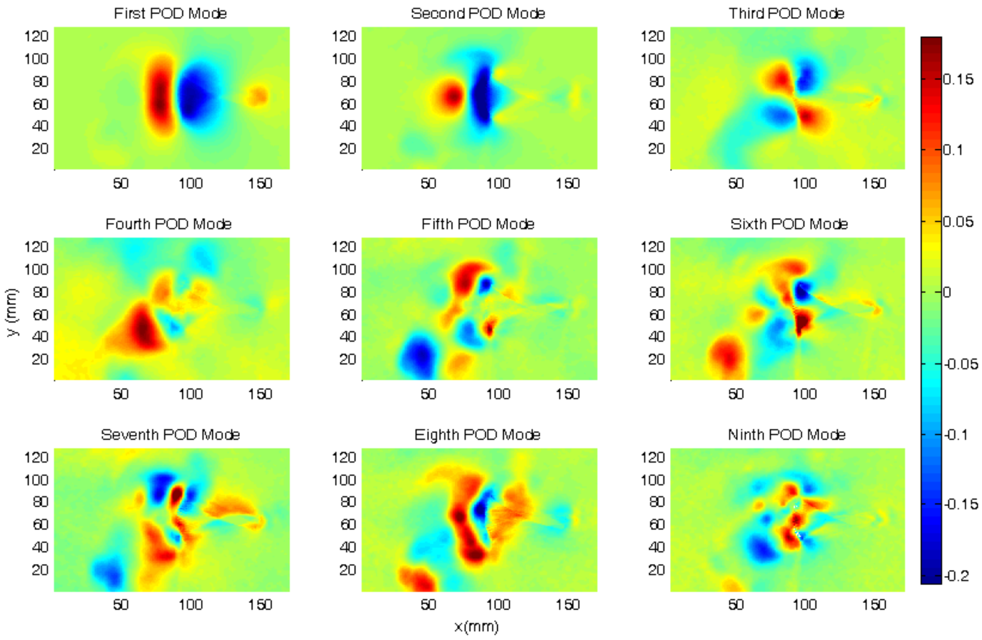


Figure 7.24 POD Modes in Y Direction for the pitching motion of elliptic airfoil

100 modes out of 250 modes have been chosen, which correspond roughly to 98.86% of the flow energy. Energy level vs. number of modes for elliptic airfoil can be seen in Figure 7.25. In order to confirm that the selected number of modes can represent the flow accurately, reconstruction of the flow is obtained by using Eq. (5.1), where the time coefficients a_i are obtained by projecting the flow snapshots onto the POD modes as expressed in Eq. (5.7). The X and Y components of the reconstructed flow velocity for pitching motion of elliptic airfoil are shown in Figure 7.26 and Figure 7.27.

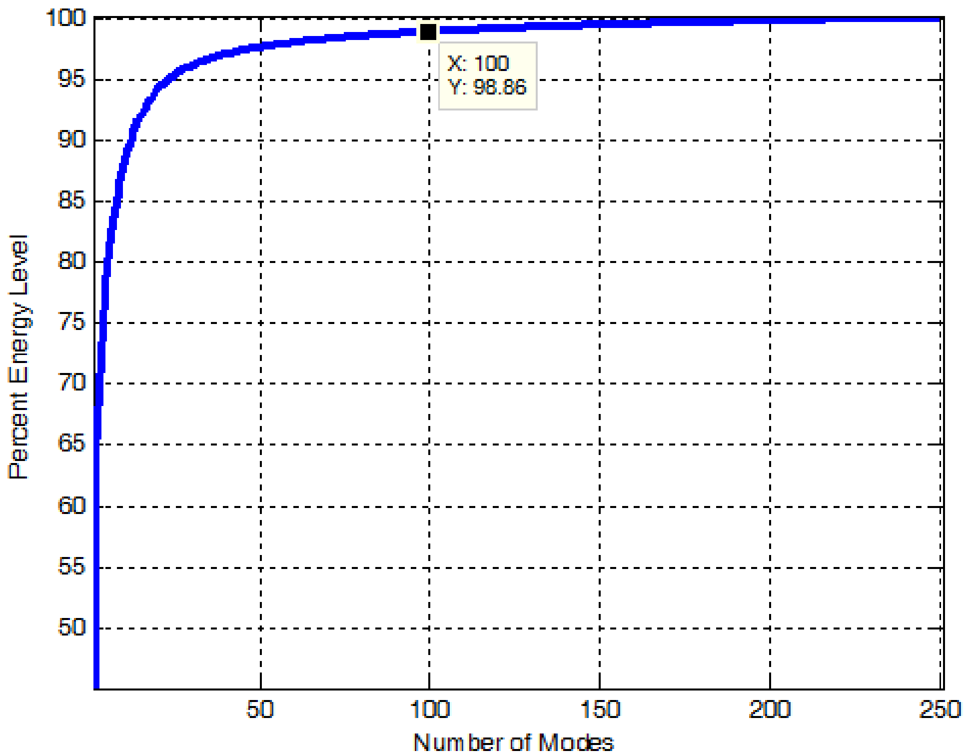


Figure 7.25 Energy Level vs Number of POD Modes for the pitching motion of elliptic airfoil

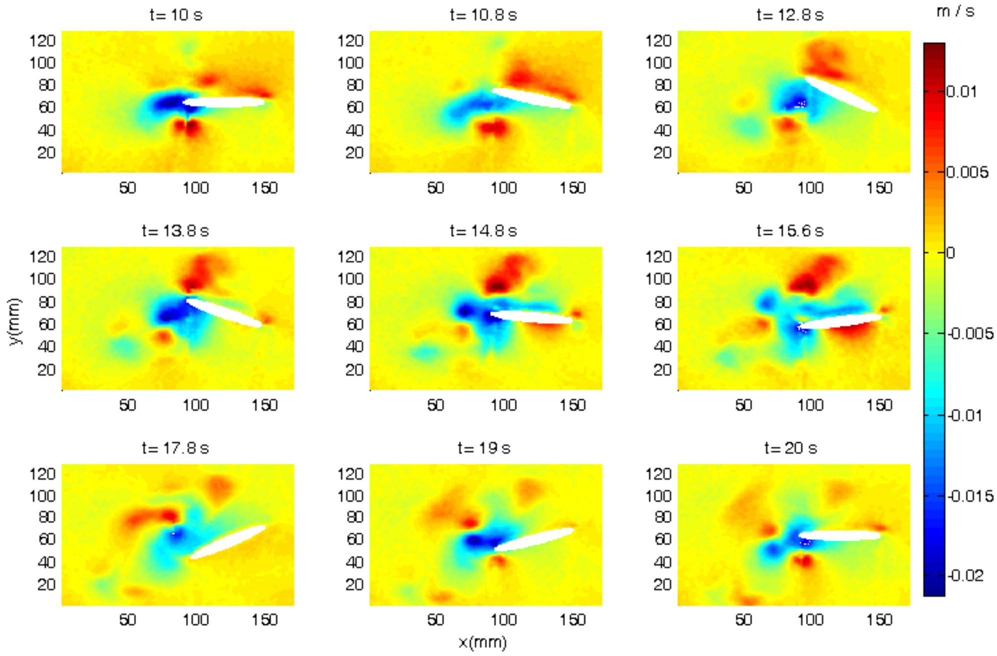


Figure 7.26 X Velocities Reconstructed with POD Time Coefficients for the pitching motion of elliptic airfoil

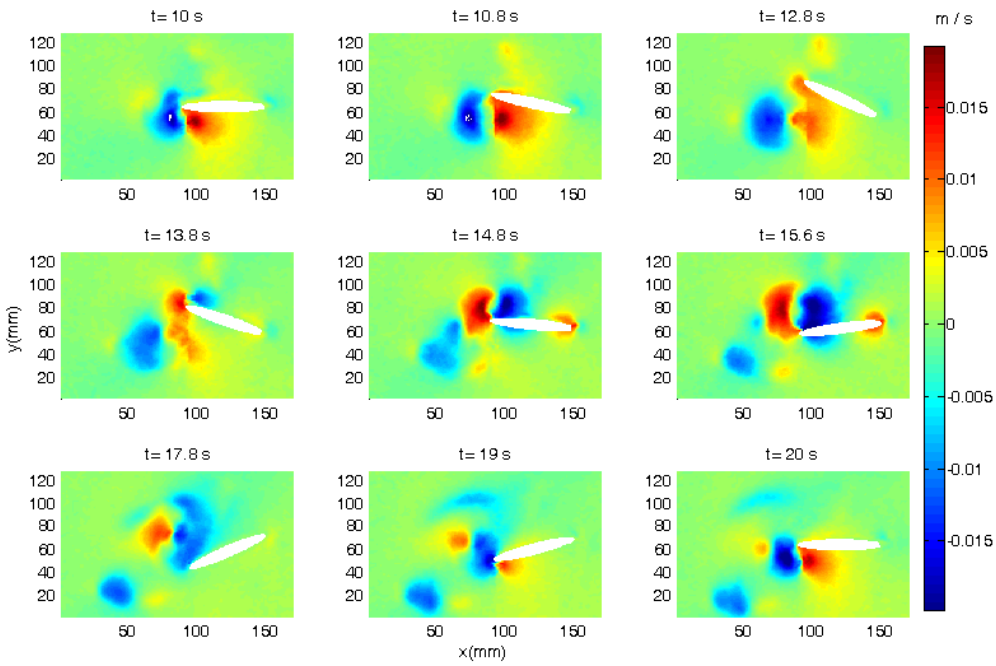


Figure 7.27 Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of elliptic airfoil

It can be seen that the reconstructed flow velocity for the pitching motion of elliptic airfoil is very similar to snapshots obtained from PIV shown in Figure 7.21 and Figure 7.22. Then, a discrete time state space dynamical model is constructed by using N4SID system identification technique for elliptic airfoil. The order of this model is 40. The comparison between the outputs of the model with the POD time coefficients can be seen in Figure 7.28. It can be seen that the model output is adequately close to the POD time coefficients for the pitching motion of elliptic airfoil. Therefore, it can be said that the dynamical model is successful in capturing the time variation of the flow. Finally, a reconstruction of the flow is performed by using the dynamical model's output as the time coefficients (a_i's) in Eq. (5.1), the results of which are shown in Figure 7.29 and Figure 7.30.

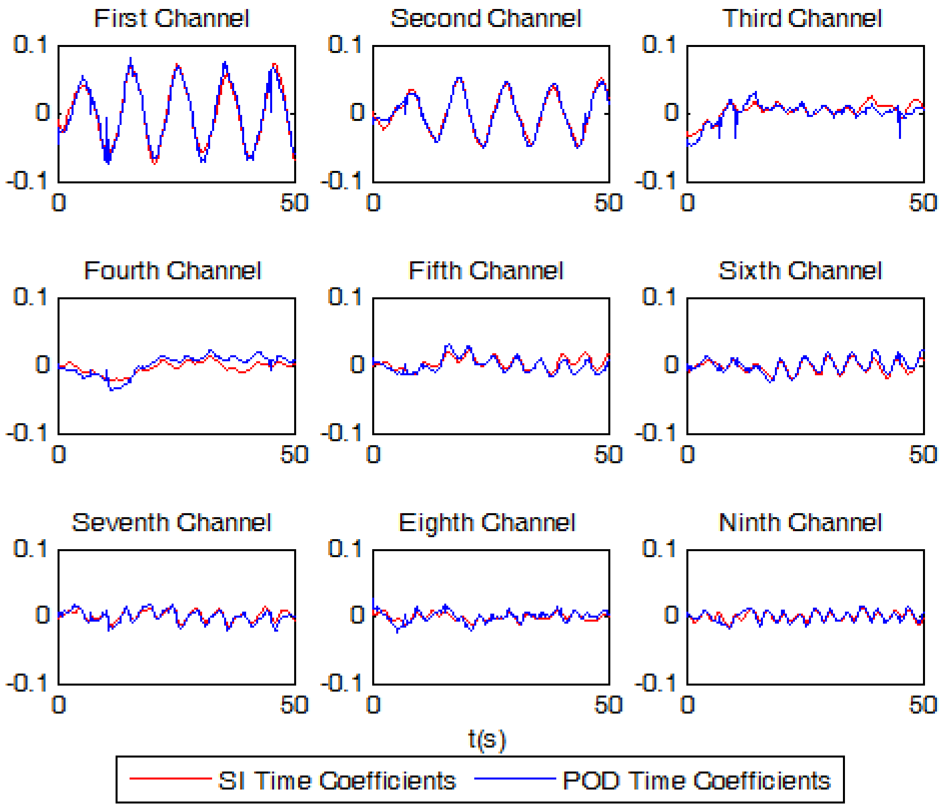


Figure 7.28 Channel Output for the pitching motion of elliptic airfoil

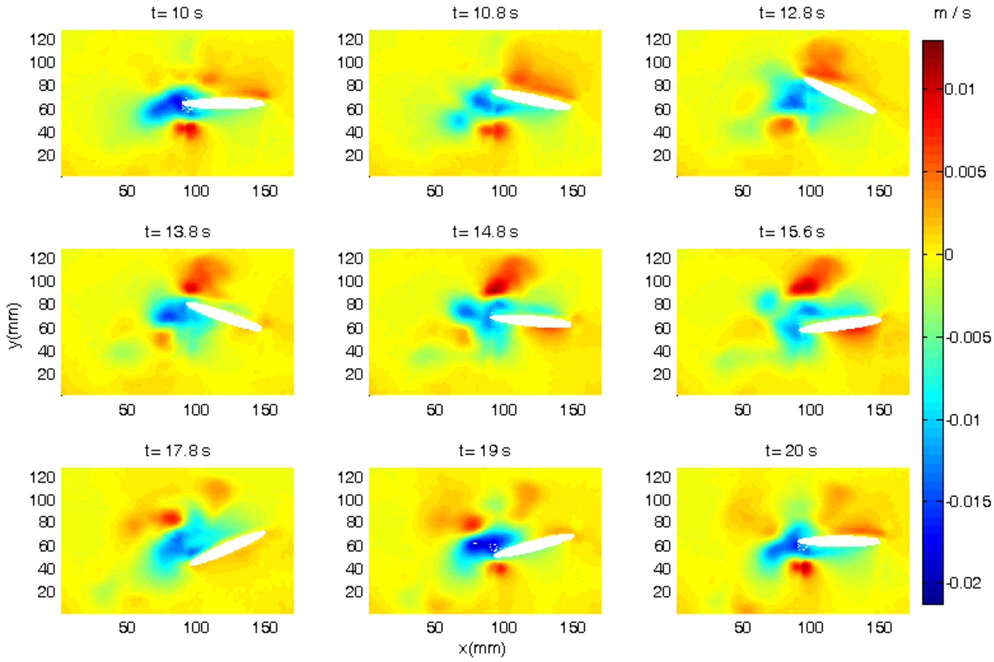


Figure 7.29 X Velocities Reconstructed with SI Time Coefficients for the pitching motion of elliptic airfoil

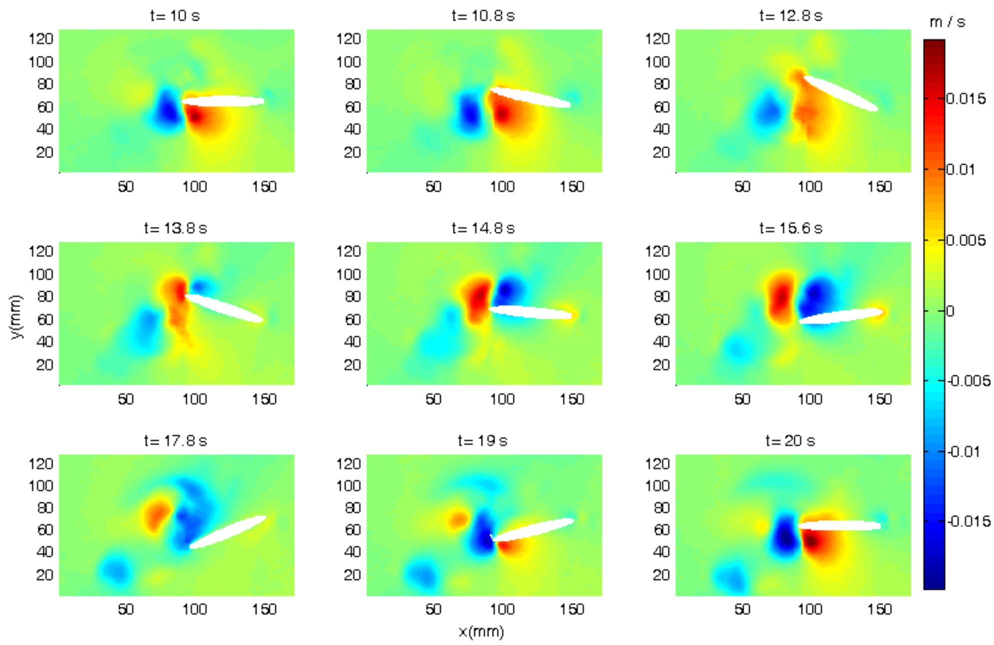


Figure 7.30 Y Velocities Reconstructed with SI Time Coefficients for the pitching motion of elliptic airfoil

If Figure 7.29 and Figure 7.30 are compared with the original PIV snapshots in Figure 7.21 and Figure 7.22, it can be seen that the results are adequately close to each other. Hence, the model obtained can represent the pitching motion of elliptic airfoil with acceptable accuracy. This model can be used for various analysis and design tasks.

7.4 Results for Flat Plate

The applied procedure for the pitching motion of flat plate is same with the pitching motion of NACA 0012 airfoil. Two of the images of flat plate obtained from the PIV measurements can be seen in Figure 3.7. Figure 7.31 and Figure 7.32 display the instantaneous velocity field over flat plate in the X and Y directions with the appropriate airfoil location superimposed on the image. The images were taken from the second period of pitching motion.

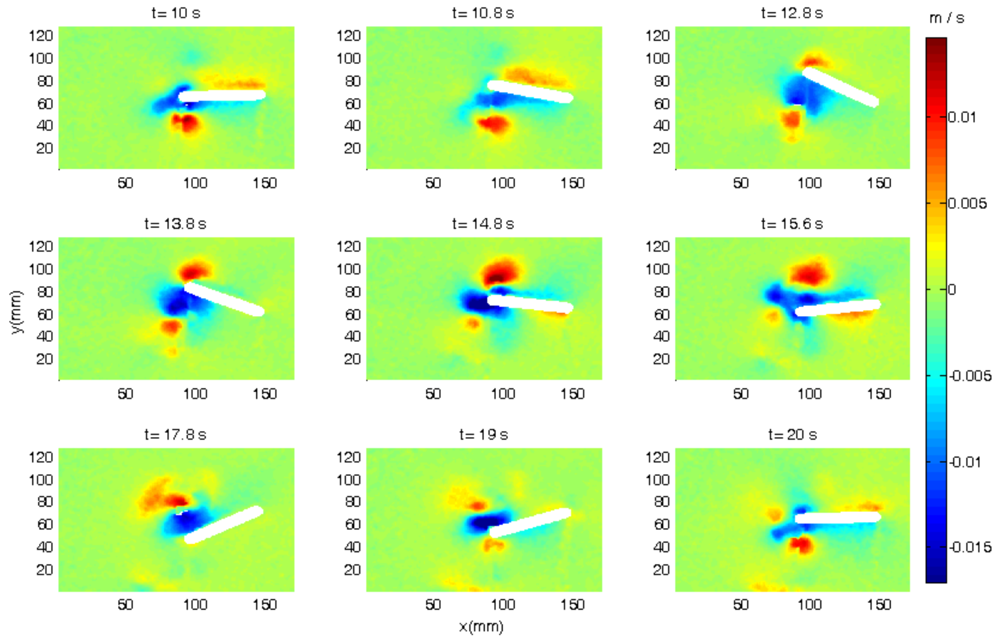


Figure 7.31 X Velocity Components of the PIV Snapshots during the second period for the pitching motion of flat plate

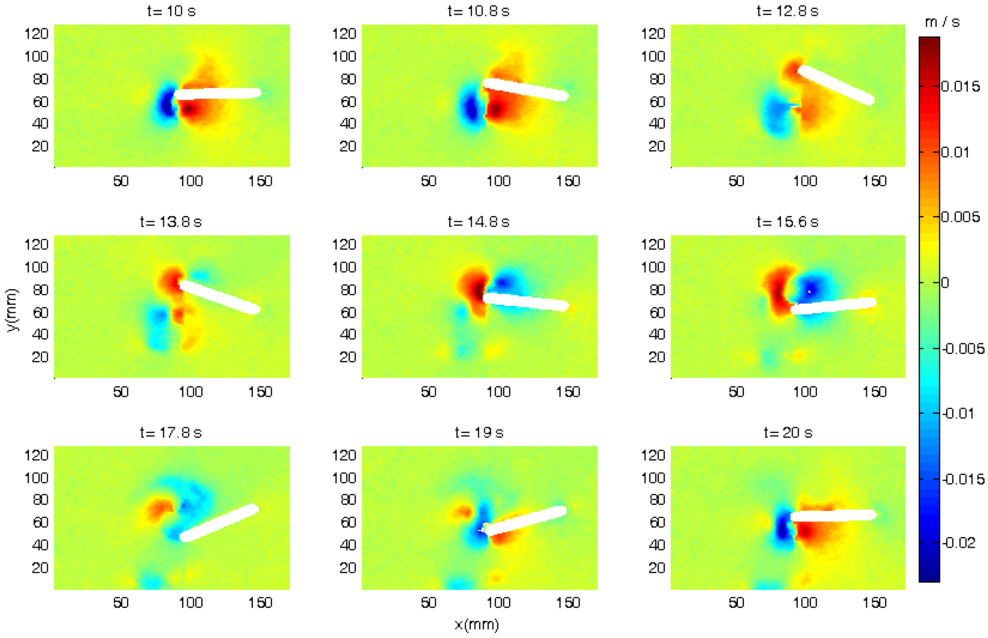


Figure 7.32 Y Velocity Components of the PIV Snapshots during the second period for the pitching motion of flat plate

POD modes of the pitching motion of flat plate is obtained after the velocity information is added inside the airfoil and its surface by using Eq. (4.3) and the mean value of the flow is removed. The X and Y components of the first nine POD modes for the pitching motion of flat plate resulting from applying POD to the flow snapshots can be seen in Figure 7.33 and Figure 7.34.

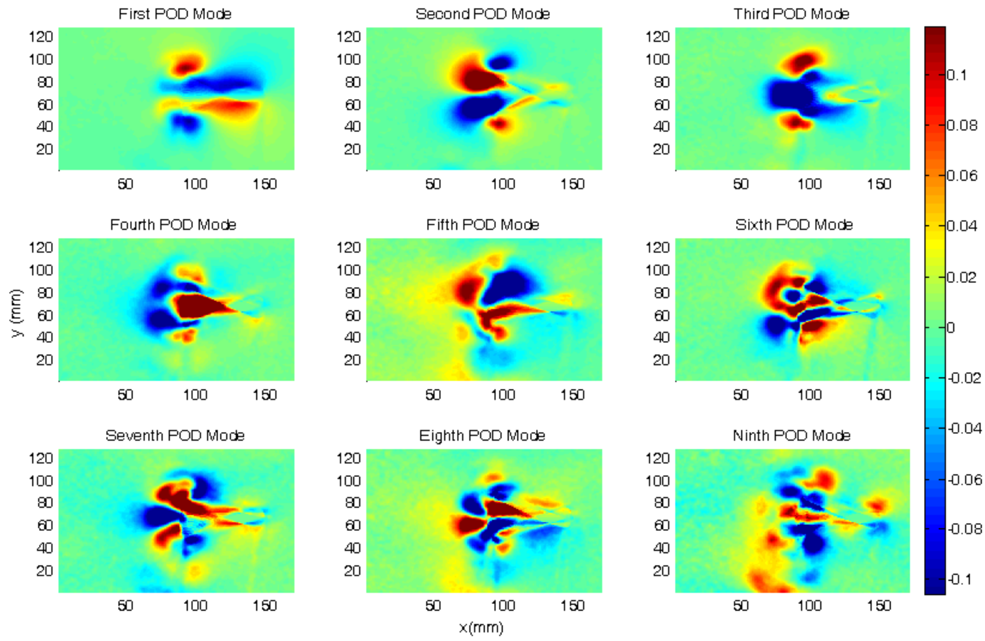


Figure 7.33 POD Modes in X Direction for the pitching motion of flat plate

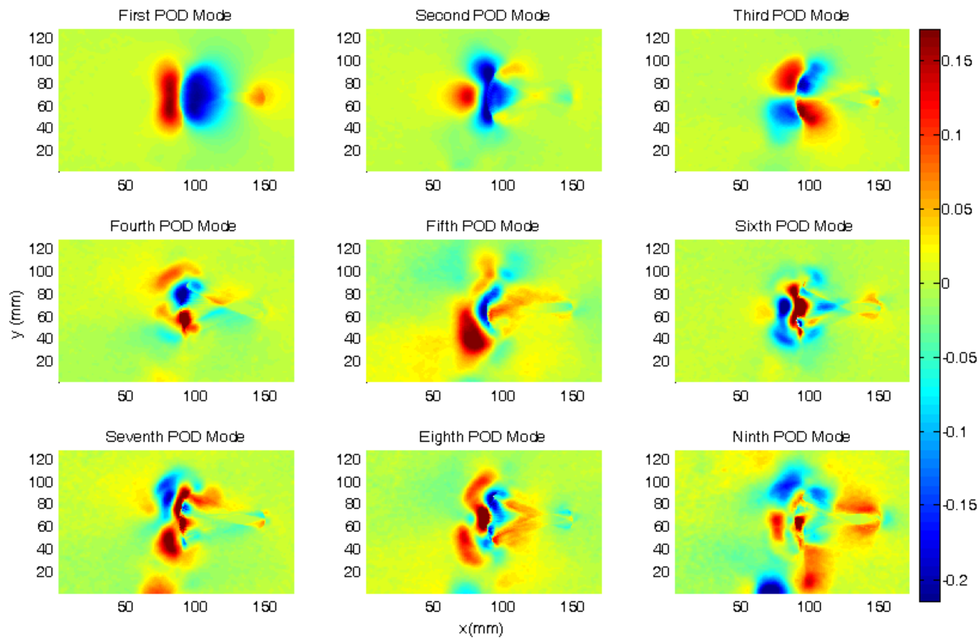


Figure 7.34 POD Modes in Y Direction for the pitching motion of flat plate

100 modes out of 250 modes have been chosen, which correspond roughly to 98.65% of the flow energy. Energy level vs. number of modes for flat plate can be seen in Figure 7.35. To confirm that the selected number of modes can represent the flow accurately, reconstruction of the flow is found by using Eq. (5.1), where the time coefficients a_i are obtained by projecting the flow snapshots onto the POD modes as expressed in Eq. (5.7). The X and Y components of the reconstructed flow velocity for pitching motion of flat plate can be seen in Figure 7.36 and Figure 7.37.

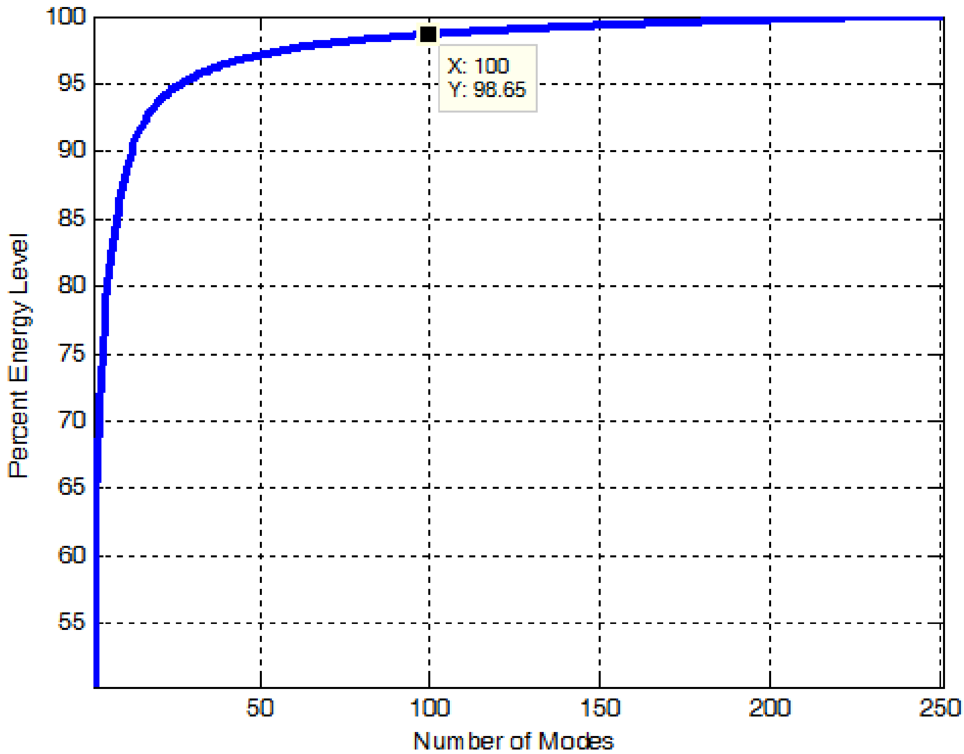


Figure 7.35 Energy Level vs Number of POD Modes for the pitching motion of flat plate

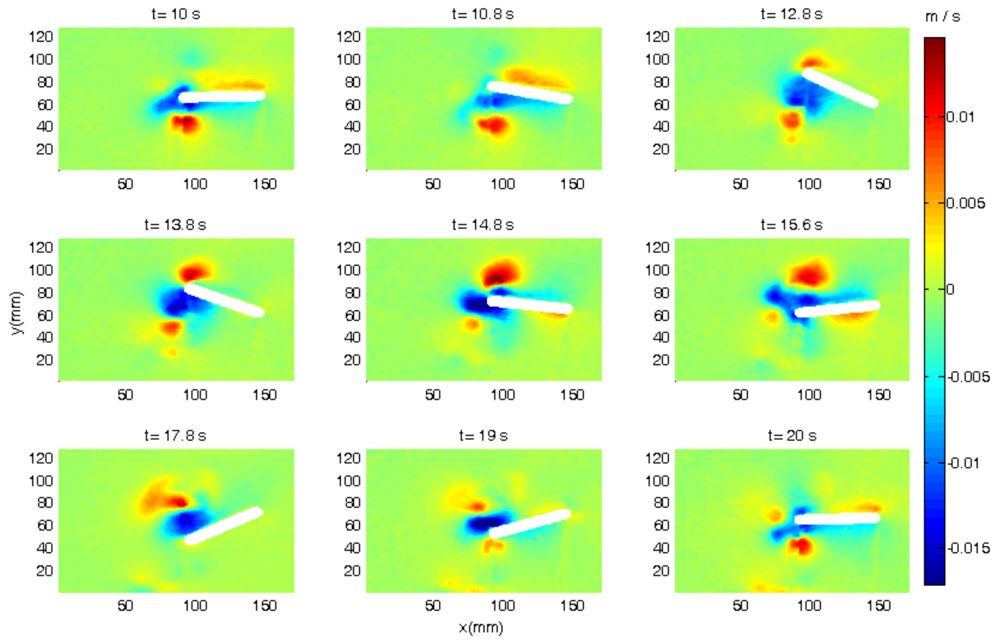


Figure 7.36 X Velocities Reconstructed with POD Time Coefficients for the pitching motion of flat plate

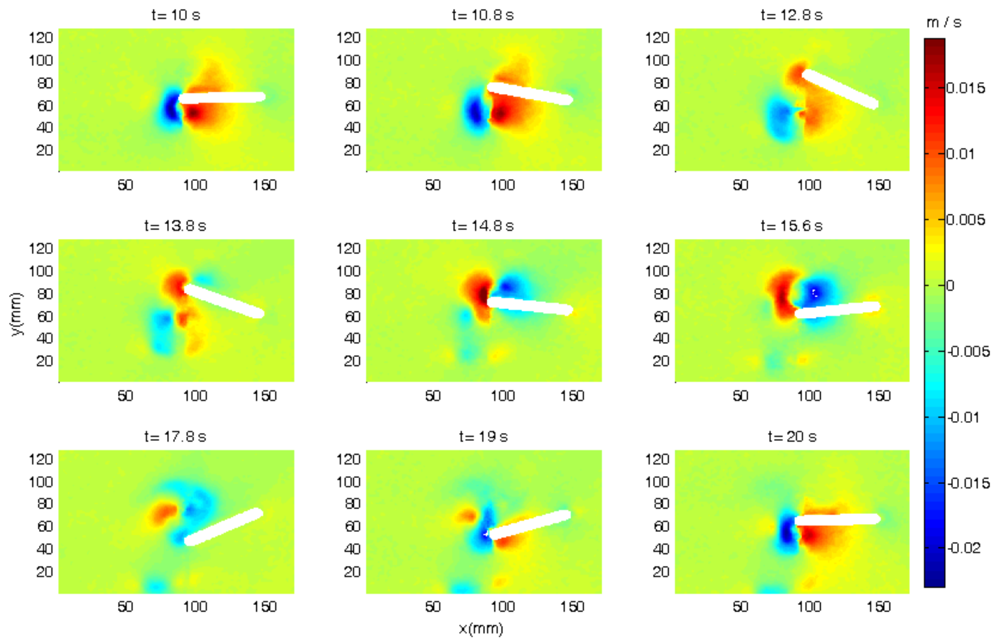


Figure 7.37 Y Velocities Reconstructed with POD Time Coefficients for the pitching motion of flat plate

It can be seen that the reconstructed flow velocity for the pitching motion of flat plate is quite similar to snapshots obtained from PIV shown in Figure 7.31 and Figure 7.32. Then, a discrete time state space dynamical model is constructed by using N4SID system identification technique for flat plate. The order of this model is 40. The comparison between the outputs of the model with the POD time coefficients can be seen in Figure 7.38. When the model output is compared with POD time coefficients, it can be seen that the model output is adequately close to the POD time coefficients for the pitching motion of flat plate. Hence, the dynamical model is successful in capturing the time variation of the flow. Finally, a reconstruction of the flow is performed by using the dynamical model's output as the time coefficients (a_i's) in Eq. (5.1), the results of which are shown in Figure 7.39 and Figure 7.40.

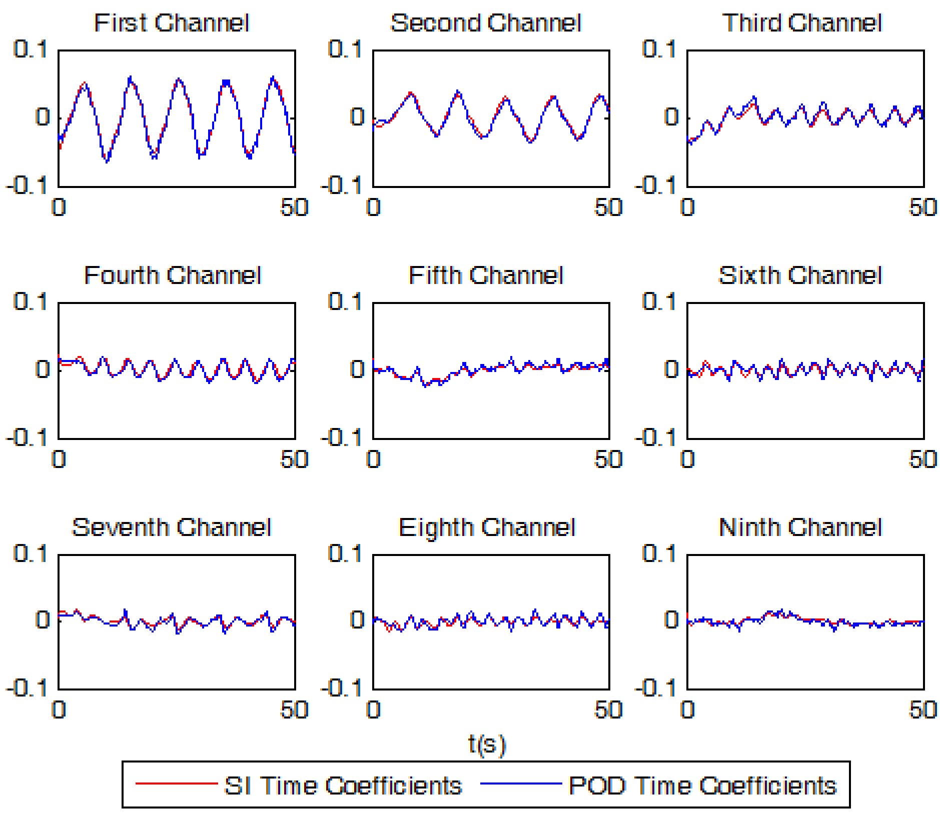


Figure 7.38 Channel output for the pitching motion of flat plate

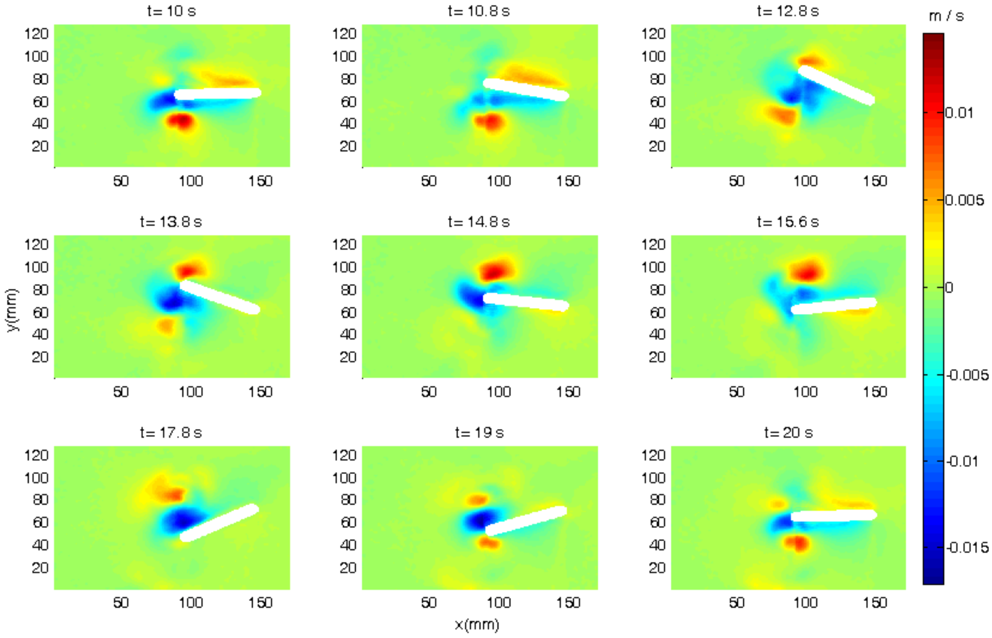


Figure 7.39 X Velocities Reconstructed with SI Time Coefficients for the pitching motion of flat plate

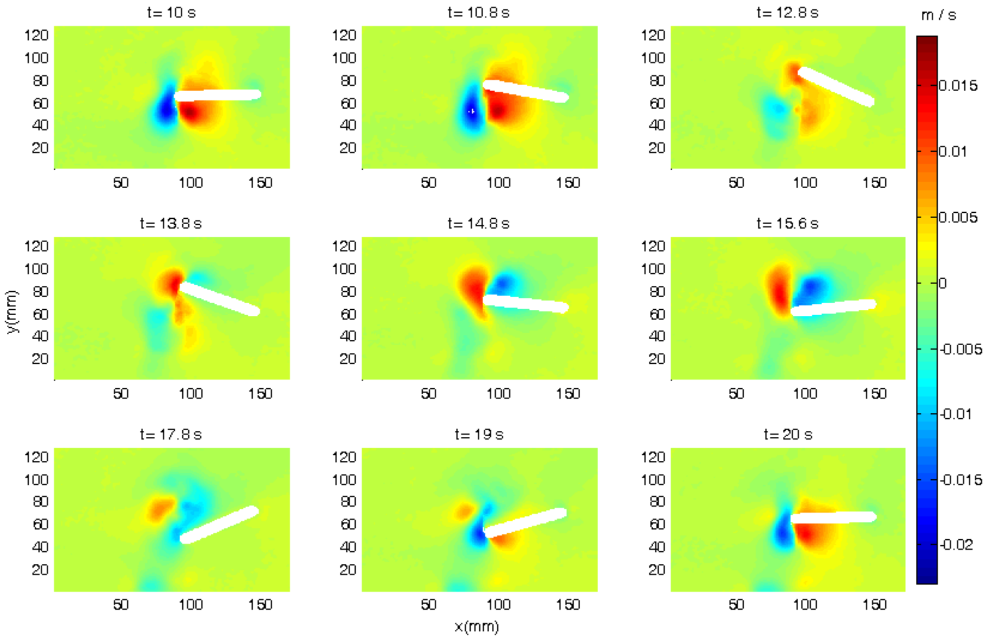


Figure 7.40 Y Velocities Reconstructed with SI Time Coefficients for the pitching motion of flat plate

If Figure 7.39 and Figure 7.40 are compared with the original PIV snapshots in Figure 7.31 and Figure 7.32, it can be seen that the results are very close to each other. Therefore, the model obtained can represent the pitching motion of flat plate with acceptable accuracy. This model can be used for various analysis and design tasks.

CHAPTER VIII

CONCLUSION

In this thesis, a modeling approach to represent the dynamical behavior of a pitching airfoil is considered. The method is based on experimental PIV images, which are processed to obtain the instantaneous velocity field of the flow. In order to determine the location and orientation of the airfoil, an image processing algorithm is implemented. This information is used to fill the region in the images corresponding to the airfoil with proper velocity data. These post-processed images are used to compute the POD modes and time coefficients of the system. It was observed that nearly 99% of the flow energy can be captured by using 100 modes for each experiment and a reconstruction for this case represents the original flow satisfactorily. When energy levels for 100 modes are compared for each type of airfoil, the model of the pitching motion of NACA 0012 captures %98.93 of the flow energy therefore it is the most accurate case to represent the original flow. A discrete-time state space model of order 40 was then fit to the time coefficient trajectories. Comparing the model outputs to the time coefficient trajectories and also obtaining reconstruction of the flow using these outputs as time coefficients, it was seen that the model represents the time variation of the flow with sufficient accuracy. Such a mathematical model can be used in analyzing certain characteristics of the flow (e.g. stability) or performing certain synthesis tasks (e.g. controller design).

100 modes are good at capturing the flow energy and provide similarity between model outputs and experimental data. However, the order of discrete-time state space model is 40, which is high. If less than 100 modes are selected, similarity of outputs

and capturing the flow energy are reduced. In this case the order of discrete-time state space model can be less than 40.

For future work, chirp, triangle or square waves will be used for the input of the system, which can be blown at any point on airfoil during the simulation. The effect of the input on the dynamical model will be investigated. The time variation of the flow will try to be captured.

REFERENCES

[1] Dial K. P., *An inside look at how birds fly: experimental studies of the internal and external processes controlling flight*, 1994 Report the Aerospace Profession, 38th Symposium Proceedings, The Beverly Hilton, Beverly Hills, CA, 1994

[2] Ho S., Nassef H., Pornsinsirirak N., *Unsteady aerodynamics and flow control for flapping wing flyers*, Progress in Aerospace Science, pp. 635–681, 2003

[3] Azuma A., Masato O., Kunio Y., T.J. Mueller T.J., *Aerodynamic characteristics of wings at low Reynolds Numbers*, Fixed and flapping wings aerodynamics for micro air vehicle applications, In Progress in Astro. & Aero Vol. 195, pp.341–398., 2001

[4] Fearing R.S., Chiang K. H., Dichinson M. H., *Wing transmission for a micromechanical flying insect*, Proceedings of the 2000 IEEE, International Conference on Robotics & Automation, San Francisco, CA, pp.1509–1516., 2000

[5] Lentink D., van Heijst G. J., Muijres F. T., van Leeuwen J. L., *Vortex interactions with flapping wings and fins can be unpredictable*, Royal Society Biology Letters Vol. 6 pp. 394-397, 2010

[6] Rowley C. W., Colonius T., Murray R. M., *Model reduction for compressible flows using POD and Galerkin projection*, Physica D Vol. 189, pp.115-129, 2004

[7] Erbil T. N., Kasnakoglu C., *Feedback flow control employing local dynamical modeling with wavelets*, Mathematical and Computer Modeling of Dynamical Systems, Vol. 15, pp.493-513, 2009

[8] Knoller R., *Die Gesetze des Luftwiderstandes, Flug- und Motortechnik*,Vol.3, No. 21, pp. 1-7, 1909.

[9] Betz A., *Ein Beitrag zur Erlaerung des Segelfluges, Zeitschrift fur Flugtechnik und Motorluftschiffahrt*, Vol. 3, pp. 269-272, 1912

[10] Jones K. D., Dohring C. M., Platzer M. F., *Experimental and Computational Investigation of the Knoller-Betz Effect*, AIAA Journal, Vol. 36, No. 7, pp. 1240–1246, 1998.

[11] Birnbaum W., Das ebene Problem des Schlagenden Fluegels, Zeitschrift fuer Angewandte Mathematik und Mechanik, Vol. 4, No. 4, pp. 277-292, 1924.

[12] von Karman T., and Burgers J. M., *General Aerodynamic Theory - Perfect Fluids*, Aerodynamic Theory, edited by W. F. Durand, Vol. 2, Julius Springer, Berlin, p.308, 1934

[13] McCroskey W. J., Carr L. W., and McAlister K. W., *Dynamic stall experiments on oscillating airfoils,* AIAA J., Vol. 14, No. 1., pp. 57-63, 1976

[14] Kurtulus D. F., *Ability to forecast unsteady aerodynamic forces of flapping airfoils by artificial neural network*, Neural Computing & Applications Vol. 18 No. 4, pp.359-368, 2008

[15] Deng X., Schenato L., Sastry S. S., *Model identification and attitude control scheme for a micromechanical flying insect*, International Conference on Robotics and Automation, Taiwan, 2003

[16] Kurtulus D. F., David L., Farcy A., Alemdaroglu N., *Aerodynamic characteristics of flapping motion in hover*, Exp fluids Vol. 44, pp.23-36, 2008

[17] Zbikowski R, *On aerodynamic modeling of an insect-like flapping wing in hover for micro air vehicle*, Phil. Trans. R. Soc. Lond A, pp.273-290, 2002

[18] Bai P., Cui E.-J., Zhan H.-L., *Aerodynamic Characteristics, Power Requirements and Camber Effects of the Pitching-Down Flapping Hovering,* Journal of Bionic Engineering, Vol. 6, pp.120-134, 2009

[19] Hua H., Kumar A. G., Abate G., Albertani R., *An experimental investigation on the aerodynamic performances of flexible membrane wings in flapping flight,* Aerospace Science and Technology, Vol. 14, pp.575-586, 2010

[20] Mazaheri K., Ebrahimi A., *Experimental investigation of the effect of chordwise flexibility on the aerodynamics of flapping wings in hovering flight*, Journal of Fluids and Structures, Vol. 26, pp.544–558, 2010

[21] Nguyen T.- T., Byun D., *Two-Dimensional Aerodynamic Models of Insect Flight for Robotic Flapping Wing Mechanisms of Maximum Efficiency*, Journal of Bionic Engineering Vol. 5, pp.1−11, 2008

[22] Amiralaei M. R., Alighanbarin H., Hashemi S. M., *An investigation into the effects of unsteady parameters on the aerodynamics of a low Reynolds number pitching airfoil,* Journal of Fluids and Structures, Vol. 26, pp.979-993, 2010

[23] Günaydınoğlu, E., and Kurtuluş, D.F., *Numerical Investigation of Pure Plunge and Pitch/Plunge Motions at Low Reynolds Numbers,* International Symposium of Light-Weight Unmanned Aerial Vehicle Systems and Subsystems, Ostend, Belgium, 11-13 March 2009

[24] Günaydınoğlu, E., and Kurtuluş, D.F., *Flow Characteristics of Pitching and Plunging SD7003 Airfoil for Different Reduced Frequencies,* Ankara International Aerospace Conference, Ankara, Turkey, 17-19 August 2009

[25] Günaydınoğlu, E., and Kurtuluş, D.F., *Unsteady Aerodynamic Analysis of a Flapping Flat Plate at Low Reynolds Numbers*, First International Workshop on Computational Engineering - Special Topic Fluid-Structure Interaction. Research Unit 493 Fluid-Structure Interaction: Modelling, Simulation, Optimisation, Herrsching, Germany, 12-14 October 2009

[26] Günaydınoğlu, E., and Kurtuluş, D.F., *Çırpan Düz Levha Kesiti Etrafındaki Akışın Sayısal Yöntemlerle İncelenmesi,* 5. Savunma Teknolojileri kongresi SAVTEK 2010, Orta Doğu Teknik Üniversitesi, Ankara, 23-25 Haziran 2010

[27] Tsai B.-J., Fu Y.-C., Design and aerodynamic analysis of a flapping-wing micro aerial vehicle, Aerospace Science and Technology, Vol 13, pp.383-392, 2009

[28] Liang C., Ou K., Premasuthan S., Jameson A., *High-order accurate simulations of unsteady flow past plunging and pitching airfoils*, Computers & Fluids Vol. 40, pp.236–248, 2011

[29] Sarkar S., Venkatraman K., *Numerical simulation of thrust generating flow past a pitching airfoil*, Computers & Fluids, Vol. 35, pp.16–42, 2006

[30] Tuncer, I.H. and Kaya, M., *Optimization of Flapping Airfoils For Maximum Thrust and Propulsive Efficiency*, AIAA Journal, Vol 43, p: 2329-2341, 2005.

[31] Prasad A. K., Hopkins L. M., Kelly J. T., Wexler A. S., *Particle image velocimetry measurements in complex geometries*, Experiments in Fluids, Vol. 29, pp.91-95, 2000

[32] Raffel M., Willert C., Wereley S., Kompenhans J., *Particle image velocimetry: a practical guide*, Springer Science & Business, New York, 1998.

[33] Scarano F., Riethmuller M. L., *Iterative multigrid approach in PIV image processing with discrete window offset*, Experiments in Fluids, Vol. 26, pp.513-523, 1999.

[34] Daichin, Kang, W., Zhao L., *PIV Measurements of the near wake flow of an airfoil above a free surface*, Science Direct Journal of Hydrodynamics Ser.B, Vol. 19, No. 4, pp.482-487, 2007.

[35] Druault P., Challiou C., *Use of proper orthogonal decomposition for reconstructing the 3D in-cylinder mean-flow Field from PIV data*, Comptes Rendus Mécanique, Vol. 335, No.1, pp.42-47, 2007.

[36] Guibert P., Alizon F., Druault P., *Use of proper orthogonal decomposition for time interpolation from PIV data*, Experiments in Fluids, Vol. 39, No. 6, pp.1009-1023, 2005.

[37] Aquino W., *An object-oriented framework for reduced-order models using proper orthogonal decomposition (POD),* Comput. Methods Appl. Mech. Engrg. Vol. 196, pp. 4375–4390, 2007

[38] Sirovich L., *Turbulence and the dynamics of coherent structures*. I – Coherent structures, Quart. Appl. Math., Vol. 45, No. 3, pp.561–571, 1987.

[39] Holmes P., Lumley J. L., Berkooz G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Univ. Press, Cambridge, UK, 1996

[40] Noack B. R., Siev K. A., Morzyinski M., Tamor G., Thiele F., *A hierarchy of low-dimensional models for the transient and post-transient cylinder wake,* Journal of Fluid Mechanics, Vol. 497, pp. 335-363, 2003

[41] Wilcox K., Peraire J., *Balanced model reduction via proper orthogonal decomposition method,* AIAA Journal, Vol. 40, No. 11, pp.2323-2330, 2002.

[42] Tran H. T., Ly H. V., *Modeling and control of physical process using proper orthogonal decomposition*, Mathematical and Computer Modeling, Vol. 33, No.1, pp.223-236, 2001.

[43] Bourguet R., Braza M., Dervieux A., *Reduced-order modeling for unsteady transonic flows around an airfoil,* Phys. Fluids, Vol. 19, pp.1-4, 2007

[44] Liberge E., Hamdouni A., *Reduced order modeling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder,* Journal of Fluids and Structures, Vol. 26, pp. 292–311, 2010

[45] Ravindran S. S., *Control of flow separation over a forward-facing step by model reduction*, Comput. Methods Appl. Mech. Engrg. Vol. 191i pp. 4599–4617, 2002.

[46] Kung, S. Y., *A new identification and model reduction algorithm via singular value decomposition*, Proc. of the 12th Asilomar Conference on Circuits, Systems and Computers, Pacific Grove, pp. 705-714, 1978.

[47] Moonen M., De Moor B., Vandenberghe L., Vandewalle J., *On- and off-line identification of linear state space model*, International Journal of Control. Vol. 49, pp.219-232, 1989.

[48] Moonen, M. and Ramos J., *A subspace algorithm for balanced state space system identification*. ESAT/SISTA report 1991-07, Kath. Universiteit Leuven, Dept. E.E., Belgium, 1991

[49] Van Overschee P., De Moor B., *Subspace algorithms for the stochastic identification problem*, 30th IEEE conference on decision and control, Brighton, U.K., pp. 1321-1326, 1991

[50] Gerhard J., Pastoor M., King R., Noack B. R., Dillmann A., Morzynski M., Tadmor G., *Model-based control of vortex shedding using low-dimensional Galerkin models*, AIAA Paper, 2003

[51] Khalil M., Adhikari S., Sarkar A., *Linear system identification using proper orthogonal decomposition*, Mechanical Systems and Signal Processing, Vol. 21, No. 8, pp.3123-3145, 2007.

[52] Perret L., Delville J., Collin E., *Polynomial identification of POD based low-order dynamical system*, Journal of Turbulence, Vol. 7 No. 17, 2006.

[53] Sjövall P., Abrahamsson T., *Component system identification and state-space model synthesis*, Mechanical Systems and Signal Processing, Vol. 21, pp. 2697–2714, 2007.

[54] Van Overschee P., De Moor B., *N4SID: Subspace Algorithms for the Identification of Combined Deterministic- Stocastic Systems*, Automatica, Vol. 30, No. 1, pp. 75-93, 1994.

[55] Grant I., Owens E., Yan Y., *Particle image velocimetry measurements of the separated flow behind a reward facing step. Experiments in fluids*, Vol. 12, pp.238-244, 1992

[56] Koechner W., *Solid-State Laser Engineering*, 2nd ed. Springer-Verlag, New York, Chap. 2.3, pp. 48–53, 1988

[57] Uzol O., Katz J., Flow *measurement techniques in turbomachinery*, Springer Handbook of Experimental Fluid Mechanics, Section 14, pp.919-958, 2007.

[58] Galletti B., Bottaro A., Bruneau C. H., Iollo A., *Accurate model reduction of transient and forced wakes*, European Journal of Mechanics - B/ Fluids Vol. 26, pp. 354–366, 2007.

[59] Lucthenburg D. M., Noack B. R., Schlegel M., *An introduction to the POD and Galerkin method for fluid flows with analytical examples and MATLAB source codes*, Berlin Institute of Technology, Technical Report, 2009.

[60] Kostas J., Soria J., Chong M., *Particle image velocimetry measurements of a backward- facing step flow*, Experiments in Fluids, Vol. 33, pp. 838-853, 2002.

[61] Adams R., *Radial Decomposition of Discs and Spheres*, Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing, Vol. 55, No. 5, pp. 325–332, September 1993.

[62] Jones R., Soille P., *Periodic lines: Definition, cascades, and application to granulometrie*, Pattern Recognition Letters, Vol. 17, pp. 1057–1063, 1996.

[63] Prajna S., *POD Model Reduction with Stability Guarantee*, Decision and Control, 42nd IEEE Conference, Vol.5, pp. 5254 - 5258, 2003.

[64] Chatterjee A., *An introduction to the proper orthogonal decomposition*, Current Science, Vol. 78, No.7, pp.808-817, 2000.

[65] Holmes P., Lumley J. L., Berkooz G., *Turbulence, coherent structures, dynamical system, and symmetry*, Cambridge University Press, 1996.

[66] Ogata K., *Discrete Time Control Systems*, 2nd Edition, Prentice Hall, 1995

[67] Ljung L., *System Identification: Theory for the User*, PTR Prentice Hall, Upper Saddle River, NJ, 1999.

[68] Van Overschee P., De Moor B., *Subspace identification for linear systems: Theory-Implementation-Applications*, Kluwer Academic Publishers, Dordrecht, 1996.

[69] Larimore W. E., *Statistical optimality and canonical variate analysis system identification*, Signal Process. Vol. 52, pp. 131–144, 1996

APPENDIX A

MATLAB CODES

main.m

```
addpath('C:\Users\SURU\Desktop\29_12_2010\naca\30\p');
%% Construction of X and Y Components of Velocity Matrices
UX= [];
UY= [];
pvxAll = {}; pvyAll = {}; dxAll = {}; dyAll = {};
for i=8:258
FN=[num2str(i),'.txt'];
NewData=importdata(FN);
E = NewData;
if i==8
pivMax = max(E);
pivXmax = pivMax(1);
pivYmax = pivMax(2);
pivMin = min(E);
pivXmin = pivMin(1);
pivYmin = pivMin(2);
NG = 300;
xil= linspace(pivXmin,pivXmax,NG);
yil=linspace(pivYmin,pivYmax,NG);
dxil=xil(2)-xil(1);
```

```
dyil=yil(2)-yil(1);
 [xi,yi] = meshgrid(xil,yil);
end
ui = griddata(E(:,1),E(:,2),E(:,3),xi,yi);
vi = griddata(E(:,1),E(:,2),E(:,4),xi,yi);
%% Image Processing
if i<10
imname=['NACA 0012_30pitch.2dpzb581.00000' num2str(i),'.tif'];
elseif i<100
imname=['NACA 0012_30pitch.2dpzb581.0000' num2str(i),'.tif'];
elseif i<1000
imname=['NACA 0012_30pitch.2dpzb581.000' num2str(i),'.tif'];
end
[pvx,pvy,dx,dy] = allperimeters(imname,pivXmin,pivXmax,pivYmin,pivYmax,i);
pvxAll{i} = pvx; pvyAll{i} = pvy; dxAll{i} = dx; dyAll{i} = dy;
W1 = 1250*dx;
W2 = (1200-588)*dy;
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
%% Velocities Inside Airfoil
omega = 0.2*pi^2/6*cos(0.2*pi*0.2*(i-8));
r = sqrt((xi-W1).^2+(yi-W2).^2)/1000;
teta = atan2(yi-W2,xi-W1);
ui2 = ui;
ui2(ind1) = omega*r(ind1).*sin(teta(ind1));
vi2 = vi;
vi2(ind1) = -omega*r(ind1).*cos(teta(ind1));
UX =[UX ui2(:)];
UY =[UY vi2(:)];
end
save kanatKenarVeDxDy pvxAll pvyAll dxAll dyAll
%% Proper Orthogonal Decomposition
```

```
[phiX,phiY,C,UXmean,UYmean,zk,VXhat,VYhat]=POD (UX,UY,0);

%% Ident Input and Output Values
ind=zeros(size(zk(1,:)'));
outd = zk(1:100,:)';
%% Energy
[evectorC evalueC] = svd(C);
energy = cumsum(diag(evalueC)/sum(diag(evalueC))*100);
inde=find(energy>95);
inde(1)
```

allperimeters.m

```
function [kanatx,kanaty,dx,dy] =
allperimeters(filename,pivXmin,pivXmax,pivYmin,pivYmax,i)
load ('kanat_data.mat');
addpath 'C:\Users\SURU\Desktop\29_12_2010\naca\30\p'
veritabani = 'C:\Users\SURU\Desktop\29_12_2010\naca\30\p';
veritabaniDosyalari = diziniListele(veritabani, '*.tif');
goruntuSayisi = length(veritabaniDosyalari);
resimler=cell(goruntuSayisi,1);
goruntuno=i;
A=veritabaniDosyalari{i};
A=imread(A);
[sizeX,sizeY]=size(A);
a=mean(A);
a=mean(a);
ws=round(a);
N=double(A);
M=double(A);
st=0;
for i=1:sizeX
```

```
for j=1:sizeY
st=st+(M(i,j)-a)^2;
end
end
st=sqrt(st)/sizeX/sizeY;
C=st;
IM=mat2gray(A);
mIM=imfilter(IM,fspecial('average',ws),'replicate');
sIM=mIM-IM-C;
BW=im2bw(sIM,0);
BW=imcomplement(BW);
se = strel('line',2,90);
BW=imclose(BW,se);
se = strel('line',2,0);
BW=imclose(BW,se);
se = strel('disk',2);
BW=imclose(BW,se);
se = strel('line',2,0);
BW=imclose(BW,se);
se = strel('line',2,90);
BW=imclose(BW,se);
BW=imcomplement(BW);
G=ones(1000,1600);
G(1200,1600)=0;
BW=G.*BW;
BW = bwareaopen(BW, 25000);
BW=imfill(BW,'holes');
se = strel('line',7,0);
BW=imclose(BW,se);
se = strel('line',7,90);
BW=imclose(BW,se);
se = strel('disk',2);
```

```
BW=imclose(BW,se);
BW=bwperim(BW,4);
Q=BW;
resimler{goruntuno,1}=BW;
[r,c]=find(resimler{goruntuno,1});
[sizerx,sizery]=size(r);
xyler{goruntuno,1}=[r,c];
if cikti{goruntuno,1} > -2.65 && cikti{goruntuno,1} <= 1.5
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-8;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-8;
end
if cikti{goruntuno,1} > -5.83 && cikti{goruntuno,1} <= -2.65
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-10;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-6;
end
if cikti{goruntuno,1} > -9.41 && cikti{goruntuno,1} <= -5.83
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-6;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-6;
end
if cikti{goruntuno,1} > -12.69 && cikti{goruntuno,1} <= -9.41
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-0;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-6;
end
if cikti{goruntuno,1} > -15.53 && cikti{goruntuno,1} <= -12.69
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-4;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-3;
end
if cikti{goruntuno,1} > -18.46 && cikti{goruntuno,1} <= -15.53
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-0;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-3;
end
if cikti{goruntuno,1} > -20.77 && cikti{goruntuno,1} <= -18.46
```

```
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-0;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-1;
end
if cikti{goruntuno,1} > -23.05 && cikti{goruntuno,1} <= -20.77
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-0;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))+1;
end
if cikti{goruntuno,1} > -24.63 && cikti{goruntuno,1} <= -23.05
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))+2;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))+4;
end
if cikti{goruntuno,1} > -25.94 && cikti{goruntuno,1} <= -24.63
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))+3;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))+6;
end
if cikti{goruntuno,1} > -26.74 && cikti{goruntuno,1} <= -25.94
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))+5;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))+5;
end
if cikti{goruntuno,1} < -25.94
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))+4;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))+5;
end
if cikti{goruntuno,1} > 1.5 && cikti{goruntuno,1} <= 3.52
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-10;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-7;
end
if cikti{goruntuno,1} > 3.52 && cikti{goruntuno,1} <= 7.16
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-12;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-7;
end
if cikti{goruntuno,1} > 7.16 && cikti{goruntuno,1} <= 11.17
```

```
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-12;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-8;
end
if cikti{goruntuno,1} > 11.17 && cikti{goruntuno,1} <= 13.96
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-13;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-10;
end
if cikti{goruntuno,1} > 11.17 && cikti{goruntuno,1} <= 13.96
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-13;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-11;
end
if cikti{goruntuno,1} > 13.96 && cikti{goruntuno,1} <= 16.92
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-13;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-12;
end
if cikti{goruntuno,1} > 16.92 && cikti{goruntuno,1} <= 20.17
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-12;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-11;
end
if cikti{goruntuno,1} > 20.17 && cikti{goruntuno,1} <= 22.40
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-13;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-13;
end
if cikti{goruntuno,1} > 22.40 && cikti{goruntuno,1} <= 24.54
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-14;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-12;
end
if cikti{goruntuno,1} > 24.54 && cikti{goruntuno,1} <= 25.77
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-11;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-11;
end
if cikti{goruntuno,1} > 25.77 && cikti{goruntuno,1} <= 27.26
```

```
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))-14;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))-12;
end
if cikti{goruntuno,1} > 27.26
yenixyler{goruntuno,1}(:,1)=(xyler{goruntuno,1}(:,1))+14;
yenixyler{goruntuno,1}(:,2)=(xyler{goruntuno,1}(:,2))+11;
end
Z=yenixyler{goruntuno,1};
kanatxi= Z(:,1);
kanatyi= Z(:,2);
[kanatxi,kanatyi]=find(Q);
indc= convhull(kanatxi,kanatyi);
kanatxic= kanatxi(indc);
kanatyic= kanatyi(indc);
dx=(pivXmax-pivXmin)/sizeY;
dy=(pivYmax-pivYmin)/sizeX;
kanaty=(sizeX-kanatxic)*dx;
kanatx=kanatyic*dy;
```

POD.m

```
function [phiX,phiY,C,UXmean,UYmean,zk,VXhat,VYhat]=POD(UX,UY,ops)
N = size(UX,2);
N = size(UY,2);
row = size(UX,1);
disp('  ------------------ RESULTS AND INSTRUCTIONS ------------------')
disp(['  Number of Snapshots is                    : ' int2str(N)])
if ops == 0
UXmean = sum(UX,2)/N;
UYmean = sum(UY,2)/N;
else
UXmean = UX(:,N);
```

```
UYmean = UY(:,N);
end
% New Snapshot Ensembles
VX = zeros(row,N);
VY = zeros(row,N);
for i=1:N
VX(:,i) = UX(:,i)-UXmean;
VY(:,i) = UY(:,i)-UYmean;
end
% Calculate the covariance matrix C
a = waitbar(0,'Calculating covariance matrix C...');
C1 = zeros(row,1);
C2 = zeros(row,1);
C3 = zeros(row,1);
C4 = zeros(row,1);
for i=1:N
for j=i:N
C1 = VX(:,i);
C2 = VX(:,j);
C3 = VY(:,i);
C4 = VY(:,j);
C(i,j) = sum(sum(VX(:,i).*VX(:,j)+VY(:,i).*VY(:,j)))/N;
C(j,i) = C(i,j);
waitbar((i*N-N+j)/N^2);
end
end
close(a);
% Calculate eigenvalues and eigenvectors of C
a = waitbar(0,'Calculating eigenvalues and eigenvectors of C...');
[evectorC evalueC] = svd(C);
rankC = rank(C);
for i=1:rankC
```

```
evectorC(:,i) = evectorC(:,i)/(evalueC(i,i)^0.5);
waitbar(i/rankC);
end
close(a);
% Calculate the POD basis
a = waitbar(0,'Calculating POD Basis...');
phiX = zeros(row,N);
phiY = zeros(row,N);
for i=1:rank(C)
totalX = zeros(row,1);
totalY = zeros(row,1);
for j=1:N
totalX = totalX + evectorC(j,i)*VX(:,j);
totalY = totalY + evectorC(j,i)*VY(:,j);
waitbar((i*N-N+j)/rankC/N);
end
phiX(:,i) = totalX;
phiY(:,i) = totalY;
end
close(a);
% Time coefficients
zk = [];
for i=1:rank(C)
for j=1:N
zk(i,j)= sum(sum(VX(:,j).*phiX(:,i)+VY(:,j).*phiY(:,i)))/N;
end
end
% Reconstruction with POD time coefficients
VXhat = zeros(size(VX));
VYhat = zeros(size(VY));
for j=1:N
for i=1:100
```

```
VXhat(:,j)= VXhat(:,j) + zk(i,j).*phiX(:,i);
VYhat(:,j)= VYhat(:,j) + zk(i,j).*phiY(:,i);
end
end
```

reconstruction.m

```
% Reconstruction with SI time coefficients
AA=n4s40.A;
BB=n4s40.B;
CC=n4s40.C;
DD=n4s40.D;
x0=n4s40.x0;
T=0.2;
sys= ss(AA,BB,CC,DD,T);
[zki,ti] = lsim(sys,ind,0:0.2:250*0.2,x0);
zki=zki';
VXhat2 = zeros(size(UX));
VYhat2 = zeros(size(UY));
for j=1:size(zki,2)
for i=1:size(zki,1)
VXhat2(:,j)= VXhat2(:,j) + zki(i,j).*phiX(:,i);
VYhat2(:,j)= VYhat2(:,j) + zki(i,j).*phiY(:,i);
end
end
```

plot.m

```
load kanatKenarVeDxDy.mat;
load nacaws.mat;
a=[50; 54; 64; 69; 74; 78; 89; 95; 100];
%% PIV görüntüleri X hızları
```

```
for i=1:9
subplot(3,3,i)
pvx = pvxAll{a(i)+7}; pvy = pvyAll{a(i)+7}; dx = dxAll{a(i)+7}; dy =
dyAll{a(i)+7};
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
phiui = reshape(UX(:,a(i,:)),size(xi));
phiui2 = phiui;
phiui2(ind1) = NaN;
surf(xi,yi,phiui2); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
v=caxis;
title('t= 10 s');
elseif i==2
title('t= 10.8 s');
caxis(v);
elseif i==3
title('t= 12.8 s');
caxis(v);
elseif i==4
ylabel('y(mm)');
xlabel('m / s');
title('t= 13.8 s');
caxis(v);
elseif i==5
title('t= 14.8 s');
caxis(v);
elseif i==6
title('t= 15.6 s');
caxis(v);
```

```
elseif i==7
title('t= 17.8 s');
caxis(v);
elseif i==8
xlabel('x(mm)');
title('t= 19 s');
caxis(v);
elseif i==9
title('t= 20 s');
caxis(v);
end
end
%% PIV görüntüleri Y hızları
for i=1:9
subplot(3,3,i)
pvx = pvxAll{a(i)+7}; pvy = pvyAll{a(i)+7}; dx = dxAll{a(i)+7}; dy =
dyAll{a(i)+7};
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
phiui = reshape(UY(:,a(i,:)),size(xi));
phiui2 = phiui;
phiui2(ind1) = NaN;
surf(xi,yi,phiui2); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
v=caxis;
title('t= 10 s');
elseif i==2
title('t= 10.8 s');
caxis(v);
elseif i==3
```

```
title('t= 12.8 s');
caxis(v);
elseif i==4
ylabel('y(mm)');
xlabel('m / s');
title('t= 13.8 s');
caxis(v);
elseif i==5
title('t= 14.8 s');
caxis(v);
elseif i==6
title('t= 15.6 s');
caxis(v);
elseif i==7
title('t= 17.8 s');
caxis(v);
elseif i==8
xlabel('x(mm)');
title('t= 19 s');
caxis(v);
elseif i==9
title('t= 20 s');
caxis(v);
end
end
%% ilk 10 pod modu X için
for i=1:9
subplot(3,3,i)
phiui = reshape(phiX(:,i),size(xi));
surf(xi,yi,phiui); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
```

```
if i==1
vv=caxis;
title('First POD Mode');
elseif i==2
title('Second POD Mode');
caxis(vv);
elseif i==3
title('Third POD Mode');
caxis(vv);
elseif i==4
title('Fourth POD Mode');
ylabel('y (mm)');
caxis(vv);
elseif i==5
title('Fifth POD Mode');
caxis(vv);
elseif i==6
title('Sixth POD Mode');
caxis(vv);
elseif i==7
title('Seventh POD Mode');
caxis(vv);
elseif i==8
xlabel('x(mm)');
title('Eighth POD Mode');
caxis(vv);
elseif i==9
title('Ninth POD Mode');
caxis(vv);
end
end
%% ilk 10 pod modu Y için
```

```
figure;
for i=1:9
subplot(3,3,i)
phiui = reshape(phiY(:,i),size(xi));
surf(xi,yi,phiui); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
vv=caxis;
title('First POD Mode');
elseif i==2
title('Second POD Mode');
caxis(vv);
elseif i==3
title('Third POD Mode');
caxis(vv);
elseif i==4
title('Fourth POD Mode');
ylabel('y (mm)');
caxis(vv);
elseif i==5
title('Fifth POD Mode');
caxis(vv);
elseif i==6
title('Sixth POD Mode');
caxis(vv);
elseif i==7
title('Seventh POD Mode');
caxis(vv);
elseif i==8
xlabel('x(mm)');
title('Eighth POD Mode');
```

```
caxis(vv);
elseif i==9
title('Ninth POD Mode');
caxis(vv);
end
end
%% POD zaman katsayıları ile geriçatılama X
figure
for i=1:9
subplot(3,3,i)
pvx = pvxAll{a(i)+7}; pvy = pvyAll{a(i)+7}; dx = dxAll{a(i)+7}; dy =
dyAll{a(i)+7};
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
phiui = reshape((VXhat(:,a(i,:))+UXmean),size(xi));
phiui2 = phiui;
phiui2(ind1) = NaN;
surf(xi,yi,phiui2); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
caxis(v);
title('t= 10 s');
elseif i==2
title('t= 10.8 s');
caxis(v);
elseif i==3
title('t= 12.8 s');
caxis(v);
elseif i==4
ylabel('y(mm)');
xlabel('m / s');
```

```
title('t= 13.8 s');
caxis(v);
elseif i==5
title('t= 14.8 s');
caxis(v);
elseif i==6
title('t= 15.6 s');
caxis(v);
elseif i==7
title('t= 17.8 s');
caxis(v);
elseif i==8
xlabel('x(mm)');
title('t= 19 s');
caxis(v);
elseif i==9
title('t= 20 s');
caxis(v);
end
end
%% POD zaman katsayıları ile geriçatılama Y
figure
for i=1:9
subplot(3,3,i)
pvx = pvxAll{a(i)+7}; pvy = pvyAll{a(i)+7}; dx = dxAll{a(i)+7}; dy =
dyAll{a(i)+7};
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
phiui = reshape((VYhat(:,a(i,:))+UYmean),size(xi));
phiui2 = phiui;
phiui2(ind1) = NaN;
surf(xi,yi,phiui2); view(2); shading flat;
```

```
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
caxis(v);
title('t= 10 s');
elseif i==2
title('t= 10.8 s');
caxis(v);
elseif i==3
title('t= 12.8 s');
caxis(v);
elseif i==4
ylabel('y(mm)');
xlabel('m / s');
title('t= 13.8 s');
caxis(v);
elseif i==5
title('t= 14.8 s');
caxis(v);
elseif i==6
title('t= 15.6 s');
caxis(v);
elseif i==7
title('t= 17.8 s');
caxis(v);
elseif i==8
xlabel('x(mm)');
title('t= 19 s');
caxis(v);
elseif i==9
title('t= 20 s');
caxis(v);
```

```
end
end
%% system identification ile geriçatılama X
figure;
for i=1:9
subplot(3,3,i)
pvx = pvxAll{a(i)+7}; pvy = pvyAll{a(i)+7}; dx = dxAll{a(i)+7}; dy =
dyAll{a(i)+7};
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
phiui = reshape((VXhat2(:,a(i,:))+UXmean),size(xi));
phiui2 = phiui;
phiui2(ind1) = NaN;
surf(xi,yi,phiui2); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
caxis(v);
title('t= 10 s');
elseif i==2
title('t= 10.8 s');
caxis(v);
elseif i==3
title('t= 12.8 s');
caxis(v);
elseif i==4
ylabel('y(mm)');
xlabel('m / s');
title('t= 13.8 s');
caxis(v);
elseif i==5
title('t= 14.8 s');
```

```
caxis(v);
elseif i==6
title('t= 15.6 s');
caxis(v);
elseif i==7
title('t= 17.8 s');
caxis(v);
elseif i==8
xlabel('x(mm)');
title('t= 19 s');
caxis(v);
elseif i==9
title('t= 20 s');
caxis(v);
end
end
%% system identification ile geriçatılama Y
figure
for i=1:9
subplot(3,3,i)
pvx = pvxAll{a(i)+7}; pvy = pvyAll{a(i)+7}; dx = dxAll{a(i)+7}; dy =
dyAll{a(i)+7};
IN = inpolygon(xi,yi,pvx,pvy);
ind1 = find(IN);
phiui = reshape((VYhat2(:,a(i,:))+UYmean),size(xi));
phiui2 = phiui;
phiui2(ind1) = NaN;
surf(xi,yi,phiui2); view(2); shading flat;
axis([pivXmin pivXmax pivYmin pivYmax]);
grid off;
if i==1
caxis(v);
```

```
title('t= 10 s');
elseif i==2
title('t= 10.8 s');
caxis(v);
elseif i==3
title('t= 12.8 s');
caxis(v);
elseif i==4
ylabel('y(mm)');
xlabel('m / s');
title('t= 13.8 s');
caxis(v);
elseif i==5
title('t= 14.8 s');
caxis(v);
elseif i==6
title('t= 15.6 s');
caxis(v);
elseif i==7
title('t= 17.8 s');
caxis(v);
elseif i==8
xlabel('x(mm)');
title('t= 19 s');
caxis(v);
elseif i==9
title('t= 20 s');
caxis(v);
end
end
%% ilk 10 POD moduna karşılık gelen zaman katsayılarının grafiği
for i=1:9
```

```
subplot(3,3,i)
p=plot(x,y(:,i));
set(p,'Color','red','LineWidth',1);
hold on
p=plot(x,zk(i,:));
set(p,'Color','blue','LineWidth',1);
if i==1
title('First Channel');
elseif i==2
title('Second Channel');
elseif i==3
title('third Channel');
elseif i==4
title('Fourth Channel');
elseif i==5
title('Fifth Channel');
elseif i==6
title('Sixth Channel');
elseif i==7
title('Seventh Channel');
elseif i==8
title('Eighth Channel');
xlabel('t(s)');
elseif i==9
title('Ninth Channel');
end
end
%% Energy Level
plot(energy,'LineWidth',3); grid; axis tight;
xlabel('Number of Modes');
ylabel('Percent Energy Level');
```