الجمهورية الجزائرية الديمقراطية الشعبية

LA REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida 1

Institut d'Aéronautique et des Études Spatiales

Département Construction Aéronautique

Mémoire de fin d'études

En vue de l'obtention du diplôme de

Master en Aéronautique

Option : **Avionique**

THEME

Adaptive RBF Neural Network with sliding mode controller for a nonlinear flexible wing aircraft

Proposé et dirigé par :

Dr. Dilmi Smain

Réalisé par :

Mr Lemita Mansour

Promotion : 2022 / 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF COMMITTEE APPROVAL

Dr. , Chair

Institute of Aeronautics and Spaces studies

Dr.

Institute of Aeronautics and spaces studies

Dr.

Institute of Aeronautics and Spaces studies

Approved by:

Dr.

DEDICATION

I dedicate this research to my unwavering pillars of support - my loving family, the guiding force of my life, and the source of endless encouragement. Through every step of this journey, your love, understanding, and belief in me have been an invaluable source of strength.

I also extend my heartfelt dedication to the divine presence that has guided me on this path. My faith has provided solace, inspiration, and the unwavering belief that I am part of a greater purpose. It is with gratitude and reverence that I acknowledge the guiding light that has illuminated my way.

Lastly, I dedicate this work to myself - a testament to my unwavering determination, resilience, and relentless pursuit of knowledge. It is a celebration of the countless hours poured into research, the challenges overcome, and the personal growth achieved. This dedication is a reminder that with passion and dedication, anything is possible.

May this research contribute to the greater body of knowledge, inspire others, and pave the way for future discoveries.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of this research project. Without their support, guidance, and valuable input, this study would not have been possible.

First and foremost, I am deeply thankful to my supervisor, **Mr. Smain Dilmi**, for their unwavering guidance, expertise, and continuous encouragement throughout the research process. Their insightful feedback and constructive criticism have immensely shaped the direction of this study.

I am grateful to Institute of aeronautics and spatial studies for providing the necessary resources and facilities that have enabled me to carry out this research. The support of the Construction Department and the library staff is greatly appreciated.

Last but not least, I would like to express my deepest gratitude to my family for their unwavering love, understanding, and support throughout this endeavor. Their encouragement and belief in me have been a constant source of motivation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Nomenclature:

Lettres grecques :

Symbole	**Unité**
$\boldsymbol{\alpha}$: Angle of attack.	$[rad]$
$\boldsymbol{\beta}$: Trailing edge control surface deflection angle.	$[rad]$
$\boldsymbol{\gamma}$: Leading edge control surface deflection angle.	$[rad]$
$\boldsymbol{\rho}$: Air density.	$[Kg/m\ 3\ ]$
µ: Valeur singulière structurée.	[-]

Lettres Latines :

Symbole	**Unité**
$\boldsymbol{L}$: lift force.	[N]
$\boldsymbol{M}$: The aerodynamic moment.	[kg.m²]
$\boldsymbol{T}$: The kinetic energy.	[J]
$\boldsymbol{U}$: Potential energy.	[J]
$\boldsymbol{Q_j}$: The component of the external forces which work according to degree of freedom.	[N]
$\boldsymbol{U_J}$: The displacement field.	[m]
$\boldsymbol{I_{cg}}$: The moment of inertia around the center of gravity.	[kg.m²]
$\boldsymbol{m}$: The mass of the wing.	[kg]
$\boldsymbol{V_{cg}}$: The speed of the center of gravity.	[m/s]
$\boldsymbol{V_{ea}}$: Elastic center velocity.	[m/s]
$\boldsymbol{\delta_h}$: Virtual vertical displacement.	[m]
$\boldsymbol{\delta_\alpha}$: The virtual pitch displacement.	[rad]
$\boldsymbol{\delta_{wp}}$: Virtual work due to lift force.	[N.m]
$\boldsymbol{\delta_{wm}}$: The virtual work due to the moment M.	[N.m]
$\boldsymbol{\delta_w}$: The total virtual work.	[N.m]
$\boldsymbol{C_h}$: The damping coefficient for the vertical movement.	[Kg/s]
$\boldsymbol{C_\alpha}$: The damping coefficient for pitch.	[Kg.m^2 /s]
$\boldsymbol{h}$: Vertical displacement.	[m]
$\boldsymbol{b}$: Medium airfoil half chord.	[m]
$\boldsymbol{x_\alpha}$: The Static Unbalance Parameter.	[-]
$\boldsymbol{m_T}$: The total mass of the wing and its supporting structure.	[kg]
$\boldsymbol{m_w}$: The mass of the wing only.	[Kg]
$\boldsymbol{k_h}$: The structural stiffness coefficient for vertical movement.	[N/m]

$\boldsymbol{k_\alpha}$: The structural stiffness coefficient for the pitching motion. [Nm/rad]

$\boldsymbol{I_{ea}}$: Moment of inertia around the elastic axis. [Kg $.m^2$]

$\boldsymbol{c_{l\alpha}}$: The derivative of lift coefficient compared to α. [-]

$\boldsymbol{c_{l\beta}}$: The lift coefficient derivative compared to β. [-]

$\boldsymbol{c_{l\gamma}}$: The lift coefficient derivative compared to γ. [-]

$\boldsymbol{c_{m\alpha}}$: The moment coefficient derivative compared to α. [-]

$\boldsymbol{c_{m\beta}}$: The moment coefficient derivative compared to β. [-]

$\boldsymbol{c_{m\gamma}}$: The moment coefficient derivative compared to γ. [-]

$\boldsymbol{S}$: The area of the wing section. [m]

$\boldsymbol{V}$: Flow velocity. [m/s]

$\boldsymbol{t}$: Time. [s]

$\boldsymbol{t_s}$: Response time. [s]

$\boldsymbol{Q_h}$: The generalized force according to h. [N]

$\boldsymbol{Q_\alpha}$: The generalized force according to α. [N]

$\boldsymbol{t_f}$: The terminal time. [s]

$\boldsymbol{q_{nom}}$: Rated dynamic pressure. [Pa]

Les matrices et vecteurs :

$\boldsymbol{x} \in R^n$: Vector that represents the n state variables (state vector).

$\boldsymbol{y} \in R^q$: Vector representing the q outputs.

$\boldsymbol{u} \in R^p$: Vector representing the p commands.

$\boldsymbol{A} \in R^{n \times n}$: Linear system state matrix.

$\boldsymbol{B} \in R^{n \times p}$: Control matrix of a linear system.

$\boldsymbol{C} \in R^{q \times n}$: Observation matrix of a linear system.

$\boldsymbol{D} \in R^{q \times p}$: Forward action matrix of a linear system.

I: the inertia matrix

F: the damping matrix

E: the stiffness matrix

ABSTRACT

Our research will concern the study of an adaptive neural network sliding mode controller using radial basic function to reduce the aeroelastic instabilities of a nonlinear aeroelastic system. The strategy of the designed controller is to provide the ability to damp rapidly the high amplitude oscillation that occurred in the system and to enhance the flight conditions beyond the open-loop critical flutter speed. Also, the proposed controller can be used to estimate the model and its nonlinear dynamics. The two-degree-of-freedom nonlinear aeroelastic system describes the pitch and plunge motion of the aircraft wing section equipped with leading- and trailing edge control surfaces. Furthermore, the selected model considers the quasi-steady aerodynamic model and the structural stiffness nonlinearities. The objective of the simulations is to show the behavior of the system in open and closed-loop and to demonstrate the ability of the controller to reduce the oscillation amplitude in the subcritical flight speed range and to derive the state trajectories to the origin even in the presence of the disturbances and the uncertainties.

الملخص:

يهتم هذا البحث بدراسة وحدة تحكم نمط انزلاقي للشبكة العصبية التكيفية باستخدام الوظيفة الأساسية الشعاعية للتقليل من عدم الاستقرار الهوائي للنظام غير الخطي المرن. تتمثل استراتيجية وحدة التحكم المصممة في توفير القدرة على التخميد السريع للتذبذب عالي السعة الذي حدث في النظام وتحسين ظروف الطيران بما يتجاوز سرعة الرفرفة الحرجة ذات الحلقة المفتوحة. أيضًا، يمكن استخدام وحدة التحكم المقترحة لتقدير النموذج وديناميكياته غير الخطية. يصف النظام غير الخطي بدرجتين من الحرية حركة الانحدار والاندفاع لجزء من جناح الطائرة المجهز بأسطح تحكم في الحواف الأمامية والخلفية. علاوة على ذلك، فإن النموذج المختار يأخذ في الاعتبار النموذج الديناميكي الهوائي شبه الثابت والصلابة الهيكلية الغير خطية. الهدف من عمليات المحاكاة هو إظهار سلوك النظام في حلقة مفتوحة ومغلقة وإثبات قدرة وحدة التحكم على تقليل سعة التذبذب في نطاق سرعة الطيران غير الحرج واعادة حالة المسار إلى الأصل حتى في وجود الاضطرابات.

Résumé :

Notre recherche portera sur l'étude d'un contrôleur de mode glissant de réseau neuronal adaptatif utilisant la fonction de base radiale pour eviter les instabilités aéroélastiques d'un système aéroélastique non linéaire. La stratégie du contrôleur conçu est de fournir la capacité d'amortir rapidement l'oscillation de haute amplitude qui s'est produite dans le système et

d'améliorer les conditions de vol au-delà de la vitesse de flottement critique en boucle ouverte. De plus, le contrôleur proposé peut être utilisé pour estimer le modèle et sa dynamique non linéaire. Le système aéroélastique non linéaire à deux degrés de liberté décrit le mouvement de tangage et de plongée de la section d'aile de l'avion équipée de gouvernes de bord d'attaque et de fuite. De plus, le modèle sélectionné prend en compte le modèle aérodynamique quasi stationnaire et les non-linéarités de rigidité structurelle. L'objectif des simulations est de montrer le comportement du système en boucle ouverte et fermée et de démontrer la capacité du contrôleur à réduire l'amplitude d'oscillation dans la plage de vitesse de vol sous-critique et à dériver les trajectoires d'état jusqu'à l'origine même dans la présence des perturbations et des incertitudes.

General introduction:

The aviation industry has gone through many stages over the decades to develop it and reach the highest level of safety and control of the aircraft during the flight. This is what engineers have worked in this field to achieve through a series of research and experiments.

As a result of these experiments, engineers have found that the use of flexible wings that made from composite materials and polystyrene gives the wing freedom to bend and twist while exposing it to aerodynamic forces during flight, this makes the wing resistant to fracture while exposing it to these forces while maintaining the general shape of the wing and its characteristics, and also flexible wings can reduce drag and increase lift which can develops the level of safety, control and stability of the aircraft during the flight.

Aeroelastic is a field that studies the relationship between the aerodynamic forces and the airframe and the exact effect of these forces on the structure of the airplane and how it responds to them and the consequences resulting from these responses. This field is very important for engineers while designing the airframe, especially the wings, and using sophisticated simulation systems to study these designs and the effect of the aerodynamic forces applied on them, which can help the scientists to build a perception about the design characteristics allowing for creating a stable, safe and efficient design.

During the study of aeroelastic by engineers one of the most important things on which this study depends is the flutter of the wing, which is a phenomenon caused when the natural vibration frequencies of a flexible wing match the frequency of the airflow over it, which can start the oscillations of the wing or “flutter”. This phenomenon is very serious especially for flexible wings because at the beginning of flexible wing vibration and due to an external force, it can creates a feedback loop leading to a rapid and potentially destructive oscillation, resulting in a rapid and destructive fluctuation that ultimately results in a loss of total or partial control of the aircraft with a serious negative effect on the airframe.

The cost of structural damage caused by the flutter can be significant and this is what makes it a major problem in the development of aviation. This is why engineers have worked to find serious solutions to eliminate this phenomenon by developing many solutions based on many studies using the latest computer simulations systems and wind tunnel tests to identify and analyze potential flutter issues before an aircraft is ever built, and this has resulted in many solutions such as ; increasing the stiffness of the wing structure by adding more structural supports or increasing the thickness of the wing, or installing damping systems such as hydraulic or elastomeric dampers, to absorb the vibrations caused by flutter,

or by improving the aerodynamics of the wing which can also help to prevent flutter by modifying the wing's shape or adding features, such as winglets or vortex generators, that improve its stability and reduce drag.

One of the most controllers used by engineers to improve the control of the airplane during the flight is the adaptive sliding mode controller, which is commonly used in the design of control systems that require a rapid and powerful response such as aircraft movement and speed control systems. This type of controller has proved successful in the field of control because of its great ability to handle non-linear and random systems. Another advantage of sliding mode control is its durability in the face of disturbances and uncertainties, because this system is designed to respond quickly and aggressively to any deviations from the required behavior, ensuring the system remains stable and functions as intended. So based on what has been mentioned, the problematic that our research discussing is:

Is it possible to use the adaptive sliding mode controller's properties to eliminate the problem of the flutter phenomenon for a flexible wing?

The objectives that we want to achieve after this research are:

- Discussing the possibility of using an adaptive sliding mode system to control the flutter phenomenon of flexible wings.
- Using the simulation program to find out the effectiveness of our sliding mode system in controlling the flutter phenomenon.
- Comparison of the results obtained in the case of open and closed loop.

Hypotheses:

- The stability will be increased, and flutter will be suppressed, by using an adaptive sliding mode controller to manage flutter and aeroelasticity in a flexible wing. Due to the adaptive nature of the controller, it can adjust and react to the changing dynamics of the wing, effectively reducing the negative impacts of aeroelasticity and averting the potentially dangerous occurrence known as flutter.
- A sliding mode adaptive controller can offer robustness to uncertainties and disturbances that affect the flexible wing. Variations in aerodynamic conditions, structural characteristics, and outside disturbances can all affect aeroelastic systems. Even in the midst of such disturbances, the adaptive sliding mode controller can adjust to these uncertainties, maintaining accurate and dependable control.

Methodology of work:

This research contains two parts, the first one presents the theorical part of this research and it contains three chapters:

Chapter I Throughout this chapter we will delve into the intricacies of aeroelasticity, the flutter phenomenon and its proposed remedy (aeroservoelasticity), and conclude with an overview of the sliding mode controller.

Chapter II My undertaking will focus on developing aeroelastic equations that dictate how air systems behave using Lagrange equations. To accomplish this, I will compose linear equations in two degrees of liberty. Following this, I will create a digital application to simulate the dynamic behavior of TAMU WING II – a wing model.

Chapter III Our objective herein is to provide an extensive analysis on the adaptive sliding mode control system (RBF neural network and sliding mode control) and illustrate its ability to mitigate flutter's effects. Through an elaborate discussion on this control technique, we aim to exhibit its practical utility in real-life scenarios.

The second part of this research concerns the practical part, when we will use the MATLAB program to simulate, study and testing our theories and we will take the wing TAMU II as our model in these experiments. This part is represented by the fourth chapter:

Chapter VI Our upcoming chapter will unveil the results of our thorough experimentation with a comprehensive analysis and accompanying explication. Furthermore, we shall conclude our study by addressing the underlying problem and providing a fitting response.

CHAPTER I:

Generalities

Introduction:

Aircraft constructions, due to their great flexibility, are prone to distortion under stress. When these loads are created by aerodynamic forces, which depend on the geometry of the structure and the orientation of the various structural components to the surrounding airflow, structural distortion results in changes in aerodynamic load, which leads to further distortion, and so on. Aeroelasticity refers to the interaction of aerodynamic and elastic forces.[1]

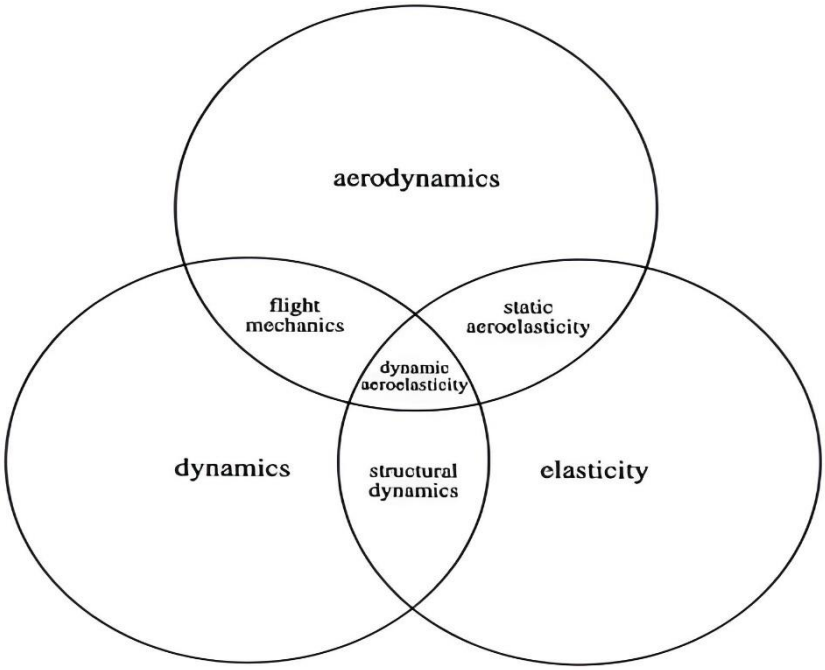

Figure I-1: Schematic of the field of aeroelasticity[1]

The stability (or rather instability) of a structure under wind is an intriguing subject in aeroelasticity. Because the aerodynamic force for a particular configuration of the elastic body increases fast with wind speed and the elastic stiffness is independent of wind speed, there may be a threshold wind speed at which the structure becomes unstable. Such instability may result in extreme deformations and the disintegration of the structure.

Flutter of structures such as airplanes or suspension bridges is a big issue when slight accidental disruptions cause more or less intense oscillations. It is defined as an issue of dynamic aeroelastic instability because of the interaction of aerodynamic, elastic, and inertia forces. The steady-state, or static, aeroelastic instability is a specific situation of an oscillation with zero frequency in which the inertia force can be ignored in general. [3]

Aeroservoelasticity (ASE) is an essential interdisciplinary topic in aerospace engineering that exists at the intersection of unsteady aerodynamics, structural dynamics, and

control systems. It is a study of the dynamic interactions between air loads, structural deformations, and autonomous flight control systems that modern aircraft encounter. With the introduction of flexible, lightweight structures, higher airspeeds, and closed-loop automatic flight control, the importance of ASE in current airplane design has grown significantly. Because aeroservoelasticity exists at the intersection of aerodynamics, structures, and control, its impact on aircraft design and operation necessitates a full grasp of these fundamental areas inasmuch as they contribute to the construction of an accurate mathematical model (Fig. I.2). While aeroelastic interactions have been researched for nearly a century, the effect of an active control system on dynamic aeroelasticity is a relatively new issue that has gained prominence with the introduction of current fly-by-wire designs. [4]

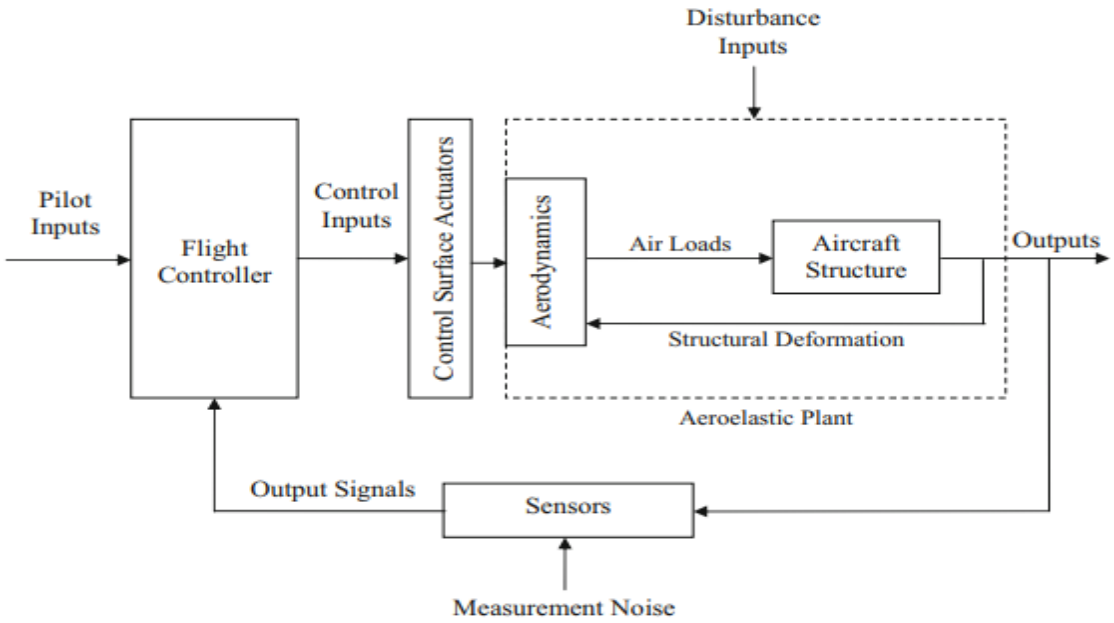

Figure I-2: Block diagram of a typical aeroservoelastic system[4]

An adaptive sliding mode controller is a control approach used in aeroservoelasticity that combines the principles of sliding mode control with adaptive control to address challenges associated with the interactions of aerodynamics, structures, and controls in aviation systems.

The adaptive sliding mode controller in aeroservoelasticity evaluates and adapts unknown factors such as aerodynamic coefficients, structural properties, and control gains. It continuously updates the control settings using measurement feedback and system identification techniques to maintain stable and accurate control performance. By combining sliding mode control with adaptive processes, the adaptive sliding mode controller seeks to

achieve robustness, stability, and accurate tracking of desired flying characteristics in the presence of uncertainties and dynamic interactions. [5]

I.1. Historical aspect of aeroelasticity:

Roxbee Cox and Pugsley invented the term 'aeroelasticity' in the early 1930s. It is a difficult phenomenon, especially when inertia forces are involved, and its solution necessitates the use of numerous mathematical and experimental techniques. [6]

The early years 1903-1919: this period was marked by:

- In their historic flight, Orville and Wilbur Wright used aeroelastic phenomena to control the roll of their biplane by using wing warping instead of ailerons. They were also aware of the undesirable aeroelastic effect of a propeller's loss of thrust owing to blade twisting from their research on the performance of thin propellers with broad blades. They discovered that when subjected to high thrust loads, the propeller tip twisted to partially empty itself. [6]
- On December 8, 1903, only nine days before the Wright brothers' takeoff at Kitty Hawk, Smithsonian Institute Professor Samuel P. Langley failed for the second time in an effort to launch his powered flying machine from a Potomac River houseboat. The failure was caused by insufficient wing-tip stiffness, which resulted in wing torsional divergence, a non-oscillatory aeroelastic instability that can be considered flutter at zero frequency. [6]
- During World War II, the great British engineer and scientist F. W. Lanchester made the first significant advances in flutter. Many deadly structural failures on two fighter types on the German side during World War I were ascribed to aeroelastic static divergence difficulties. [6]
- H. Reissner, author of numerous publications on aircraft structures, developed a detailed analysis of wing torsional divergence in 1926, demonstrating the importance of the relative locations of the aerodynamic center of pressure and the elastic axis. [6]

Post WWI to about 1930: this period was marked by:

- In 1918, W. Ackerman was assigned a thesis problem on airfoil theory by Professor Prandtl in Göttingen, and the unfinished work was eventually reassigned to W. Birnbaum concerning the classical vortex theory of the two-dimensional steady flow of thin airfoil. [6]

- In 1923, A. G. von Baumhauer and C. Koning12 reported the results of their practical and theoretical work. They were primarily concerned with the binary flutter of the wing in vertical bending paired with ailerons motion. [6]
- In 1929, Frazer and Duncan of the NPL released a very thorough monograph that British workers dubbed "The Flutter Bible." It identified and studied phenomena using reduced wind tunnel models. [6]
- From 1925 through 1929, numerous other notable German researchers investigated flutter using beam-rod principles, but with quasi-steady aerodynamics that ignored the trailing wake effects. [6]

1930 to WW II: this period was marked by:

- In 1932, the de Havilland Puss Moth airplane, a general purpose single engined monoplane with folding V-strut wings, was involved in a series of catastrophic accidents. The ARC Accidents Investigation Subcommittee's thorough report in 1936 compiled more than 50 distinct detailed investigations. The conclusion that not just wing flutter but also rudder and elevator flutter may have been involved, that the V-struts were a factor in the wing flutter, and that the rudder flutter appeared to require a starting impulse such as stormy or turbulent weather, was particularly interesting. [6]
- In 1935, Küssner developed an empirical calculation based on the reduced torsional frequency (wb/V) after correlating several flutter events and accidents. [6]
- In 1936, experimental study at low speeds on the effects of angle of attack on aeroelastic phenomena began. J. Studer began investigating the influence of high angles on flutter speed, an intrinsically nonlinear problem, under the supervision of Professor Ackeret. [6]
- Taylor and Browne investigated the possibility of a new type of instability known as propeller whirl flutter in a study of vibration isolation of aviation engines in 1938. [6]

WW II to the mid-1950s: Rapid changes occurred in aviation development during World War II. The trend toward faster speeds and all-metal aircraft continued. External weaponry, tip tanks, and other appendages were carried by fighter aircraft and long-range bombers of various shapes, with low and high aspect ratios. On the P-80, for example, there was a problem with tip-tank flutter. Flutter issues arose on the battlefield as a result of appendages or battle damage, which could result in a loss of balance weights or reduced stiffness. [6]

Impact of new technologies on aeroelasticity: Composites were discovered four decades ago as a breakthrough technology that will change aviation engineering. Since then, various studies on the use of composites in wing design have been done, leading to the development of aeroelastic tailoring based on a combination of ply layup and fiber orientation. Aeroelastic tailoring has been used on a few wings. On the other hand, with a few exceptions, all current rotor blades are made of composites. Aeroelastic tailoring for composite blades has been extensively proven. However, aeroelastic tailoring is not used in composite rotor blades, and composite blades are widely used due to their good fatigue properties. [7]

Around 15 years ago, active materials were found as potentially beneficial for a number of aerospace applications as both sensors and actuators, and the field of'smart structures' or 'adaptive structures'' has grown since then. Combining active materials, controllers, and microprocessors is becoming increasingly popular. Many major applications, both fixed-wing and rotary-wing, are related to aeroelasticity, and a number of survey articles on the subject have been written. It's worth wondering if the future of adaptive materials-based actuation for aeroelastic applications will be similar to the future of composite materials and their use to aeroelastic tailoring. [7]

I.2. Concept of aeroelasticity:

The classical theory of elasticity is concerned with the stress and deformation of an elastic body in the presence of prescribed external forces or displacements. The external loading operating on the body is, in general, independent of the body's deformation. It is commonly considered that the deformation is minor and has no significant impact on the action of external forces. In such a circumstance, we frequently ignore changes in body size and base our estimates on the initial shape. Even in issues of bending and buckling of columns, plates, or shells, either the external loading or the boundary limitations are taken into account as mandated. However, the situation is different in most significant aeroelasticity situations. The aerodynamic forces are highly dependent on the body's orientation in relation to the flow. The elastic deformation is significant in determining the external loading. Until the elastic deformation is determined, the magnitude of the aerodynamic force is unknown. In general, the external load is unknown until the situation is resolved. [3]

Aeroelasticity, according to Collar, is the study of the mutual interaction of aerodynamic forces applied to structures in the face of air flows, as well as the impact of this study on aircraft design [3]. Collar proposed several years ago that aeroelasticity may be

advantageously represented as a triangle of disciplines, dynamics, solid mechanics, and (unsteady) aerodynamics. figure (I.3) [2].

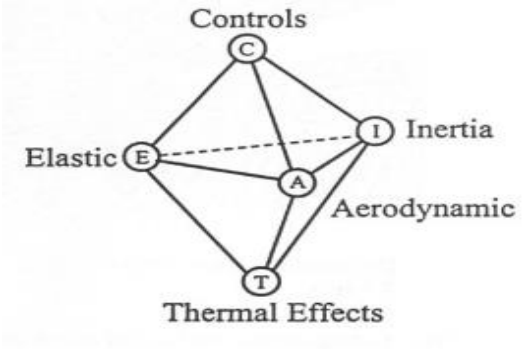

Figure I-3: Expanded Collar triangle[2].

Aeroelasticity is a flow-structure interaction phenomena that occurs when the fluid is significantly "lighter" than the structure. The term "aeroelasticity" often refers to the study of structure vibrations induced by flow, as opposed to static deformations caused by average aerodynamic forces [9]. Thus, flow-structure interactions that result in vibrations have three causes:

- **Turbulence in the upstream flow**, this creates an unstable forcing and causes the structure to shake. This vibratory response, which is normally handled with without addressing structural flow coupling, cannot be avoided immediately and must be estimated. [10]
- **The separation of alternating vortices**, as a result, the aerodynamic force is virtually periodic. When the frequency of this force surpasses the frequency of an eigenmode of a structure, the collision phenomenon occurs, and the flow-structure coupling becomes severe. [9]
- **A high average wind speed**, Because of its flexibility, the structure creates forces linked with its movement. These moving forces can make the system unstable above a certain speed due to a variety of coupling phenomena known as flutter. They virtually invariably destroy structures and should be avoided at all costs. Thus, the fundamental purpose of flutter research is to find the critical speed. [9].

I.3. Classification of aeroelasticity:

In this part we will provide an overview of the major aeroelasticity classes.

I.3.1. Based on the mathematical models used to describe the aeroelastic behavior of structures:

In the first classification based on the mathematical model used to describe the aeroelasticity of the structure, we distinguish two major types:

I.3.1.1. Linear aeroelasticity:

Linear aeroelasticity is a branch of aeroelasticity that studies the complicated connections between aerodynamics, structural dynamics, and control systems where the system's behavior can be accurately represented by linear relationships. It examines aircraft and aerospace structures that have been subjected to minor deformations and flow field perturbations. We want to know how the shape of the objects and how they are manipulated effect how they move. [10]

Flutter is a common occurrence in linear aeroelasticity; self-excited and potentially dangerous oscillations may arise when the structure interacts with the aerodynamic forces; persistent oscillations result when aerodynamic forces interact with structural dynamics; calculating flutter speed and anticipating flutter mode forms are two aspects of flutter analysis. [10]

I.3.1.2. Non-linear aeroelasticity:

Nonlinear aeroelasticity is the study of the interactions between inertial, elastic, and aerodynamic forces on engineered structures exposed to airflow and exhibiting nonlinearity. Over the last 30 years, nonlinear aeroelasticity has grown in popularity as a research topic. Faster computers, increasingly flexible structures, automatic control systems for aircraft and other engineering goods, new materials, optimisation-based design methodologies, and other factors have all contributed to this advancement. [11]

The study of nonlinear ordinary and partial differential equations is known as nonlinear dynamics. Nonlinear equations, unlike linear differential equations, have no universal analytical solutions and, in some circumstances, many solutions may coexist under the same operating conditions. Furthermore, nonlinear systems can have a far wider range of solutions than linear systems. The free stream airspeed and air density (or flight altitude) are the primary working conditions of an aeroelastic system, but the Reynolds number, Mach number, and mean angle of attack can also be significant. The number and type of nonlinear equations of motion solutions might change dramatically when these system characteristics change. Bifurcation analysis is the study of the changing nature of solutions as system parameters are changed. [11]

Historically, much effort has been expended on designing and constructing engineering structures that are as straight as feasible. Despite this, nonlinearity, whether weak or strong, has always existed in engineering systems [11]. The three main sources of nonlinear functions seen in these systems are:

- **The structure:** The interesting structural nonlinearities occur during the normal operation of the underlying engineering system. Geometric nonlinearity (caused by large deformations), clearance (i.e. free play, contact, and other non-smooth phenomena), dissipative (i.e. friction or other nonlinear damping forces), and inertial (of particular interest in rotors and turbomachinery) are the most common types of nonlinearity appearing in structures. [11]
- **The aerodynamics:** Aerodynamic nonlinearities are caused by unstable separated flow, oscillating shock waves, or a combination of the two. It should be highlighted that nonlinearity in aerodynamics is inertial, dissipative, and elastic. [11]
- **The control system:** Passive and/or active control systems are increasingly being incorporated into engineering constructions. These systems can be designed to either stabilize the structure (by suppressing or mitigating undesired vibrations) or to control it (as in aircraft autonomous flight control systems). Passive systems can be viewed as structural elements and hence fall under the structural nonlinearity category (if nonlinear). However, active systems can have a number of planned and incidental nonlinearities that can be disabled by running the structure in open loop mode. [11]
- External storage on airplanes that carry them (mostly military aircraft) can also be a cause of nonlinearity. External fuel tanks, explosives, and missiles can all create store-induced oscillations, especially in transonic flight situations. [11]

Nonlinear aeroelastic flutter refers to self-excited oscillations caused by the system's interaction with structural dynamics and nonlinear aerodynamic forces. Nonlinear flutter has more complex behavior than linear flutter because of nonlinear interactions between structural deflections and aerodynamic loads, which are characterized by the stability limit and flutter speed. Large deformations, geometric nonlinearities, unstable aerodynamics, and control surface nonlinearities can all induce nonlinear effects. Nonlinear flutter is studied using a variety of mathematical and numerical techniques. Although perturbation approaches such as the method of different scales can yield approximate answers, numerical methods such as time-domain simulations and bifurcation analysis provide a more accurate

representation of the nonlinear dynamics involved. Limit cycles, amplitude modulation, bifurcations, and other nonlinear flutter events can be studied using these methods. [11]

I.3.2. Based on the structural behavior:

In this time, the classification is based on the structural behavior, where we have two different types, static and dynamic aeroelasticity while each one contains different natures of structural response. This classification is showed in the figure I.4:

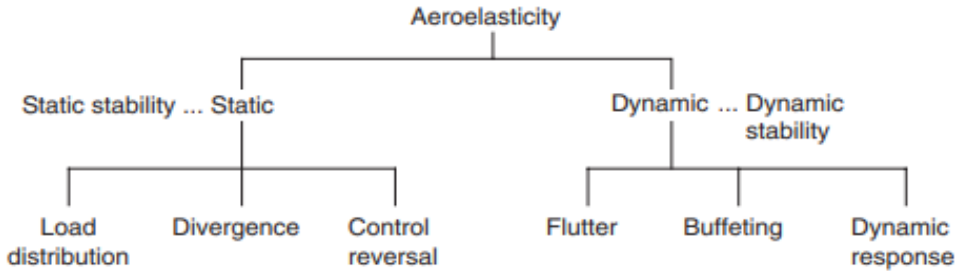

Figure I-4: Different natures of structural response (The various aeroelasticity problems)[11]

I.3.2.1. Static aeroelasticity:

The study of flight-vehicle phenomena connected with the interaction of aerodynamic loading caused by steady flow and the subsequent elastic deformation of the lifting-surface structure is known as static aeroelasticity. These phenomena are characterized as being indifferent to structural deflection rates and accelerations. There are two types of design problems encountered in this field. The first, and most universal to all flight vehicles, are the effects of elastic deformation on air loads, as well as the impacts of air loads on elastic deformation, under normal operating conditions. These factors can have a significant impact on performance, handling characteristics, flight stability, structural-load distribution, and control efficacy. The static instability of the lifting-surface structure has the potential to cause catastrophic failure in the second category of challenges. This instability is commonly referred to as "divergence," and it has the potential to limit the flight envelope. Simply put, divergence happens when a lifting surface deforms under aerodynamic loads in such a way that the applied load increases, and the increasing load deflects the structure farther until it fails. Such a failure is caused by more than just a load that is too large for the structure as designed; rather, the aerodynamic forces interact with the structure to cause a loss of effective stiffness. [12]

Static aeroelasticity is a subject of aeronautical engineering concerned with the complicated interaction between aerodynamics and structural mechanics. By analyzing and

comprehending the equilibrium and deformation properties of aerospace structures, static aeroelasticity contributes in the design of safe, efficient, and high-performance flight vehicles. Continuous research and development in this field is essential for the advancement and refinement of aerospace technologies.

I.3.2.2. Dynamic aeroelasticity:

The dynamic interactions between aerodynamics and structural mechanics in airplane wings are studied in dynamic aeroelasticity, an important area within aeronautical engineering. It considers the unstable flow circumstances encountered during flight, including as gusts, maneuvers, and control inputs, which all produce varying aerodynamic forces on the structure. These forces can result in undesirable dynamic reactions as flutter, divergence, limit cycle oscillations, and aeroelastic instabilities. [13]

When analyzing an airplane's dynamic reaction in flight, it is common to assume that the structure is fully rigid. Externally applied forces, such as air or ground loads, are put into equilibrium with aerodynamic and inertial forces that arise as a result of the airplane's translational and rotational motion as a rigid body, and these forces are considered to comprise the total force system acting on the airplane. In two ways, this approach introduces mistake. Deformation of the structure may result in additional aerodynamic forces that impact the airplane's overall response. Furthermore, if the dynamic reaction is the direct result of rapidly applied external forces, the airplane not only moves and rotates, but structural vibrations are also created. The latter have a large impact on the internal stress distribution in the structure but a slightly lesser impact on the overall reaction of the airplane. [13]

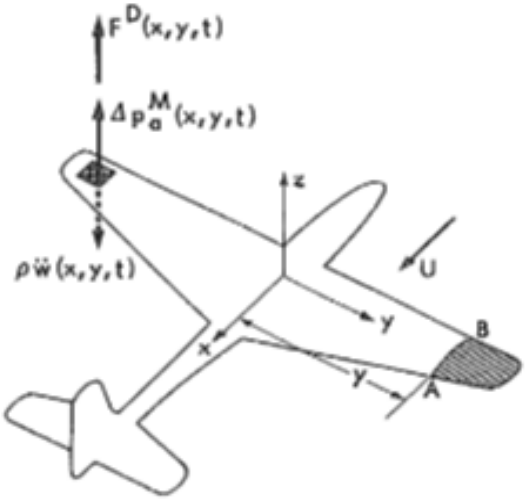

Figure I-6: Elastic airplane subjected to disturbing force. [13]

External forces that are applied quickly might come from a variety of sources during the life of a contemporary airplane. Wind gusts are one of the most powerful disruptive forces. In civil aircraft, gusts are caused by atmospheric turbulence, however in military aircraft, gusts can be caused by both atmospheric turbulence and severe blast-induced shock waves. Impact forces during landing are a source of large, rapidly applied external forces, which are of particular importance in relation to the internal stress components of dynamic response problems. Furthermore, rapidly variable external forces may be linked with catapulting, bomb dropping, rapid maneuvering caused by sudden deflection of control surfaces, and turbulent wake behind wings, nacelles, or other airplane components. [13]

I.4. Flutter phenomenon:

Flutter is defined before as the dynamic instability of an elastic body in an airstream. It is most commonly encountered in aircraft structures that are subjected to high aerodynamic loads, such as wings, tail units, and control surfaces. Flutter happens at a critical or flutter speed V_f, which is the lowest airspeed at which a given structure will oscillate with sustained simple harmonic motion.

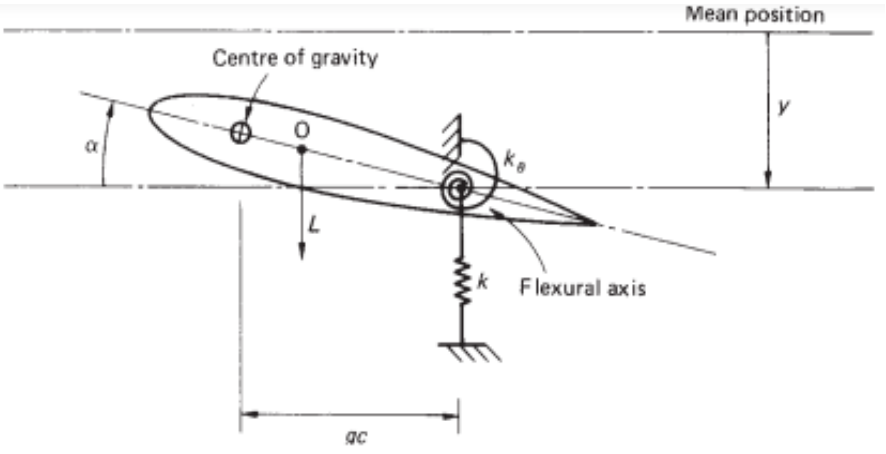

Figure I-7: Flutter of a wing section[11]

Flight at speeds less than or greater than the flutter speed represents stable and unstable (divergent) structural oscillation, respectively. In general, an elastic system with only one degree of freedom cannot be unstable unless it has some unusual mechanical property, such as a negative spring force or a negative damping force. However, systems with two or more degrees of freedom can be unstable without demonstrating unusual behaviors. For certain phase differences, the forces associated with each individual degree of freedom might interact, resulting in divergent oscillations. The flutter of a wing with coupled flexural and

diving modes is an important example of this sort of instability. Classical flutter occurs when two distinctly different types of oscillating motion interact, causing the resultant motion to be divergent.[1]

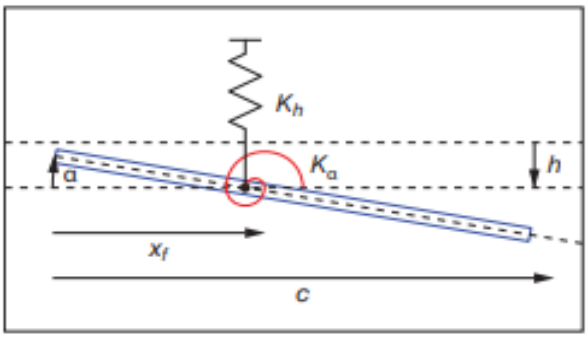

Figure I.8: Pitch-plunge wing section[11]

Non-classical flutter may involve only one form of motion; for example, "aileron buzz" occurs at high subsonic speeds and is connected with the shock wave on the wing ahead of the aileron. When the aileron oscillates downward, the flow over the upper surface of the wing accelerates, magnifying the shock and resulting in a decrease in pressure in the boundary layer behind the shock. As a result, the aileron tends to be sucked back to its neutral position. When the aileron is raised, the shock intensity decreases and the pressure in the boundary layer rises, tending to drive the aileron back to its neutral position. At low frequencies, these pressure changes are around 180 degrees out of phase with aileron deflection and hence act as aerodynamic damping forces. At higher frequencies, a pressure component arises in phase with the aileron velocity, exciting the oscillation. If this exceeds all other dampening acts on the aileron, a high frequency oscillation occurs with only one type of motion present, rotation of the aileron about its hinge.[1]

Experiments on wing flutter demonstrate that the oscillation is self-sustaining, requiring no external oscillator or driving agency. The motion can persist or expand throughout a range of wind speeds that varies depending on the design of the wing and the test conditions. Flutter happens at any wind speed above the critical for a basic cantilever wing. In other cases, such as aileron flutter, there may be one or more speed ranges for which flutter occurs, and these are confined at both ends by critical speeds at which an oscillation of constant amplitude may just maintain itself. Because flutter is an oscillation caused by aerodynamic forces and requires no external source of energy other than the airstream, it is only feasible if the oscillating body, whose mean position is considered to be stationary, can extract energy from

the airstream. As a result, the potential of flutter can be considered in terms of the energy relationship.[3]

By investigating the stability of tiny motions, we can obtain an acceptable specification of a system's flutter qualities in a substantial fraction of all cases. It is then sufficient to study a vibration with exponential time dependence e^{pt} (p complex), because superposition allows all other little motions to be built up from it. If minor deformations are dynamically unstable, the scenario is undesirable on any manned or automatically controlled aircraft, regardless of the stability of larger ones. In practice, if the smaller displacements are stable, the bigger displacements are usually stable as well. In fact, they may have substantially better stability, as in the case of amplitude limiting owing to stalling on flutter oscillations, which is frequently observed in the wind tunnel on very flexible aeroelastic models.[13]

Flutter has been predicted using purely theoretical methods, analog computing, wind tunnel or rocket experiments on scaled dynamic models, and full-scale aircraft flight testing. All freshly designed aircraft are subjected to a ground resonance test early in the prototype's life to identify true normal modes and frequencies. The principal goals of such testing are to verify the accuracy of the computed normal modes on which flutter predictions are based and to reveal any unexpected anomalies in the aircraft's vibrational behavior. Typically, the aircraft is supported by a low frequency support system or even by its deflated tyres. On the wings and tail, electrodynamic exciters are attached in pairs, with accelerometers serving as measuring devices. The resonance frequencies are often discovered first by recording the amplitude and phase of a specified number of accelerometers throughout a given frequency range. After determining the resonant frequencies, the aircraft is excited at each of these frequencies in turn, and all accelerometer data are acquired at the same time. The choice of which of these is most cost-effective in a given situation is determined by a variety of parameters, including the expected margin of safety from flutter, the Mach number range, and the number of alternative mass and structural configurations to be studied. A quick computation based on severe simplifying assumptions can sometimes be enough to ensure that the designer has nothing to fear from flutter. This is defined as a passive solution to increase flutter performance, which is not ideal for our situation. [13]

On the other hand, there is an excellent prospect of increasing flutter performance by installing in the structure a correctly designed, swiftly responding automatic control system, triggered in closed-loop fashion by the motion to be stabilized; this is referred to as an active solution. However, we cannot rely on the human pilot to adjust for flutter, as he does for

certain instabilities of the rigid airplane, because the frequencies are too high for his effective response, leading to what is known as "Aeroservoelasticity".[1]

I.5. Concept of aeroservoelasticity:

Aeroservoelasticity (ASE) is a tough problem to deal with since it exists at the intersection of aerodynamics, control, and structural dynamics. However, it is a significant subject, critical to the design of modern aircraft, and can only be disregarded at the designer's peril. [14]

It's beneficial to go over several designs that have been tested in flight or wind tunnels with active flutter suppression and load alleviation equipment. A B-52E airplane, the most comprehensive control configured vehicle (CCV), was tested in a program co-led by the Air Force Dynamics Lab, Boeing, and NASA Langley. This plane possessed flutter mode control (FMC), maneuver load control (MLC), and ride control (RCS), as well as some extra features like gust load alleviation (GLA). On August 2, 1973, a flight test was conducted to show the flutter suppression system, and a flight speed of 10 knots over critical open-loop speed was achieved. Flutter was created by ballasting the additional tip tanks with 2000 pound lead weights, and control was provided by modified outboard ailerons and flaperons that regulated a 2.4 Hz wing bending mode. NASA's DAST (drones for aeroservoelastic testing) program conducted flight testing of flutter suppression control laws and gust load reduction approaches in the late 1970s and early 1980s. The goal of this program was to test clean wings from transport category planes with supercritical air foils in the transonic range. Aeroservoelastic encounters have also occurred on fighter type aircraft, such as the YF-16, which experienced control system interactions with the flexible wing mode; the YF-17, which experienced control system interactions with both the flexible wing and the rigid body mode; the F/A-18, which experienced aeroelastic oscillations induced by the control system; and the X-29A, which experienced body degree of freedom interactions with the forward swept wing. A major percentage of flutter suppression experimental research has been undertaken on aeroelastically scaled wind tunnel models, with a focus on wing/store flutter suppression).[15]

Because of the problem's intricacy, a considerable number of investigations have been theoretical in character, with the goal of comprehending the fundamental issues or supporting the experimental effort.

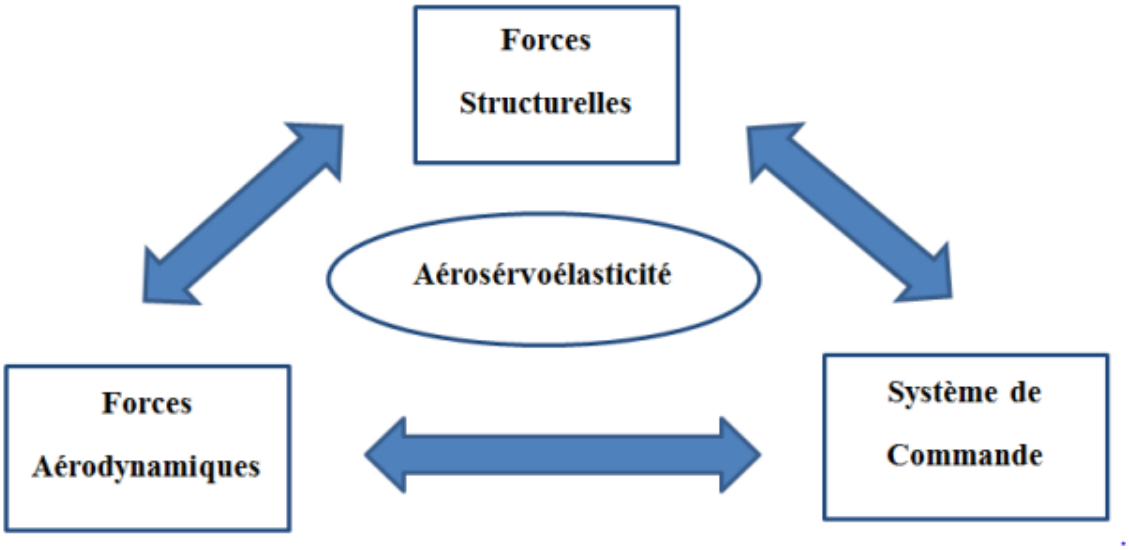

Figure I-9: Discipline interaction triangle of aeroservoelasticity

I.5.1. Control in ASE systems:

The aeroservoelastician has a dual control dilemma. The plant has practical constraints that preclude a continuous and smooth change of the dynamic variables at all space points. This is the natural uncontrollability (or unreachability) of an infinite-dimensional system that is attempted to be governed by a finite number of improperly modulated control inputs. On the other hand, even if the designer had a big army of control input variables at his disposal, developing a valid principle (control law) controlling each of them would be challenging. This is the other intrinsic restriction that derives from an incomplete understanding of plant dynamics and results in a deficient mathematical model of the plant. Thus, the ASE control design process is an endless struggle with the problem of underactuated and unpredictable plant dynamics [14]. Attempts to regulate an unpredictable ASE system have two components:

- Creating an accurate mathematical model of the plant by correctly capturing every significant physical process, followed by the construction of a controller based on the plant model. [14]
- Using an online identification of the real plant based on its measured input-output record to adjust controller parameters in response to changing plant behavior. [14]

The fundamental difficulty in ASE mathematical analysis and design is developing an appropriate unstable aerodynamic model of aircraft wings and tails (or canards). At subsonic and supersonic speeds, the aeroelastic plant for flutter suppression of a thin wing-like surface

is determined utilizing small-disturbance, prospective aerodynamic models with a harmonic (frequency response) theory. This type of model is linear and may be used to directly construct an aerodynamic transfer matrix, and then a linear, time-invariant state-space model via analytic continuation in the Laplace domain. However, there are several critical flow regimes where a linearized model is ineffective. ASE applications involving unstable separated flows and transonic shock-induced flows are inherently nonlinear and necessitate advanced computational fluid dynamics (CFD) modeling techniques. The ASE plant for such a case is further complicated by the wing's separated wake and/or leading-edge vortex interacting with the tail, resulting in an irregular and often disastrous deformation of the tail either on its own or as a result of fast and massive elevator deflections. A wing-tail-elevator coupling of a shock-vortex interaction necessitates fully viscous flow modeling, which can only be accomplished using the Navier-Stokes method. Another example of nonlinear ASE is the regulation of unstable control surface buzz and shock-induced flow separation experienced by a transonic aircraft, which results in nonlinear flutter or limit-cycle oscillation (LCO). Depending on the geometry, structural stiffness, Mach number, and Reynolds number, an acceptable CFD model would require a full-potential code, coupled inviscid/boundary-layer approach, or a Navier-Stokes method. Semiempirical models for separated and shock-induced flows are sometimes developed using wind-tunnel test results since they do not require unstable CFD computations to be done in loop with structural dynamic and control-law calculations. However, the reliability of such a correlation must be carefully examined before it is used in ASE design and analysis. In the literature, the use of CFD models (Euler/Navier-Stokes) to construct linearized aerodynamic transfer functions/matrices has also been recommended. This clearly aims to use the same linearized model for a variety of flight conditions (Mach number, angle of attack, and so on) rather than having to repeat a CFD computation for each such condition. An auto-regressive moving average (ARMA) model can be used to update the approximation coefficients, bringing such a method very close to an adaptive control application (although in the open loop). However, even in a restricted range of operating circumstances, approximating a nonlinear system with a linear transfer function can be erroneous. An alternate way for developing an ASE model is to use flight-test data, such as the neural-network identification method proposed by Boely and Botez. [16]

Despite early breakthroughs in proving active flutter suppression/load relief, ASE has mostly remained an experimental area and has yet to be deployed on any aircraft. The challenge in constructing a multivariable control system that is sufficiently resilient to the

parametric uncertainties in the underlying unstable aerodynamic model is largely responsible for this astounding failure. Clearly, aircraft designers and operators are hesitant to take risks until (what may be regarded) sufficiently trustworthy ASE modeling and analytical procedures become common. [17]

A closed-loop controller for an ASE application must be extremely resilient to modeling errors due to the inherent uncertainty of an unstable aerodynamic model. In addition, such a controller must be able to react to changing flying conditions. As a result, an ASE control rule must not only be resilient, but also self-adaptive, making it mathematically nonlinear even for a linear aeroelastic plant operating in the subsonic and supersonic domains. Additionally, developing a control law based on nonlinear aeroelastic iterative models can be a time-consuming and computationally costly procedure. Adaptive control has been a focus of active research for several decades, and several helpful design concepts that can be applied to ASE have arisen. However, in many circumstances, these are "application specific" (rather than broad), if not wholly ad hoc. Thus, deriving an ASE control-law for a specific situation is as difficult as the challenge of aeroelastic modeling. As a result, ASE has remained a daunting technological challenge. [18]

Separated and shock-dominated flows interact with the aircraft structure in a nonlinear manner, resulting in unstable oscillations, hysteresis, or limit cycles. Nonlinear aeroelastic plant control necessitates either descriptive function approximations, Lyapunov-based controllers, feedback linearization, or sliding-mode (variable structure) control. Furthermore, resilience is a significant consideration for nonlinear plants. Adaptation of controller parameters for both linear and nonlinear plants adds another layer of complexity to the ASE design and analysis. Depending on the application, this can be addressed by gain scheduling, self-tuning regulation, model reference adaptive laws, or recursive backstepping. [19]

I.6.Concept of sliding mode controller:

One of the fundamental subjects of modern control theory is control in the presence of uncertainty. There is always a mismatch between the actual plant dynamics and the mathematical model used to construct the controller in the formulation of any control problem. External disruptions, unknown plant characteristics, and parasitic dynamics are the most common causes of these differences (or mismatches). Control engineers face a difficult problem in designing control laws that give the necessary closed-loop system performance in the presence of these disturbances/uncertainties. This caused great interest in the development of so-called robust control approaches, which are meant to overcome the problem. Despite

considerable and effective development of robust adaptive control, sliding mode control (SMC) is still the most successful strategy in dealing with constrained uncertainties/disturbances and parasitic dynamics. [20]

Sliding modes were originally recognized as a particular mode in variable structure systems (VSS). These systems are made up of a range of structures, with rules for switching between them in real time to ensure appropriate system performance, whereas utilizing a single fixed structure could be unstable. As a result, VSS can be thought of as a collection of subsystems, each of which has a fixed control structure and is valid for specific regions of system behavior. The closed-loop system appears to be designed to have new qualities not found in any of the constituent substructures alone. Furthermore, these qualities include insensitivity to certain (so-called matched) external shocks and model errors, as well as resistance to parasitic dynamics, in a special mode known as a sliding mode. A very important practical aspect of sliding modes is the ability to achieve reduced-order dynamics of the compensated system (termed partial dynamical collapse). One of the earliest books on the subject to be published in English is. In the late 1950s, the Soviet Union began to develop these revolutionary ideas. [21]

SMC's concept is based on the introduction of a "custom-designed" function known as the sliding variable. The sliding manifold (or sliding surface) is defined when the properly configured sliding variable equals zero. While the system trajectories belong to the sliding manifold, the right design of the sliding variable offers suitable closed-loop system performance. The idea behind SMC is to steer the system's trajectory to the properly chosen sliding manifold and then maintain motion on the manifold using control, thereby exploiting the sliding mode's main features: its insensitivity to external and internal disturbances matched by the control, ultimate accuracy, and finite-time convergence of the sliding variables to zero. [22]

Itkis wrote the first well-cited English paper on SMC in 1976. By 1980, the major contributions to SMC theory had been accomplished and presented in Utkin's 1981 monograph (in Russian) and its English translation. DeCarlo et al. conducted a full assessment in [23]. The two-step approach for SMC design was explicitly outlined in these articles.

The initial objective is to create a switching function that ensures the system motion on the sliding manifold (referred to as the sliding motion) meets the design parameters. The second stage is to choose a control law that will make the sliding manifold appealing to the system state in the presence of both external and internal disturbances/uncertainties. It should

be noted that this control law is not always discontinuous. Observers based on SMCs can estimate systemstates in the presence of unknown external disturbances, which can also be explicitly reconstructed online by an observer. [23]

Control chattering was still a barrier to SMC deployment. The fundamental purpose for the emerging so-called second-order sliding was to address control chattering. Thus, the already established conventional SMC theory gained a substantial boost in the mid-1980s, when new "second-order" notions surfaced, and then again in the early 2000s, when "higher-order" concepts were proposed. The following factors compelled the introduction of these new paradigms:

- The traditional sliding mode design approach requires the system relative degree to be one in relation to the sliding variable. This can severely limit the sliding variable's selection. [24]
- Additionally, a sliding mode controller frequently produces high-frequency switching control action, which causes the so-called chattering effect, which is difficult to avoid or mitigate. [24]

Higher-order sliding mode (HOSM) controllers, which can drive to zero not just the sliding variable but also its k-1 consecutive derivatives (k^{th}-order sliding mode), alleviate these inherent issues of traditional SMC. Because the high-frequency control switching is "hidden" in the upper derivative of the sliding variable, the unique strategy is applicable for any relative degrees and considerably reduces the well-known chattering effect. [24]

HOSM gives sliding accuracy proportionate to the k^{th} power of the sampling period when implemented in discrete time, making it an enhanced accuracy robust control mechanism. Because only the k^{th} derivative of the sliding manifold is proportional to the high-frequency switching control signal, the switching amplitude is well dampened at the sliding manifold level, reducing chattering greatly. [24]

The approach's unique power is demonstrated by the invention of practical arbitrary-order real-time resilient exact differentiators, the performance of which is demonstrated to be asymptotically optimal in the presence of Lebesgue-measurable input disturbances. In advanced HOSM-based observers, the HOSM differentiators are utilized to estimate the system state in the presence of unknown external disturbances, which are also recreated online by the observers. Furthermore, HOSM-based parameter observers have been constructed. [24]

A HOSM controller combined with the previously stated HOSM-based differentiator results in a robust and precise output-feedback controller. There is no need for precise mathematical models of the plant. SMC of arbitrary smoothness can be created by artificially raising the system's relative degree, which considerably reduces chattering. For example, if virtual control in terms of the control derivative is designed in terms of SMC, the continuous control function can be produced. The control function will be continuous in this situation because it equals the integral of the high-frequency switching function. When dealing with parasitic/unmodeled dynamics, the SMC function will switch at a reduced frequency (control chattering). Chattering attenuation is obtained by designing the SMC in terms of the control function's derivative. [24]

A wide range of applications, including DC/DC and AC/DC power converters, control of AC and DC motors and generators, aircraft and missile guidance and control, and robot control, demonstrate the usefulness of standard SMC and HOSM control and observation approaches. [24]

Conclusion:

This chapter present the theorical part, where we discussed the different concepts used in this research, starting with an historical aspect of the aeroelasticity. After, that we explained the concept of aeroelasticity as a phenomenon presented by the interaction of aerodynamic and elastic forces, and we presented different types of aeroelasticity with some details. One of these types is the dynamic aeroelasticity which is so important in our research because it contains the most important issue in this field and also it presents the problematic of our research which is the flutter phenomenon.

Then we discussed the flutter phenomenon, its concept and historical aspect, also we defined the different types of flutter and some solutions presented by engineers to eliminate or reduce this problem. One of the most useful technics against this problem is the aeroservoelasticity. After that, we discussed in this chapter the concept of aeroservoelasticity starting with a brief historical aspect, to an analytical explanation of the different technics of the aeroservoelasticity to control the flutter phenomenon, especially the adaptive sliding mode controller that we will use in this research.

Finally, we finished this chapter with discussing the concept of the sliding mode controller which defined as the most useful and efficient methods that used to reduce the flutter phenomenon with a brief historical aspect, before we talk about it with more details in the third chapter in this research.

CHAPITRE II: Modulization of an aeroelastic system (TMU II wing)

Introduction:

The first step in studying system control is to model it. Modeling a system may entail developing a mathematical entity capable of explaining and forecasting the dynamic behavior of the system when subjected to external pressures. [31]

This mathematical model is generated using formalisms such as Newton's formalism and Lagrange's formalism, with the latter being a particularly effective and powerful tool for putting the most complicated systems under equations. It offers a way for determining the equations of motion by computing simply the kinetic and potential energy of the system. [31]

In this chapter, we will use the Lagrange formalism on a two-dimensional wing section to derive the equations of motion that govern the behavior of the aeroelastic system.

Once the mathematical models are generated, we will develop a numerical application for the TAMU WING II wing model.

II.1. Model hypotheses:

We make assumptions to eliminate the physical effects of low importance in order to facilitate the study of our system. The assumptions considered in our case are:

- The effect of gravity is ignored,
- The pitch angle is assumed to be small,
- The lift force L and the moment M are calculated in a quasi-steady state.

II.2. Structural modulization:

The equations of motion that control the motion of a two-dimensional wing can be obtained from the Lagrange equations.

In the wing, there are two degrees of freedom: pitch movement (torsion) and vertical movement (flexion). The wing is capable of pitching about an elastic axis, which is perpendicular to the shear axis. The structural stiffness for both motions is represented by two springs with stiffness K and K_h that are both constant.

As shown in Figure (II.1), notable points on the wing include center of gravity "c_g", elastic center "c_e", and aerodynamic center "c_a".

The parameters a and e such that $-1 < a < 1$ and $-1 < e < 1$ determine the position of the center of gravity "c_g" and the elastic axis.

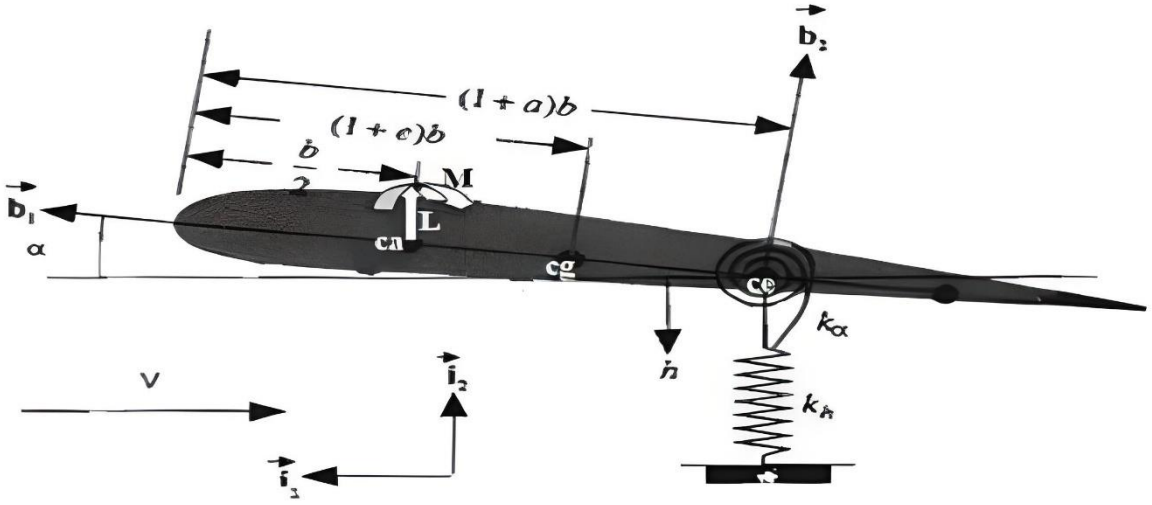

Figure II.1: The geometry of the two-dimensional wing section

The $\vec{\imath}$ frame represents the inertial frame, while the $\vec{b}$ frame is fixed to the wing, its origin is fixed at the elastic center "c_e" and oriented as shown below with the $\overrightarrow{b_1}$ axis is directed towards the leading edge. Lift L and moment M are measured at the aerodynamic center.

The vertical motion and pitch equations can be obtained using Lagrange's equations by calculating the potential and kinetic energy of the system.

II.2.1. Potential energy:

The potential energy is entirely made up of elastic energy; it represents simply the elastic energy stored on the two springs, with gravity being ignored [30]. This equation gives the entire potential energy:

$$U = \frac{1}{2}k_h h^2 + \frac{1}{2}k_\alpha \alpha^2 \qquad \text{(II.1)}$$

Where: K_α and K_h are the stiffness coefficients of vertical motion and pitch motion, respectively.

II.2.2. The kinetic energy:

The total kinetic energy of the system is given by [30]:

$$T = \frac{1}{2}mI_{cg}.\vec{v}_{cg} + \frac{1}{2}I_{cg}\dot{\alpha}^2 \qquad \text{(II.2)}$$

With:

I_{cg}: Is the moment of inertia around the center of gravity.

m: represents the mass of the wing.

V_{cg}: This is the velocity of the center of gravity, which is calculated through the following formula [1]:

$$\overrightarrow{v_{cg}} = \vec{v}_{ce} + \dot{\alpha}\vec{b}_3 \times b[(1+\alpha) - (1+e)]\vec{b}_1 \quad \text{(II.3)}$$

With:

$\vec{v}_{ce}$: The elastic center velocity.

α: The angle of incidence.

And we have:

$$\overrightarrow{v_{e_a}} = -\dot{h}\vec{\imath}_2 \quad \text{(II.4)}$$

Where: h is the vertical displacement.

As well as:

$$\overrightarrow{b_2} = \vec{b}_3 \times \vec{b}_1 \quad \text{(II.5)}$$

So, the center of gravity velocity becomes:

$$\overrightarrow{v_{c_g}} = -\dot{h}\overrightarrow{\imath_2} + \dot{\alpha}b(a-e)\vec{b}_2 \quad \text{(II.6)}$$

The relation between the reference $\vec{b}$ and the reference $\vec{\imath}$ is given by the following matrix relation [30]:

$$\begin{bmatrix}\overrightarrow{b_1}\\ \overrightarrow{b_2}\end{bmatrix} = \begin{bmatrix} cos(\alpha) & sin(\alpha) \\ -sin(\alpha) & cos(\alpha)\end{bmatrix}\begin{bmatrix}\overrightarrow{l_1}\\ \overrightarrow{l_2}\end{bmatrix} \quad \text{(II.7)}$$

We use the relation (II.7) and we assume that the pitch angle is low ie. that cos(α) ≈1 and sin(α) ≈ α, to calculate $\vec{\boldsymbol{v}}_{\boldsymbol{cg}}.\vec{\boldsymbol{v}}_{\boldsymbol{cg}}$ we will have :

$$\overrightarrow{V_{cg}} \cdot \overrightarrow{V_{cg}} = \dot{h}^2 + b^2 x_a^2 \dot{\alpha}^2 + 2bx_\alpha \dot{h}\dot{\alpha} \quad \text{(II.8)}$$

Where: x_a=(e-a) is the static imbalance parameter.

We substitute the relation (II.8) in (II.2) we obtain:

$$T = \frac{1}{2}m\left(\dot{h}^2 + b^2x_a^2\dot{\alpha}^2 + 2bx_\alpha\dot{h}\dot{\alpha}\right) + \frac{1}{2}I_{cg}\dot{\alpha}^2 \qquad \text{(II.9)}$$

The relationship between the moment of inertia around the elastic axis I_{ea} and the moment of inertia around the center of gravity I_{ea} is given by [28]:

$$I_{ea} = I_{cg} + mb^2x_a^2(cos(\alpha))^2 \cong I_{cg} + mb^2x_a^2 \qquad \text{(II.10)}$$

We replace the value of in (II.8) we find:

$$T = \frac{1}{2}m\left(\dot{h}^2 + 2bx_\alpha\dot{h}\dot{\alpha}\right) + \frac{1}{2}I_{ea}\dot{\alpha}^2 \qquad \text{(II.11)}$$

II.2.3. Generalized forces:

Generalized forces are calculated using the virtual work method where the work given by a virtual displacement due to external forces.

The total virtual work done by the lift force L and the moment M is given by:

$$\delta w = Q_h\delta h + Q_\alpha\delta\alpha \qquad \text{(II.12)}$$

With:

δh: The virtual vertical displacement.

$\delta\alpha$: The virtual pitch displacement.

To calculate the virtual displacement due to the lift force, the velocity of the aerodynamic center is needed [28]:

$$\overrightarrow{V_{ca}} = -\dot{h}\overrightarrow{l_2} + \dot{\alpha}b\left(\frac{1}{2} + \alpha\right)\overrightarrow{b_2} \qquad \text{(II.13)}$$

Then, the virtual displacement due to lift can be obtained simply by replacing the point above each variable in equation (II.14) by δ in front of it:

$$\overrightarrow{\delta P_{ca}} = -\delta h\overrightarrow{l_2} + b\delta\alpha\left(\frac{1}{2} + \alpha\right)\overrightarrow{b_2} \qquad \text{(II.14)}$$

So the virtual work due to the lift force is:

$$\delta w_p = L\left[-\delta h + b\left(\frac{1}{2} + \alpha\right)\delta\alpha\right] \qquad \text{(II.15)}$$

Where: L is the lift force.

The angular velocity of the wing is $\dot{\alpha}\,\vec{b}_3$ which gives the virtual rotation due to the moment [28]:

$$\delta R_{ca} = -\delta\alpha\overrightarrow{b_3} \qquad \text{(II.16)}$$

So, the virtual work due to moment M is:

$$\delta w_m = M\delta\alpha \qquad \text{(II.17)}$$

The total virtual work done by the lift force L and the moment M is therefore:

$$\delta w = \delta w_p + \delta w_m = L\left[-\delta h + b\left(\frac{1}{2} + \alpha\right)\delta\alpha\right] + M\delta\alpha \qquad \text{(II.18)}$$

By correspondence of two equations (II.12) and (II.18), the generalized forces become:

$$\begin{aligned} Q_h &= -L \\ Q_\alpha &= M + x_\beta \cdot L \end{aligned} \qquad \text{(II.19)}$$

With:

$$x_\beta = b\left(\frac{1}{2} + \alpha\right) \qquad \text{(II.20)}$$

Now the equations of motion can be obtained by combining all the pieces, and using Lagrange's equations as follows:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial(T-U)}{\partial\dot{h}}\right) - \frac{\partial(T-U)}{\partial h} &= Q_h \\ \frac{d}{dt}\left(\frac{\partial(T-U)}{\partial\dot{\alpha}}\right) - \frac{\partial(T-U)}{\partial\alpha} &= Q_\alpha \end{aligned} \qquad \text{(II.21)}$$

After derivation, the equations of motions become:

$$\begin{aligned} m\ddot{h} + mx_\alpha b\ddot{\alpha} + k_h h &= -L \\ mbx_\alpha \ddot{h} + I_{ea}\ddot{\alpha} + k_\alpha &= M \end{aligned} \qquad \text{(II.22)}$$

To model the effects of viscous damping, the Rayleigh dissipation function used is [3]:

$$d = \frac{1}{2} C_h \dot{h}^2 + \frac{1}{2} C_\alpha \dot{\alpha}^2 \qquad \text{(II.23)}$$

Where: C_h and C_α are damping coefficients for the two motions, vertical and pitch respectively.

The relation (II.23) is included in the Lagrange equations as follows:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial(T-U)}{\partial \dot{h}}\right) + \frac{\partial d}{\partial \dot{h}} - \frac{\partial(T-U)}{\partial h} &= Q_h \\ \frac{d}{dt}\left(\frac{\partial(T-U)}{\partial \dot{\alpha}}\right) + \frac{\partial d}{\partial \dot{\alpha}} - \frac{\partial(T-U)}{\partial \alpha} &= Q_\alpha \end{aligned} \qquad \text{(II.24)}$$

By substituting equations (II.1), (II.11), (II.18) and (II.23), in equations (II.24) we finally obtain the aeroelastic equations of motion with two degrees of freedom for vertical motion and pitch motion.

$$m\ddot{h} + mx_\alpha b\ddot{\alpha} + C_h\dot{h} + k_h h = -L \qquad \text{(II.25)}$$

$$mbx_\alpha \ddot{h} + I_{ea}\ddot{\alpha} + k_\alpha \alpha + C_\alpha \dot{\alpha} = M \qquad \text{(II.26)}$$

I: the inertia matrix

F: the damping matrix

E: the stiffness matrix

II.3. Aerodynamic model:

The quasi-stationary aerodynamic force L and moment M for a wing with both trailing edge and leading-edge control surfaces (See Figure (II.2)) is given as follows [29]:

$$L(t) = \rho V^2 bSC_{l\alpha}\left[\alpha + \left(\frac{\dot{h}}{v}\right) + \left(\frac{1}{2} - a\right) b\left(\frac{\dot{a}}{v}\right)\right] + \rho V^2 bSC_{l\beta}\beta + \rho V^2 bSC_{l\gamma}\gamma \qquad \text{(II.27)}$$

$$M(t)= \rho V^2 b^2 s C_{m\alpha}[\alpha+(\frac{\dot{h}}{v})+(\frac{1}{2}-a)b(\frac{\dot{\alpha}}{v})]+\rho V^2 b^2 s C_{m\beta}\beta+ \rho V^2 b^2 s C_{m\gamma}\gamma \qquad (II.28)$$

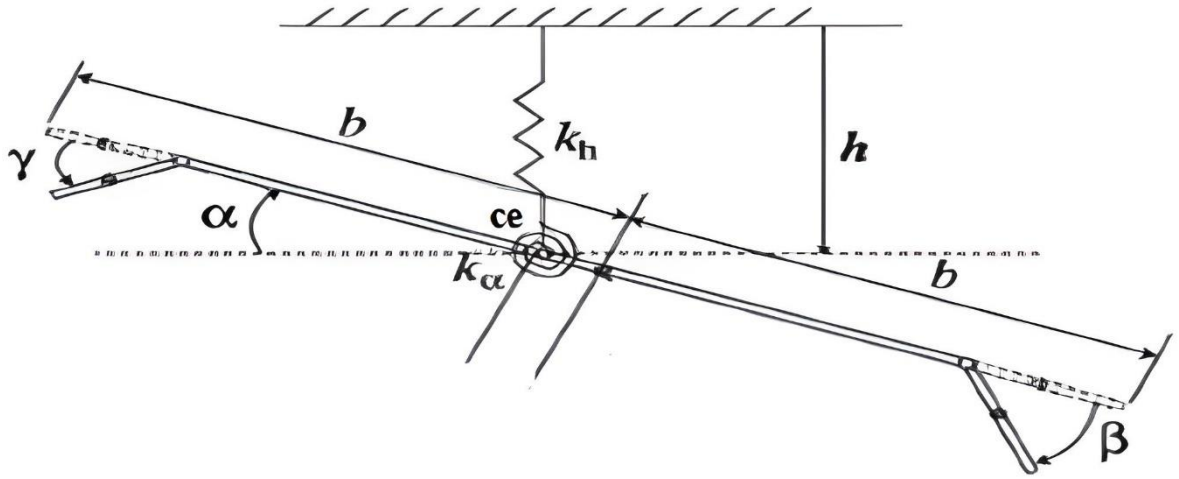

Figure II.2: The aeroelastic model of a two-dimensional wing section with the two leading edge and trailing edge control surfaces.[29]

II.4. Application on the TAMU WING II wing model :

The model used for our study is called TAMU WING II, it was established in the University of Texas, this model is tested in a wind tunnel at low speed to study the effectiveness of the suppression of LCO (Limit cycle oscillations) on the linear aeroelastic system.

In this model used, there are two FUTABA S9402 servo motors operate the control surfaces, each motor capable of producing a torque of 0.654 N.m at 5 V, with control surface deflections linearly proportional to the applied voltage.

In addition, two optical encoders from E2-1024-375-h are mounted on the rotation axes of the leading edge and trailing edge control surfaces to allow measurement of rudder deflection angles so that they can be compared to ordered inputs [29].

The length of the leading edge and trailing edge surfaces are 15 and 20% of the average chord length respectively.

The equations of motion (II.25) and (II.26) for this model are given as follows [29]:

$$m_T\ddot{h} + m_w x_\alpha b\ddot{\alpha} + C_h h + k_h h = -L \qquad (II.29)$$

$$m_w x_\alpha b\ddot{h} + I_{ea}\ddot{\alpha} + k_\alpha \alpha + C_\alpha \dot{\alpha} = M \qquad \text{(II.30)}$$

With:

m_T: The total mass of the wing and its supporting structure

m_W: The mass of the wing only.

II.4.1. TAMU WING II Parameters:

To carry out a numerical application on the equations of motion, the values used are given in the following table:

Le paramètre	La valeur	L'unité
P	1.225	Kg/m^3
A	-0.6719	Sans unité
B	0.1905	M
x_α	- (0.0998 + a)	Sans unité
S	0.5945	M
k_h	2844.4	N/m
k_α	3.525	Nm/rad
C_h	27.43	Kg/s
C_α	0.0360	Kg.m^2/s
m_w	5.230	Kg
m_T	15.57	Kg
I_{ea}	0.14193	Kg.m^2
$C_{l\alpha}$	6.757	Sans unité
$C_{m\alpha}$	(0.5+a) $C_{l\alpha}$	Sans unité
$C_{l\beta}$	3.774	Sans unité
$C_{m\beta}$	-0.6719	Sans unité
$C_{l\gamma}$	-0.1566	Sans unité
$C_{m\gamma}$	-0.1005	Sans unité

Table II.1: The parameters of TAMU WING II [29]

II.5. The TAMU WING II model state representation :

To obtain the state representation of our system, we substitute the equation (II.27) in (II.) we obtain:

$$m_T\ddot{h} + m_w x_\alpha b\ddot{\alpha} + C_h\dot{h} + k_h = -\rho V^2\ b\ s\ C_{l\alpha}\left[\alpha + \left(\frac{\dot{h}}{V}\right) + \left(\frac{1}{2} - a\right) b\left(\frac{\dot{\alpha}}{V}\right)\right] - \rho V^2\ \text{b s}\ C_{l\beta} - \rho V^2\ \text{b s}\ C_{l\gamma}\gamma \qquad \text{(II.31)}$$

And we substitute M by its value given by the equation (II.22) in (II.24) we will have:

$$m_w x_\alpha b\ddot{h} + I_{ea}\ddot{\alpha} + k_\alpha\alpha + C_\alpha(\dot{\alpha})\dot{\alpha} = \rho V^2 b^2 s C_{m\alpha}\left[\alpha + \left(\frac{\dot{h}}{V}\right) + \left(\frac{1}{2} - a\right) b\left(\frac{\dot{\alpha}}{V}\right) + \rho V^2 b^2 s C_{m\beta} + \rho V^2 b^2 s C_{m\gamma}\gamma\right. \qquad \text{(II.32)}$$

We combine equation (II.31) with equation (II.32) to eliminate $\ddot{\alpha}$ we will have:

$$\ddot{h}\frac{(m_w x_\alpha b)^2 - m_\tau I_{ea}}{m_w x_\alpha b} + h\frac{-\rho V^2 b^2 s C_{m\alpha} m_w x_\alpha b - I_{ea}[C_h + \rho V^2 b s C_{l\alpha}]}{m_w x_\alpha b} + \dot{\alpha}\frac{C_\alpha m_w x_\alpha b - \rho V^2 b^2 s C_{m\alpha}\left(\frac{1}{2} - a\right) m_w x_\alpha b - \rho V^2 b s C_{la}\left(\frac{1}{2} - a\right) I_{ea}}{m_w x_\alpha b} + \alpha\frac{k_\alpha m_w x_\alpha b - \rho V^2 b^2 s C_{m\alpha} m_w x_\alpha b - \rho V^2 b s C_{l\alpha} I_{ea}}{m_w x_\alpha b} - h\frac{k_h I_{ea}}{m_w x_\alpha b} + \beta\frac{-\rho V^2 b^2 s C_{m\beta} m_w x_\alpha b - \rho V^2 b s C_{l\beta} I_{ea}}{m_w x_\alpha b}\right] + \gamma\left[\frac{-\rho V^2 b^2 s C_{m\gamma} m_w x_\alpha b - \rho V^2 b s C_{l\gamma} I_{ea}}{m_w x_\alpha b}\right] = 0 \qquad \text{(II.33)}$$

the above equation becomes:

$$\ddot{h} = h\frac{-\rho V^2 b^2 s C_{m\alpha} m_w x_\alpha b - I_{ea}[C_h + \rho V^2 b s C_{la}]}{m_T I_{ea} - (m_w x_\alpha b)^2} + \dot{\alpha}\frac{C_\alpha m_w x_\alpha b - \rho V^2 b^2 s C_{ma}\left(\frac{1}{2} - a\right) m_w x_\alpha b - \rho V^2 b s C_{la}\left(\frac{1}{2} - a\right) I_{ea}}{m_T I_{ea} - (m_w x_\alpha b)^2}\right] - h\frac{k_h I_{ea}}{m_T I_{ea} - (m_w x_\alpha b)^2} + \alpha\frac{k_\alpha m_w x_\alpha b - \rho V^2 b^2 s C_{m\alpha} m_w x_\alpha b - \rho V^2 b s C_{l\alpha} I_{ea}}{m_T I_{ea} - (m_w x_\alpha b)^2} + \beta\frac{-\rho V^2 b^2 s C_{m\beta} m_w x_\alpha b - \rho V^2 b s C_{l\beta} I_{ea}}{m_T I_{ea} - (m_w x_\alpha b)^2} + \gamma\left[\frac{-\rho V^2 b^2 s C_{m\gamma} m_w x_{\alpha\alpha} b - \rho V^2 b s C_{l\gamma} I_{ea}}{m_T I_{ea} - (m_w x_\alpha b)^2}\right] = 0 \qquad \text{(II.34)}$$

We can write the above equation as follows:

$$\ddot{h} = a_{31}h + a_{32}\alpha + a_{33}\dot{h} + a_{34}\dot{\alpha} + b_{31}\beta + b_{32}\gamma \qquad \text{(II.35)}$$

By also combining the equation (II.31) with the equation (II.32) by eliminating this time $\ddot{h}$ we obtain:

$$\ddot{\alpha}m_w x_\alpha b - \frac{m_T I_{ea}}{m_w x_\alpha b} + \dot{h}\left[C_h + \frac{\rho V^2 bsC_{l\alpha}}{V} + \frac{m_T \rho V^2 b^2 sC_{m\alpha}}{V m_w x_\alpha b}\right] + \dot{\alpha}\left[\frac{\rho V^2 bsC_{l\alpha}}{V}\left(\frac{1}{2} - a\right)b - \frac{m_T C_\alpha}{m_w x_\alpha b} + \frac{m_T \rho V^2 b^2 sC_{m\alpha}}{V m_w x_\alpha b}\left(\frac{1}{2} - a\right)b\right] + \alpha\left[\rho V^2 bsC_{l\alpha} - \frac{m_T k_\alpha}{m_w x_\alpha b} + \frac{m_T \rho V^2 b^2 sC_{m\alpha}}{m_w x_\alpha b}\right] + k_h h + \beta\left[\rho V^2 bsC_{l\beta} + \frac{\rho V^2 b^2 sC_{m\beta} m_T}{m_w x_\alpha b}\right] + \gamma\left[\rho V^2 bsC_{l\gamma} + \frac{\rho V^2 b^2 sC_{m\gamma} m_T}{m_w x_\alpha b}\right] = 0 \qquad \text{(II.36)}$$

and we calculate $\ddot{\alpha}$ we find it as follows:

$$\ddot{\alpha} = h\frac{C_h m_w x_\alpha b + \rho V^2 bsC_{l\alpha} m_w x_\alpha b m_T \rho V^2 b^2 sC_{m\alpha}}{m_T I_{ea} - (m_w x_\alpha b)^2} + \dot{\alpha}\left[\frac{\rho V^2 bsC_{l\alpha}\left(\frac{1}{2} - a\right)b m_w x_\alpha b m_T C_\alpha(\dot{\alpha}) m_T \rho V^2 b^2 sC_{m\alpha}\left(\frac{1}{2} - a\right)b}{m_T I_{ea} - (m_w x_\alpha b)^2}\right] + h\frac{k_h m_w x_\alpha b}{m_T I_{ea} - (m_w x_\alpha b)^2} + \alpha\frac{\rho V^2 bsC_{l\alpha} m_w x_\alpha b - m_T k_\alpha + m_T \rho V^2 b^2 sC_{m\alpha}}{m_T I_{ea} - (m_w x_\alpha b)^2} + \beta\frac{\rho V^2 bsC_{l\beta} m_w x_\alpha b + \rho V^2 b^2 sC_{m\beta} m_T}{m_T I_{ea} - (m_w x_\alpha b)^2} + \gamma\frac{\rho V^2 bsC_{l\gamma} m_w x_\alpha b + \rho V^2 b^2 sC_{m\gamma} m_T}{m_T I_{ea} - (m_w x_\alpha b)^2} \qquad \text{(II.37)}$$

The above equation can be written as follows:

$$\ddot{\alpha} = a_{41}h + a_{42}\alpha + a_{43}\dot{h} + a_{44}\dot{\alpha} + b_{41}\beta + b_{42}\gamma \qquad \text{(II.38)}$$

Equations (II.35) and (II.38) can be written in the form:

$$\begin{aligned} \dot{X} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \qquad \text{(II.39)}$$

With: x is the state vector given by:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} \tag{II.40}$$

u: is the command vector given by:

$$u = \begin{bmatrix} \beta \\ \gamma \end{bmatrix} \tag{II.41}$$

y: is the output vector given by:

$$y = \begin{bmatrix} h \\ \alpha \end{bmatrix} \tag{II.42}$$

A, B are the state and input matrices given respectively:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix}$$

C, D are the output and control matrices given respectively:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

We notice that our aeroelastic system is of 4th order and it is described by a mathematical model with invariable linear parameters.

Conclusion:

Finally, the chapter on mathematical modulization of the system has provided useful insights on the power and significance of mathematical modeling in understanding and analyzing complex systems. We were able to develop a simplified version of the system that captures its basic properties and behaviors by employing mathematical approaches and ideas.

The mathematical modeling of our model is based on physical laws, essentially Lagrange's formalism which takes into account only the forces having a work.

Modeling has enabled us to develop equations and relationships that characterize the dynamics and interactions inside the system, allowing us to obtain a better understanding of its fundamental mechanics, to use it for doing the simulation and study our model.

CHAPTER III:

Adaptive sliding mode control

Introduction:

Control systems are used to regulate the behavior of a system in order to achieve a desired outcome. They are used in a wide variety of applications, such as robotics, manufacturing, and aviation.

Control systems must be adaptive and durable in order to be effective: The ability of a control system to adjust to changes in the system being governed is referred to as adaptability, while the ability of a control system to retain its performance in the presence of disruptions is referred to as robustness. Adaptability and robustness are important for control systems for a variety of reasons, including: systems being controlled are frequently subject to change, and disturbances are often unavoidable; if a control system is not robust, it may not be able to maintain its performance in the presence of these disturbances. To solve this issue the most beneficial technique for improving the flexibility and resilience of control systems is to employ an adaptive controller.

By learning the system dynamics and modifying the control input accordingly, neural networks have the ability to achieve adaptive control. This can be accomplished by training the neural network on data that contains system dynamics as well as the required control input. Once trained, the neural network can be used to operate the system in real time, and as neural networks evolve, they are likely to become more efficient, robust, and easy to comprehend. The radial basic functions neural network is one of the most useful and functional types in this field.

Adaptive sliding mode control (ASMC) is a flexible control approach that may be used to a wide range of systems. It is based on the sliding mode control (SMC) approach, a discontinuous control technique noted for its resistance to disturbances and capacity to deal with uncertainty.

ASMC use a neural network to estimate the system's unknown parameters. The estimated parameters are then used to create a resilient sliding mode controller to disturbances and uncertainty.

ASMC has a number of advantages over other control techniques. Some of the advantages of adaptive sliding mode control are:

- **Robustness to disturbances:** ASMC is robust to disturbances, which means that it can still perform well even when the system is subjected to unexpected changes.

- **Ability to handle uncertainty:** ASMC can handle uncertainty, which means that it can still perform well even when the system parameters are not known exactly.
- **Simple implementation:** ASMC is relatively simple to implement, which makes it a good choice for systems where implementation complexity is a concern.
- **Wide range of applications:** ASMC can be used to control a wide variety of systems, including robotic manipulators, power systems, and aircraft.

In this chapter, we will explain how the adaptive sliding mode controller works and how we can use an RBF neural network to adapt the SMC by approximating the unknown system dynamics and estimate its uncertain parameters.

III.1. What is a neural network:

Work on artificial neural networks, sometimes known as "neural networks," has been spurred from the start by the realization that the human brain computes in a completely different way than a normal digital computer. The brain is a computer (information-processing machine) that is extremely complex, nonlinear, and parallel. It is capable of organizing its structural parts, known as neurons, in order to do certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer currently in existence. [33]

A brain has significant structure and the potential to establish its own norms of behavior at birth, which we commonly refer to as "experience." Indeed, experience accumulates over time, with much of the growth (i.e., hardwiring) of the human brain occurring during the first two years after birth, but development continues much beyond that point. [33]

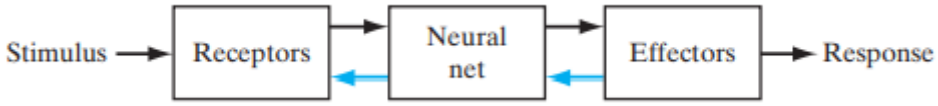

Figure III.1: Block diagram representation of nervous system.[33]

A neural network, in its most general form, is a system meant to imitate the way the brain performs a specific activity or function of interest; the network is often constructed using electronic components or is simulated in software on a digital computer. Neural networks use a huge interconnection of basic computer cells known as "neurons" or "processing units" to attain high performance.

The complex structure of the brain is not approached by artificial neural networks. However, there are two major parallels between biological and artificial neural networks.

First, the basic elements of both networks are simple computational devices that are strongly interconnected (albeit artificial neurons are far simpler than biological neurons). Second, the network's function is determined by the connections between neurons. [34]

III.2. Neuron model and network architectures:

A neuron is a crucial information-processing unit in the operation of a neural network. The block diagram in Fig. III.2 depicts a neuron model, which serves as the foundation for creating a broad family of neural networks. [33]

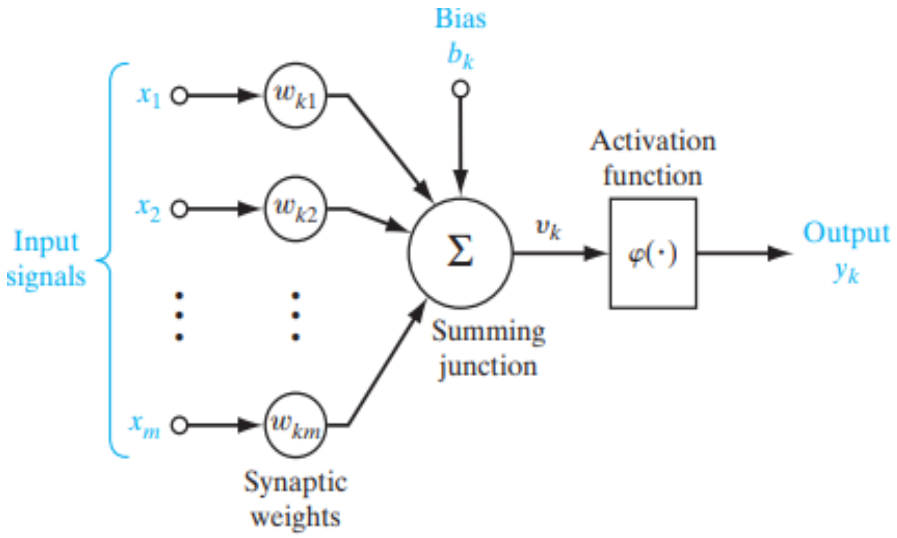

Figure III.2: Nonlinear model of a neuron.[33]

Here, we identify three basic elements of the neural model [33]:

- A group of synapses or connecting links, each of which has its own weight or strength. In particular, the synaptic weight wkj is multiplied by a signal xj at the input of synapse j connected to neuron k. It is critical to take attention of how the subscripts of the synaptic weight wkj are written. The first subscript in wkj denotes the neuron in question, while the second denotes the input end of the synapse to which the weight relates. Unlike the weight of a synapse in the brain, the synaptic weight of an artificial neuron can have both negative and positive values.
- An adder for adding the input signals, which are weighted by the neuron's individual synaptic strengths; the operations detailed here create a linear combiner.
- An activation function that limits the magnitude of a neuron's output. The activation function is also known as a squashing function since it squashes (restricts) the output signal's allowed amplitude range to some limited value.

The neural model of Fig. III.2 also includes an externally applied bias, denoted by b_k. The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively. In mathematical terms, we may describe the neuron k depicted in Fig. III.2 by writing the pair of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad \text{(III.1)}$$

And:

$$y_k = \varphi(u_k + b_k) \qquad \text{(III.2)}$$

Where $x_1, x_2, ..., x_m$ are the input signals; $w_{k1}, w_{k2}, ..., w_{km}$ are the respective synaptic weights of neuron k; u_k (not shown in Fig. III.2) is the linear combiner output due to the input signals; b_k is the bias; $\varphi(\cdot)$ is the activation function; and y_k is the output signal of the neuron. The bias b_k is an external parameter of neuron k, and the use of bias b_k has the effect of applying an affine transformation to the output u_k of the linear combiner in t.[33]

III.2.1. Neuron model:

We distinguish three types of neuron model, which are:

III.2.1.1. Single-Input Neuron:

A single-input neuron is shown in Figure III.3. The scalar input is multiplied by the scalar weight to form, one of the terms that is sent to the summer. The other input, 1 is multiplied by a bias (offset) and then passed to the summer. The summer output, often referred to as the net input, goes into a transfer function (activation function), which produces the scalar neuron output.[34]

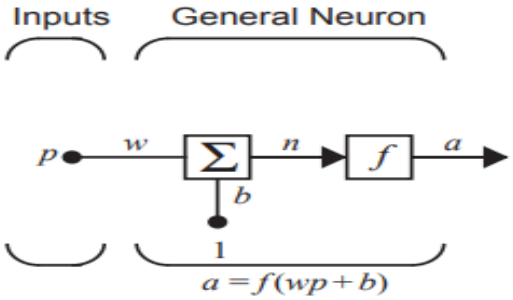

Figure III.3: single-input neuron[34]

The neuron output is calculated as:

$$a = f(wp + b)$$

The actual output is determined by the transfer function used. The bias is similar to a weight, however it has a constant input of 1. If you do not want to have a bias in a specific neuron, it can be omitted. It should be noted that w and b are both scalar neuron parameters that can be adjusted. Typically, the designer chooses the transfer function, and then the parameters are updated by some learning method so that the neuron input/output relationship fulfills some specific purpose. We have different transfer functions for different reasons, as stated in the next section. [34]

III.2.1.2. Transfer functions:

The transfer function in Figure III.3 may be a linear or a nonlinear function of n. A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve.

Most of the transfer functions used in neural networks are summarized in Table III.1:

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1+e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin

Table III.1: Transfer Functions[34].

III.2.1.3. Multiple input neuron:

Typically, a neuron has more than one input. A neuron with inputs is shown in Figure III.7. The individual inputs p_1, p_2, …, p_R are each weighted by corresponding elements w_1, w_2, …, w_R of the weight matrix W.

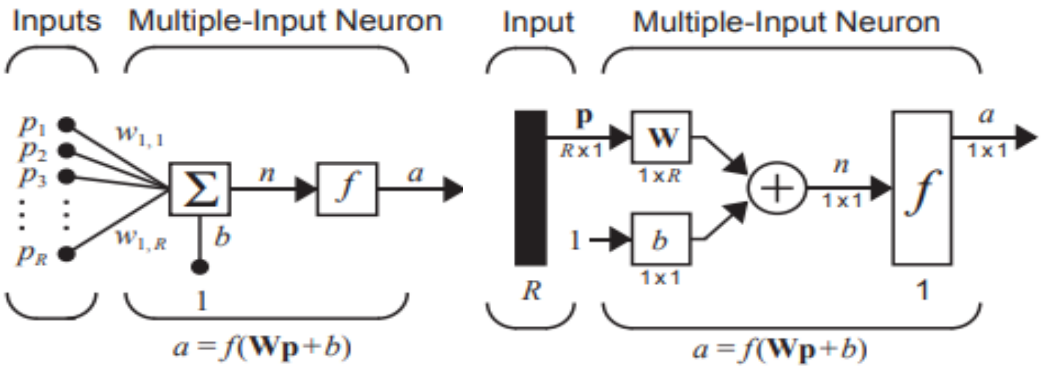

Figure III.7: Neuron with R Inputs[34].

The neuron has a bias b, which is summed with the weighted inputs to form the net input n:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \cdots + w_{1,R}p_R + b \qquad \text{(III.3)}$$

This expression can be written in matrix form:

$$n = Wp + b \qquad \text{(III.4)}$$

Where the matrix W for the single neuron case has only one row. Now the neuron output can be written as:

$$a = f(Wp + b) \qquad \text{(III.5)}$$

As shown in Figure III.7, the input vector p is represented by the solid vertical bar at the left. The dimensions of p are displayed below the variable as R*1, indicating that the input is a single vector of R elements. These inputs go to the weight matrix **W**, which has R columns but only one row in this single neuron case. A constant 1 enters the neuron as an input and is multiplied by a scalar bias b. The net input to the transfer function f is n, which is the sum of the bias b and the product **W.p** . The neuron's output a is a scalar in this case. If we had more than one neuron, the network output would be a vector.[34]

It is important to note that the number of inputs to a network is determined by the problem's external specifications. For example, if you wish to create a neural network to predict kite-flying conditions and the inputs are air temperature, wind velocity, and humidity, the network would have three inputs. [34]

III.2.2. Network architectures:

Commonly one neuron, even with many inputs, may not be sufficient. We might need five or ten, operating in parallel, in what we will call a "layer."

III.2.2.1. A layer of neurons:

A single-layer network of neurons S is shown in Figure III.8. Note that each of the R inputs is connected to each of the neurons and that the weight matrix now has S rows.[34]

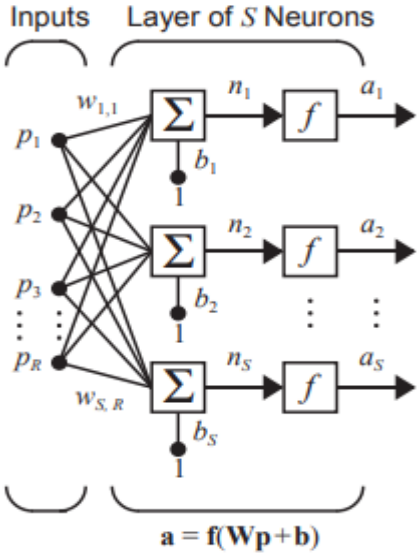

Figure III.8: Layer of S Neurons[34].

The weight matrix, summers, bias vector b, transfer function boxes, and output vector an are all part of the layer. The weight matrix W connects each member of the input vector p to each neuron. A bias b_i, a summer, a transfer function f, and an output a_i are all present in each neuron. When the outputs are added together, they generate the output vector a. The number of inputs to a layer is frequently greater than the number of neurons (i.e., RS). You may wonder whether all neurons in a layer must have the same transfer function. The answer is no; by combining two of the networks illustrated above in parallel, you can build a single (composite) layer of neurons with various transfer functions. Both networks would use the same inputs and generate part of the same outputs. [34]

The input vector elements enter the network through the weight matrix W:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \qquad \text{(III.6)}$$

As noted previously, the row indices of the elements of matrix indicate the destination neuron associated with that weight W, while the column indices indicate the source of the input for that weight. Thus, the indices in $w_{3,2}$ say that this weight represents the connection to the third neuron from the second source. Fortunately, the S-neuron, R-input, one-layer network also can be drawn in abbreviated notation, as shown in Figure III.9.[34]

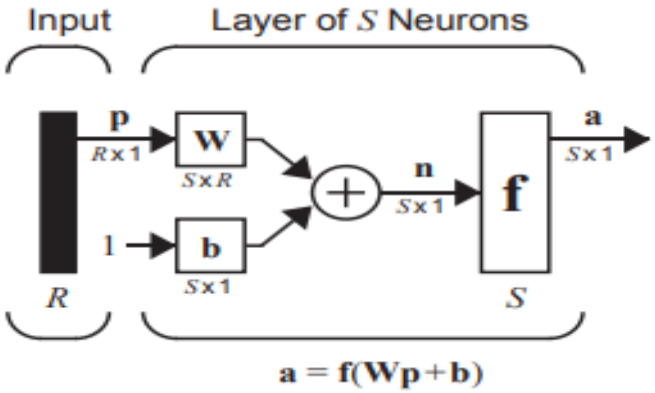

Figure III.9: Layer of Neurons[34]

III.2.2.2. Multiple layers of neurons:

Now consider a network with several layers. Each layer has its own weight matrix **W**, its own bias vector **b**, a net input vector **n** and an output vector **a**. We need to introduce some additional notation to distinguish between these layers. We will use superscripts to identify the layers. Specifically, we append the number of the layer as a superscript to the names for each of these variables. Thus, the weight matrix for the first layer is written as $\mathbf{W^1}$, and the weight matrix for the second layer is written as $\mathbf{W^2}$**.** This notation is used in the three-layer network shown in Figure III.10.[34]

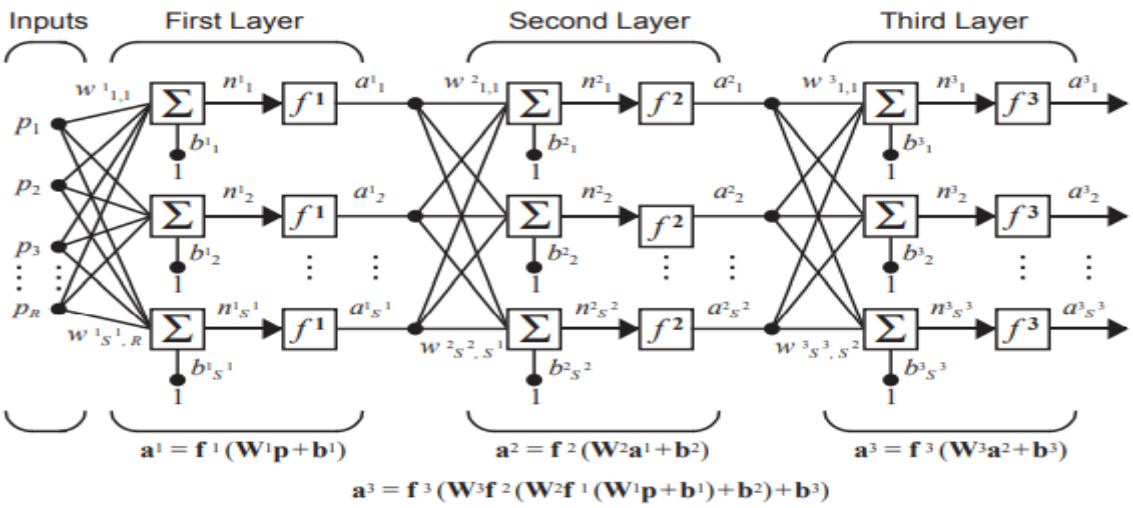

Figure III.10: Three-Layer Network[34]

As shown, there are **R** inputs, $\mathbf{S^1}$ neurons in the first layer, $\mathbf{S^2}$ neurons in the second layer, etc. As noted, different layers can have different numbers of neurons. The outputs of layers one and two are the inputs for layers two and three. Thus layer 2 can be viewed as a one-layer network with $\mathbf{R=S^1}$ inputs, $\mathbf{S=S^2}$ neurons, and an $\mathbf{S^2*S^1}$ weight matrix $\mathbf{W^2}$. The input to layer 2 is $\mathbf{a^1}$, and the output is $\mathbf{a^2}$. A layer whose output is the network output is called an output layer. The other layers are called hidden layers.[34]

Multilayer networks outperform single-layer networks. A two-layer network with a sigmoid first layer and a linear second layer, for example, can be taught to approximate most functions arbitrarily well. This is not possible with single-layer networks.

III.2.2.3. Recurent networks:

A recurrent network is a feedback network, having some of its outputs connected to its inputs. This is in stark contrast to the networks we've explored thus far, which were all feedforward with no backward connections. Figure III.11 depicts one type of discrete-time recurrent network. [34]

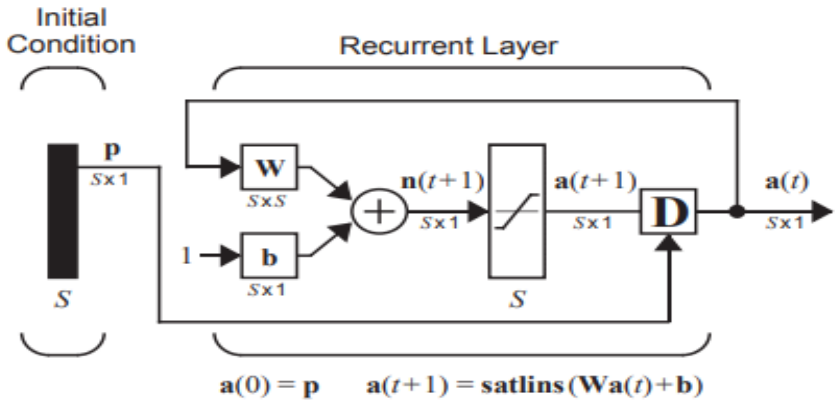

Figure III.11: Recurrent Network[34]

In this particular network the vector **p** supplies the initial conditions (i.e., a(0)=p). Then future outputs of the network are computed from previous outputs:

$$\mathbf{a}(1) = \text{satlins}(\mathbf{W}\mathbf{a}(0) + \mathbf{b}), \mathbf{a}(2) = \text{satlins}(\mathbf{W}\mathbf{a}(1) + \mathbf{b}), \ldots$$

Recurrent networks are potentially more powerful than feedforward networks and can exhibit temporal behavior.

III.3. RBF neural network:

A radial basis function neural network (RBF neural network) is a form of artificial neural network used for function approximation, classification, and regression. Artificial neural networks were proposed for the adaptive control of nonlinear dynamical systems for the first time in 1990. Since then, multilayer neural networks (MNN) and radial basis function (RBF) networks have been used in a variety of identification and control applications. RBF neural networks were addressed in 1988 and have recently gained popularity because to their high generalization ability and simple network topology, which avoids superfluous and costly calculations when compared to the multilayer feed-forward network (MFN). RBF neural networks may approximate any nonlinear function over a compact set with arbitrary accuracy, according to previous research on universal approximation theorems. Significant research has been conducted on RBF neural control for nonlinear systems. [35]

III.3.1. Architecture and algorithms of RBF network:

There are three layers in an RBF neural network: the input layer, the hidden layer, and the output layer. The input layer is made up of neurons that receive input data. The input layer has the same number of neurons as the number of features in the input data. A radial basis function, a non-linear function that calculates the distance between a given input vector

and a set of specified centers, activates neurons in the buried layer. In most cases, the centers are chosen to be at the center of each cluster in the input data: The hidden layer is made up of an array of computing units known as hidden nodes. Each hidden node has a center c vector, which is a parameter vector with the same dimension as the network input vector x; the Euclidean distance between the center and the network input vector x is defined as ||x(t)-c(t)||. The hidden layer's output is a linear combination of the radial basis functions, and the linear combination's weights are set by training the network. The output layer then utilizes a linear function to transfer the hidden layer's output to the desired output, and the linear function's weights are likewise determined by training the network. [35]

The output layer is made up of a group of linear neurons that generate the network's output. The output layer has the same number of neurons as the number of classes in the output data. The output of the hidden layer can be produced using the nonlinear activation function hj(t), as shown below[35]:

$$h_j(t) = exp\left(-\frac{\|x(t)-c_j(t)\|^2}{2b_j^2}\right), j = 1, \dots, m \qquad \text{(III.7)}$$

Where bj notes a positive scalar called a width and m notes the number of hidden nodes. The output layer is a linear weighted combination as follows [35]:

$$y_i(t) = \sum_{j=1}^{m} w_{ji} h_j(t), i = 1, \dots, n \qquad \text{(III.8)}$$

Where w_{ji} are the output layer weights, n notes the number of outputs, and y notes the network output.[35]

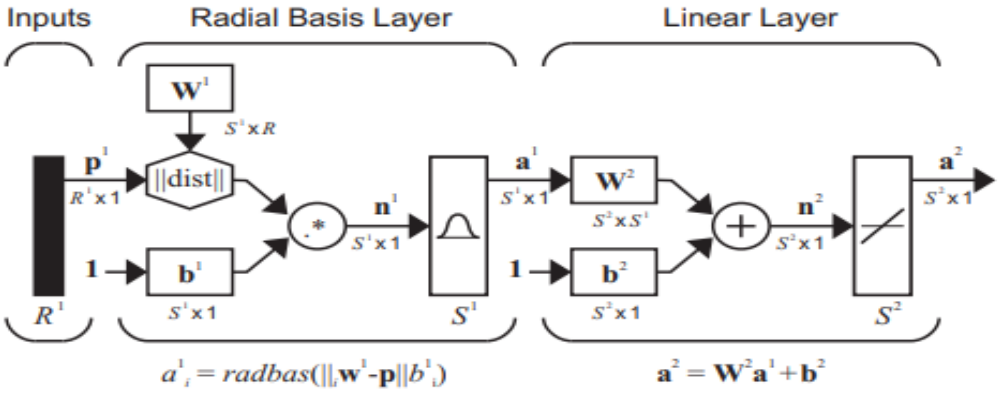

Figure III.12: Structure of an RBF network.[34]

The radial basis function (RBF) network and a two-layer perceptron network differ significantly in two ways. First, instead of conducting an inner product operation between the weights and the input (matrix multiplication) in layer 1 of the RBF network, we calculate the distance between the input vector and the rows of the weight matrix. Second, rather of adding the bias, we multiply by it. The transfer functions employed in the RBF network's first layer differ from the sigmoid functions commonly utilized in the hidden layers of multilayer perceptrons (MLP). There are various types of transfer functions that can be employed, but for the sake of clarity, we will just explore the Gaussian function, which is the most often utilized in the neural network community. This function's major feature is that it is local. This indicates that if you walk extremely far in any way from the center point, the output is close to zero. In contrast, the output of global sigmoid functions remains close to 1 as the net input approaches infinity. [34]

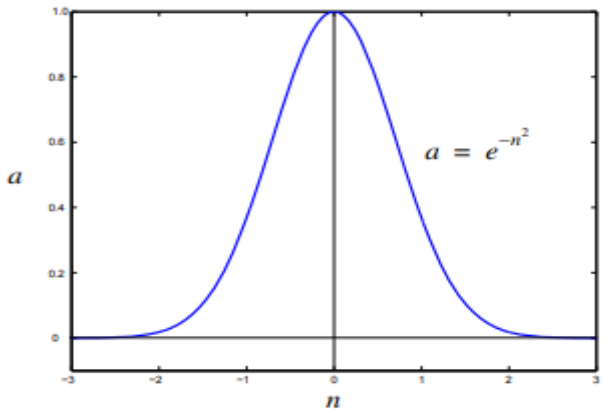

Figure III.13: Gaussian Basis Function.[33]

III.3.2. Function approximation:

Function approximation is a sort of machine learning in which a function is found that may be used to estimate the output for a given input. The purpose of function approximation is to find a function that is as near to the original function as feasible while being basic and simple to grasp. RBFNNs are an effective function approximation tool because they can estimate a wide range of functions, including some that are difficult to approximate with other forms of neural networks.

Consider a nonlinear input–output mapping described by the functional relationship [33]:

$$d=f(x) \qquad \text{(III.9)}$$

where the vector x is the input and the vector d is the output. The vector-valued function f(·) is assumed to be unknown. To make up for the lack of knowledge about the function f(·), we are given the set of labeled examples [33]:

$$\mathcal{T} = \{(x_i, d_i)\}_{i=1}^{N} \qquad \text{(III.10)}$$

The requirement is to design a neural network that approximates the unknown function f(·) such that the function F(·) describing the input–output mapping actually realized by the network, is close enough to f(·) in a Euclidean sense over all inputs, as shown by [33]:

$$\| F(x) - f(x) \| < \varepsilon \quad \textit{for all } x \qquad \text{(III.11)}$$

Where ε is a small positive number. Provided that the size N of the training sample is large enough and the network is equipped with an adequate number of free parameters, then the approximation error can be made small enough for the task. The approximation problem described here is a perfect candidate for supervised learning, with xi playing the role of input vector and di serving the role of desired response. We may turn this issue around and view supervised learning as an approximation problem.[33]

The ability of a neural network to approximate an unknown input–output mapping may be exploited in two important ways:

- **System identification**. Let Eq. (III.9) describe the input–output relation of an unknown memoryless multiple input–multiple output (MIMO) system; by a "memoryless" system, we mean a system that is time invariant. We may then use the set of labeled examples in Eq. (III.10) to train a neural network as a model of the system. Let the vector y_i denote the actual output of the neural network produced in response to an input vector x_i. The difference between d_i (associated with x_i) and the network output y_i provides the error signal vector e_i, as depicted in Fig. III.13. This error signal is, in turn, used to adjust the free parameters of the network to minimize the squared difference between the outputs of the unknown system and the neural network in a statistical sense, and is computed over the entire training sample.[33]

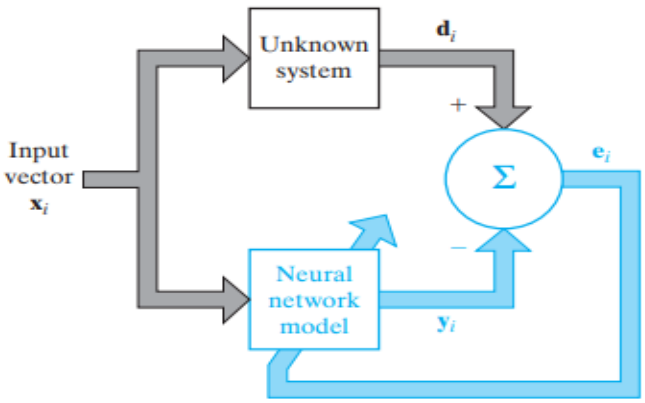

Figure III.14: Block Diagram of system identification.[33]

- **Inverse modeling**. Suppose next we are given a known memoryless MIMO system whose input–output relation is described by Eq. (III.1). The requirement in this case is to construct an inverse model that produces the vector x in response to the vector d. The inverse system may thus be described by [33]:

$$x = f^{-1}(d) \qquad \text{(III.12)}$$

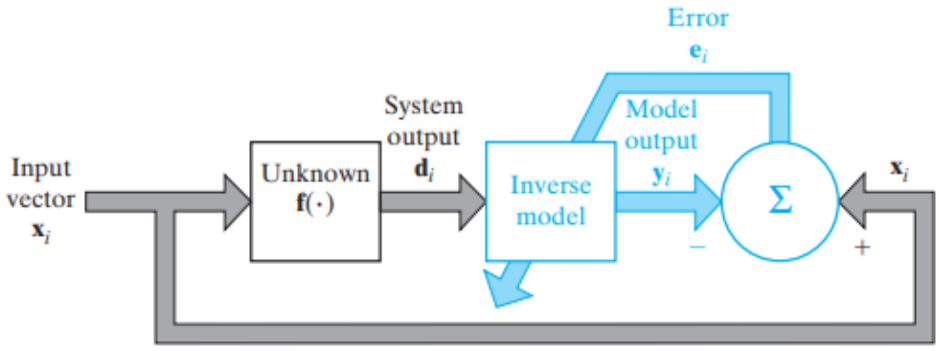

Figure III.15: Block Diagram of inverse system modeling.[33]

Where the vector-valued function $f^{-1}(\cdot)$ denotes the inverse of $f(\cdot)$. Note, however, that $f^{-1}(\cdot)$ is not the reciprocal of $f(\cdot)$; rather, the use of superscript -1 is merely a flag to indicate an inverse.[33]

III.3.3. Global vs. Local:

Before we go into the training algorithms, let's talk about the benefits and drawbacks of the global transfer functions used by MLP networks and the local transfer functions used by RBF networks. Because all of the transfer functions overlap in their activity, the MLP

generates a distributed representation. Many sigmoid functions in the first layer will have substantial outputs at any given input value. In order to create the right answer at each point, they must sum or cancel in the second layer. Each basis function in the RBF network is only active across a limited portion of the input. Only a few basis functions will be active for each given input. Each strategy has advantages and downsides. Because each neuron contributes to the response over a substantial portion of the input space, the global approach necessitates fewer neurons in the hidden layer. However, for the RBF network to offer an appropriate approximation, basis centers must be distributed across the range of the input space. This leads to the "curse of dimensionality" problem. Furthermore, if more neurons, and thus more parameters, are employed, the network is more likely to overfit the training data and fail to generalize effectively to new scenarios. On the other hand, the local technique often results in faster training, particularly when the two-stage algorithms outlined in the following section are applied. Furthermore, the local technique can be highly beneficial for adaptive training, in which the network is progressively trained while being used, like in adaptive filters or controllers. If training data only appears in one region of the input space for a length of time, a global representation will prefer to enhance its accuracy in those regions at the expense of its representation in other regions. This issue will not be similarly serious in local representations. Because each neuron is only active in a tiny portion of the input space, the weights of the neurons are not modified if the input falls outside of that region. [34]

III.3.4. Training RBF network:

In contrast to the MLP network, which is typically commonly trained using a gradient-based technique, the RBF network can be taught using a variety of methods. Gradient-based methods can be used to train RBF networks. However, due to the local character of the transfer function and the way the first layer weights and biases operate, the error surfaces of RBF networks tend to have many more undesirable local minima than those of MLP networks. As a result, gradient-based techniques are frequently insufficient for fully training RBF networks. They are, nevertheless, utilized on occasion for fine-tuning the network after it has been trained using another method. [34]

The two layers of the RBF network are treated individually in the two stages of the most commonly used RBF training methods. The algorithms differ primarily in the manner in which the first layer weights and biases are chosen. After determining the first layer weights and biases, the second layer weights can be estimated in a single step using a linear least-squares approach (see Appendix 1). The most basic of the two-stage algorithms organizes the

centers (first layer weights) in a grid pattern throughout the input range and then selects a constant bias so that the basis functions overlap to some extent. Because the most efficient approximation would insert more basis functions in regions of the input space where the function to be approximated is most complex, this technique is not ideal. Furthermore, in many real circumstances, the entire range of the input space is not employed, and hence many basis functions may be squandered. One disadvantage of the RBF network, particularly when the centers are chosen on a grid, is that it suffers from the curse of dimensionality. This means that as the dimension of the input space grows, so does the number of basis functions required. For instance, suppose we have one input variable and provide a grid of ten basis functions uniformly spaced across its range. Increase the number of input variables to two now. We would require 102 or 100 basis functions to retain the same grid coverage for both input variables. [34]

Another way for determining the centers is to choose a random subset of the training set's input vectors. This guarantees that the network's base centers are positioned in places where they will be valuable. This technique, however, is not ideal due to the randomness of the selection. [34]

To cluster the input space, a strategy such as the Kohonen competitive layer or the feature map is a more efficient solution. The cluster centers are subsequently transformed into basis function centers (see Appendix 2). This ensures that the core operations are located in high-activity areas. This strategy will be covered in a subsequent section. [34]

The last method for RBF training that we will examine is orthogonal least squares (see Appendix 3). It is based on subset selection, a general method for creating linear models. This approach begins with a large number of potential centers, often all of the training data input vectors. It chooses one center to add to the first layer weight at each stage of the procedure. The new neuron is chosen based on how much it reduces the sum squared error. Neurons are added till a certain threshold is reached. The criteria is often used to maximize the network's generalization capability. [34]

III.4. Sliding mode control:

III.4.1. Introduction and historical aspect of sliding mode control:

Differences between the actual plant and the mathematical model used for controller design are typical in the formulation of any control problem. This discrepancy could be due to unmodeled dynamics, changes in system parameters, or a simple model's approximation of complicated plant behavior. Despite such plant/model mismatches, the engineer must ensure

that the resulting controller is capable of producing the appropriate performance levels in practice. This has sparked great interest in the development of so-called robust control approaches, which aim to tackle this challenge. The sliding mode control methodology is one approach to robust controller design. [36]

Variable structure control is an aspect of sliding mode control. A set of feedback control rules and a decision rule characterize variable structure control systems (VSCS). The switching function is a decision rule that takes some measure of current system behavior as input and outputs the specific feedback controller that should be employed at that point in time. As a result, a variable structure system is formed, which can be thought of as a collection of subsystems, each of which has a fixed control structure and is valid for specified regions of system behavior. One advantage of putting this extra complexity into the system is the possibility to integrate relevant qualities from each of the system's composite structures. Furthermore, the system can be constructed to have novel qualities that are not present in any of the composite structures on their own. In the late 1950s, the Soviet Union began to use these natural notions.[36]

III.4.2. Law control of Sliding Mode Controller:

SMC works by changing the system dynamics into a sliding manifold with stable and predictable system behavior. The control law is intended to keep the system on the sliding manifold despite uncertainties and disturbances.

To develop a law of a sliding mode controller based on Lyapunov's equation for our model. At the beginning we will deposit that:

$$h=x_1 \text{ and } \alpha=x_2, \dot{h}=x_3 \text{ and } \dot{\alpha}=x_4 \qquad \text{(III.13)}$$

while:

$$\dot{x}_1=x_3 \text{ and } \dot{x}_2=x_4. \qquad \text{(III.14)}$$

We have from the mathematical model that we developed in chapter two:

$$\ddot{h}=f_1+u_1 \quad \text{and} \quad \ddot{\alpha}=f_2+u_2 \qquad \text{(III.15)}$$

such that f_1 and f_2 represent the functions of our model, u_1 and u_2 represent the control of trailing edge flap deflection angle (β) and leading-edge flap deflection angle (γ) where we will apply our sliding mode controller.

Now we have the law of sliding mode controller based on Lyapunov's equation is defined by:

$$S_1 = \lambda_1 x_1 + x_3 \quad \text{(III.16)}$$

And: $$S_2 = \lambda_2 x_2 + x_4 \quad \text{(III.17)}$$

Now, after we derivate S_1 and S_2 and we set it equal to zero, the sliding mode control law will be:

$$U_{1eq} = -\lambda_1 x_3 - f_1 \quad \text{(III.18)}$$

$$U_{2eq} = -\lambda_2 x_4 - f_2 \quad \text{(III.19)}$$

While: λ_1 and λ_2 are two constants, such as: $\lambda_1 > 0$ and $\lambda_2 > 0$.

Now, to make sure that our generated law is stable we will apply the Lyapunov theorem ($\dot{S} < 0$), and to make sure that this condition verified we will add a term to our control law that will be:

$$U_1 = -\lambda_1 x_3 - f_1 - tho_1 * sign(S_1) \quad \text{(III.20)}$$

$$U_2 = -\lambda_2 x_4 - f_2 - tho_2 * sign(S_2) \quad \text{(III.21)}$$

While: λ_1 and λ_2 are two constants, such as: $tho_1 > 0$ and $tho_2 > 0$.

III.4.3. Adaptive sliding mode control:

Sliding mode control (SMC), as previously mentioned in this section, is a robust control approach used to govern systems that are vulnerable to uncertainty and disturbances. SMC, on the other hand, can generate chattering, which is a phenomena in which the system output oscillates between two or more states. The discontinuous character of the control law can induce chattering. Chattering is undesirable in some applications because it causes wear and tear on system components [37].

This is why we use adaption to [37]:

- **Chattering:** As previously stated, SMC can create chattering in the system output. This is undesirable in some applications because it causes wear and tear on system components.

- **Uncertainties:** SMC is intended to be resistant to system dynamics uncertainty. In other circumstances, though, the uncertainties may be too great for SMC to handle. In these circumstances, modification can be employed to boost SMC's robustness.
- **Disturbances:** SMC is also intended to be resistant to disruptions. In other circumstances, though, the disruptions may be too great for SMC to handle. In these circumstances, modification can be employed to boost SMC's robustness.

Adaptive control allows the treatment of uncertain dynamic systems, linear and nonlinear, the uncertainties of which can be expressed as the product of an uncertain constant matrix and a vector of known time function [5]:

$$\Theta^* X(t) \qquad \text{(III.22)}$$

that is, θ* is an unknown constant matrix and X is a matrix the entries of which are known functions of the time, where X is defined as the regressor. This situation is encountered in both identification and control problems. In particular for the control of dynamic systems, the techniques that can be applied to systems with uncertainties of this kind rely on two steps procedures. The first step consists of solving the control problem for the system where the matrix θ is regarded as known. The outcome of this step consists in a control law characterized by a specific parametrization that is [5]:

$$u = \Theta_u^*(\Theta^*) X_u(\Theta^*, X, t) \qquad \text{(III.24)}$$

The class of problems which can be dealt with are those for which X_u does not depend on the parameter θ*, but only on the available signal X that is[5]:

$$u^* = \Theta_u^*(\Theta^*) X_u(X) \qquad \text{(III.25)}$$

in the sequel $u^* = \theta_u^* X_U$. The second step is to use a control, in the uncertain θ* case, which has the same parametrization of the ideal u but with time varying parameters θ_u:

$$u = \Theta_u(t) X_u \qquad \text{(III.26)}$$

This actual control signal can be expressed as:

$$u = \Theta_u^* X_u + \tilde{\Theta}_u(t) X_u \qquad \text{(III.27)}$$

The uncertain term $\tilde{\theta}(t)X_u$ is usually called prediction error. This signal plays a fundamental role in the adaptation mechanism where the explicit identification of θ_u^* is required. The regressor vector X_u is usually constituted by known time functions derived from the available system states through linear operations like linear filtering and linear combination. In some cases the X_u components result to be known nonlinear functions of the state. It is well known that the adaptive control scheme can be divided into two categories [5]:

- Direct adaptive control schemes
- Indirect adaptive control schemes

Control schemes in the first category explicitly compare system state trajectories with those of a reference model, translating the predicted ideal behavior of the system, which is active on line during the control process. Despite the system's parametric uncertainties, the control goal is to force some, adequately defined, error function to zero. In principle, achieving this goal does not necessitate identifying the unknown parameters, which in this case are the regulator's parameters. The second category is based on the so-called certainty equivalence principle, which states that the system is recognized using an adaptive approach that produces an estimate of the unknown plant parameter. [5].

III.5. Adaptive RBF neural network sliding mode control:

By approximating an unknown system, a radial basis function (RBF) neural network can efficiently adapt a sliding mode control (SMC) technique. The RBF network can approximate unknown system dynamics, whilst sliding mode control provides robustness and disturbance rejection. RBF networks have successfully approximated a wide range of unknown systems, such as robotic manipulators, power systems, and aviation control systems. They are a powerful tool for approximating nonlinear functions even when the system is poorly understood, and this is related to [35]:

- **Robustness:** RBF networks are resistant to noise and outliers, making them an excellent choice for modeling unknown systems.
- **Accuracy:** RBF networks can be trained to achieve high accuracy, which can make them a good choice for approximating unknown systems.
- **Speed:** RBF networks can be trained quickly, which can make them a good choice for applications where speed is important.

To generate an adaptive control law for our modal, we have:

$$f=\theta^{T}.\Psi(x) \qquad \text{(III.28)}$$

where f defines our ideal model and $\Psi(x)$ defines the initial vector and θ is define the model parameters a_{ij} and b_{ij}, on the other hand we have:

$$\hat{f} = \hat{\theta}.\Psi(x) \qquad \text{(III.29)}$$

Where $\hat{f}$ defines our real-time model that we got from our RBFNN.

Now, we will try to use $\hat{f}$ in our control law instead of f.

We have:

$$\tilde{f}=f-\hat{f}=\tilde{\theta}^{T}.\Psi(x) \qquad \text{(III.30)}$$

We can define the function of Lyapunov by:

$$V=\frac{1}{2}.S^{2}+\frac{1}{2}.\tilde{\theta}^{T}.\Gamma^{T}.\tilde{\theta} \qquad \text{(III.31)}$$

While: S is defined in equations (III.16) and (III.17) and Γ^{T} is a constant

We know that a Lyapunov function be stable if the derivative of that function below 0, so we have the derivative of the equation (III.31) as following:

$$\dot{v}=S(\lambda.x+u+\hat{f})+\tilde{\theta}^{T}(S.\Psi(x)-\Gamma^{-1}.\dot{\hat{\theta}}) \qquad \text{(III.32)}$$

So, we have the derivative of V can be equal 0 if:

$$S.\Psi(x)-\Gamma^{-1}.\dot{\hat{\theta}}=0 \qquad \text{(III.33)}$$

And from equation (III.33) we have:

$$\dot{\hat{\theta}}=\Gamma.S.\Psi(x) \qquad \text{(III.34)}$$

And this is representing the adaptive law.

Now we are able to define $\hat{f}$ because: $\hat{f} = \hat{\theta}.\Psi(x)$

And then we will be able to define the adaptive control law for our model:

$$U=- \lambda x- \hat{f}\text{-tho*sign}(S) \qquad \text{(III.35)}$$

Now let's put all these ideas together into a process [37]:

- **System Identification:** To discover the system dynamics, we collect data from the unknown system or apply a model-based technique. This information will be utilized to train the RBF neural network.
- **RBF Neural Network Design:** Create an RBF network architecture. It has three layers: an input layer, a hidden layer with RBF activation functions, and an output layer. The network's approximation capability is determined by the number of neurons in the hidden layer.
- **Training the RBF Network:** Train the RBF network using the collected data. To decrease approximation error, the training process comprises modifying the network's weights and RBF centers. For this aim, many optimization strategies such as gradient descent or evolutionary algorithms might be applied.
- **Sliding Mode Control Design:** Create a sliding mode control scheme that is appropriate for the system. This entails designing the sliding surface as well as the control rule that guarantees the system tracks the target trajectory while being resilient to disturbances.
- **RBF Adaptation:** Combine the trained RBF network and sliding mode control. In real-time, use the RBF network to approximate the unknown system dynamics. The network takes the present state of the system as input and generates an estimate of the system's dynamics. This estimate is then employed in the sliding mode control law to control the system.
- **System Control:** Use the estimated dynamics of the RBF network to implement the sliding mode control technique. Based on the estimated dynamics and the sliding surface, the control law computes the control signal. This control signal steers the system along the desired path while remaining resilient to perturbations.
- **Adaptation Updates:** Update the RBF network on a regular basis with new data from the system. This allows the network to continuously adapt and improve its approximation of the unknown system dynamics.

This approach effectively handles systems with uncertain dynamics and disturbances by combining the approximation capabilities of the RBF network with the robustness of sliding mode control. The RBF network continuously adjusts to system changes, allowing for precise approximation, while the sliding mode control offers steady and robust control performance.

Conclusion:

Finally, this study looked into combining Radial Basis Function (RBF) neural networks with adaptive sliding mode control to provide robust and adaptive control performance in the presence of uncertainties and disturbances. The proposed RBF neural network adaptive sliding mode controller has various advantages, including higher tracking accuracy, resilience, and adaptation to changing system dynamics. The controller is capable of simulating uncertain system dynamics and estimating unknown parameters using RBF neural networks, allowing for effective adaptation of the sliding mode control parameters. This adaptive capacity enables the controller to continuously adjust and enhance its performance, ensuring effective control even when operating conditions and disturbances change.

The proposed controller's stability analysis, based on Lyapunov stability theory or other comparable approaches, reveals its capacity to provide stability and resilience in the closed-loop system. While limiting the impact of uncertainties and disturbances, the controller ensures dependable and accurate tracking of reference signals.

Future research directions may include examining improved RBF network topologies, including more efficient training algorithms, and investigating optimization strategies to further improve the controller's performance and flexibility. Furthermore, the proposed controller's application to large-scale systems and integration with other control approaches in the field of aviation could be avenues for future research.

CHAPTER IV:

Simulation And Interpretation

Introduction:

In this chapter, we'll apply the sliding mode control rule to our aeroelastic system and show the simulation results in the MATLAB environment.

First, different values of V are applied to our model to determine its behave under multiple circumstances, then we will implement the sliding mode controller to see its effect on our system.

The vertical displacement (h), angle of attack (α), and their velocities ($\dot{h}$ and $\dot{\alpha}$), trailing edge flap deflection angle (β) and leading-edge flap deflection angle (γ), will be shown. and studied graphically, and the flutter speed will be identified for each case.

IV.1. Open loop simulation:

In this section, we will study the stability of the wing section model, by simulating the open loop system while observing the eigenvalues of the state matrix A. Knowing that the model is parameterized by the speed of flow, we will simulate the behavior of the system for different speed values.

The simulation results obtained in this section (open loop model) with the initial conditions $[h\ \alpha\ \dot{h}\ \dot{\alpha}]^T$=[0.01; 0.1; 0; 0]

For a flow velocity V = 10 m/s, the numerical state model of the system is given as follows:

$$\begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -214.1696 & -1.7552 & -2.6784 & -0.1260 \\ 860.0497 & -116.5179 & 8.5929 & -0.2306 \end{bmatrix} \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -3.4054 & 0.2439 \\ 1.1645 & -2.8507 \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \end{bmatrix}$$

The system has the following eigenvalues λ, damping coefficients ε, and frequencies ω:

Eigenvalues λ	Damping coefficients ε	Frequencies ω
-0.6876 ± 13.8950i	0.0494	±13.8950
-0.7669 ± 11.6682i	0.0656	±11.6682

Table IV.1: The characteristics of the open-loop system, for V = 10 m/s.

The system response is shown in Figure (IV.1):

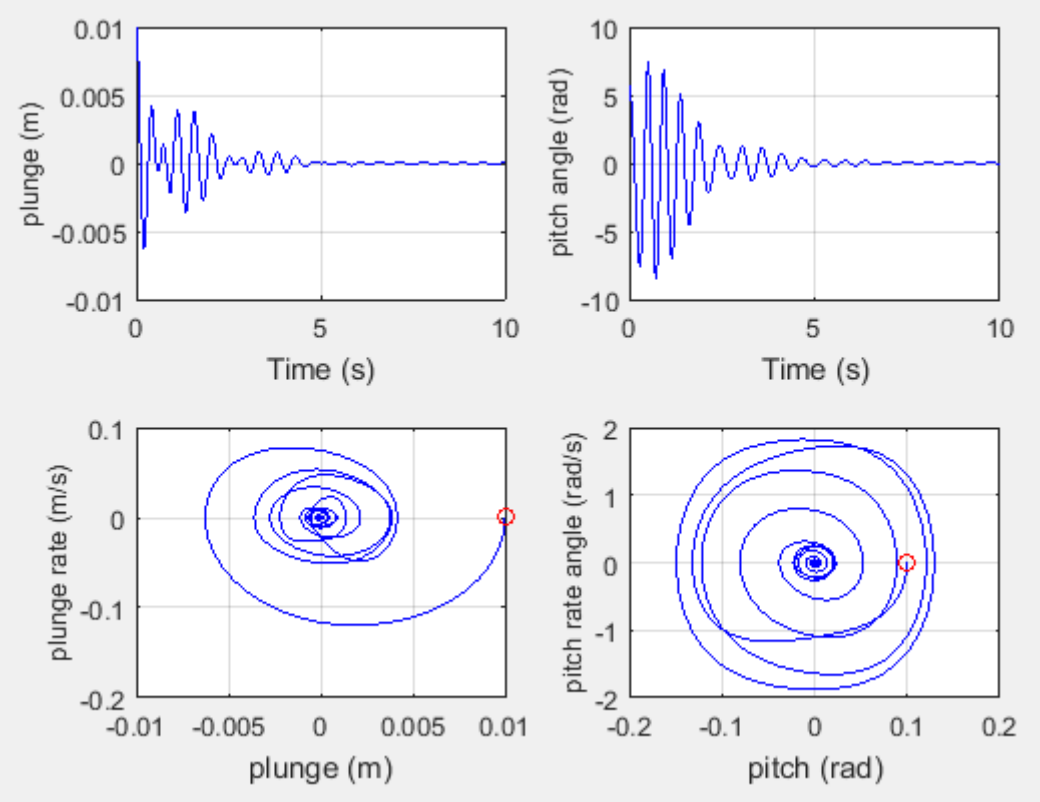

Figure IV.1: The open-loop system response for the vertical displacement h and the pitch angle α and its velocities, at a speed V = 10m/s.

We note that the eigenvalues of the state matrix have a negative real part, so the system is stable, but it is slow (response time 6 seconds) and poorly damped.

For another flow velocity V = 13m/s, the calculated mathematical model is given as follows:

$$\begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -214.1696 & -6.1557 & -2.8623 & -0.1670 \\ 860.0497 & -109.7930 & 8.6826 & -0.2106 \end{bmatrix} \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -5.7551 & 0.4122 \\ 1.9681 & -4.8177 \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \end{bmatrix}$$

In open loop, this process has the following eigenvalues λ, damping coefficients ε and frequencies ω:

Eigenvalues λ	Damping coefficients ε	Frequencies ω
1.2267 ± 13.1390i	-0.0928	±13.1390
-2.7631 ± 12.5618i	0.2147	±12.5618

Table IV.2: The characteristics of the open-loop system, for $V = 13\ m/s$.

The system response is shown in Figure (IV.2):

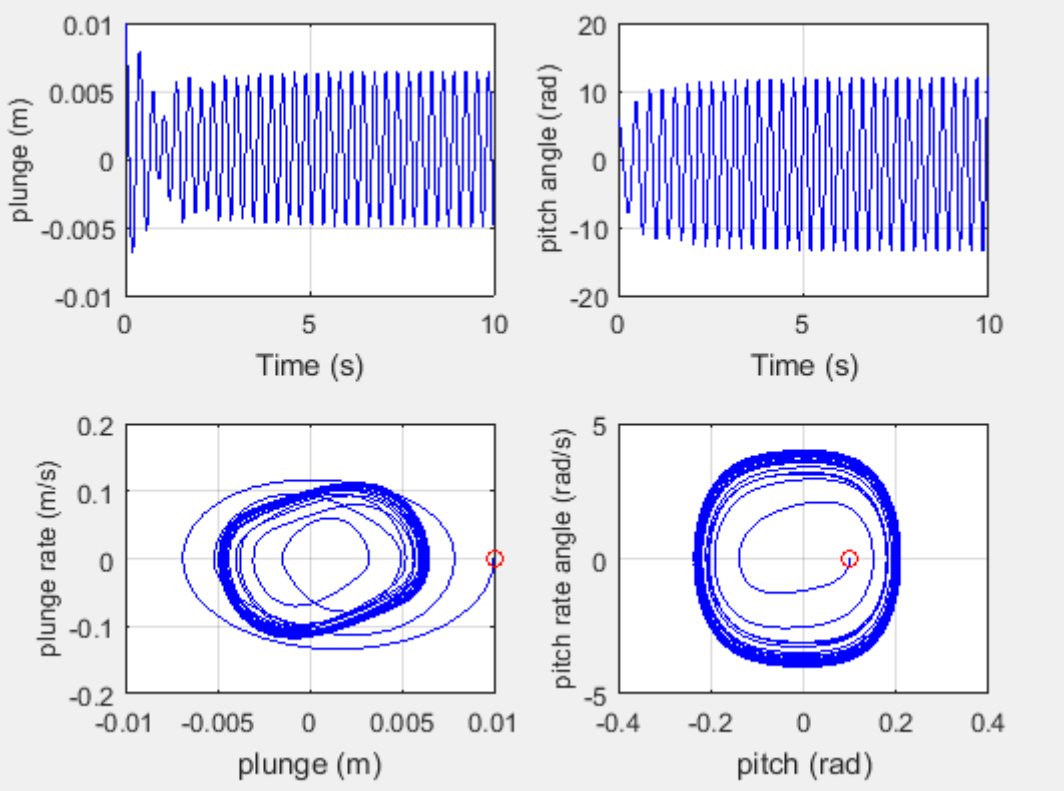

Figure IV.2: The open-loop system response for the vertical displacement h and the pitch angle α and its velocities, at a speed V = 13m/s.

For another flow velocity V = 19m/s, the calculated mathematical model is given as follows:

$$\begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -214.1696 & -11.3510 & -3.1688 & -0.2355 \\ 860.0497 & -222.7929 & 8.8321 & -0.1772 \end{bmatrix} \begin{bmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -11.0335 & 0.7902 \\ 3.7731 & -92363 \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \end{bmatrix}$$

In open loop, this process has the following eigenvalues λ, damping coefficients ε and frequencies ω:

Eigenvalues λ	Damping coefficients ε	Frequencies ω
2.4829 ± 15.8907i	-0.1550	±15.8907
-4.1558 ± 14.3153i	0.2769	±14.3153

Table IV.3: The characteristics of the open-loop system, for $V = 19\ m/s$.

The system response is shown in Figure (IV.3):

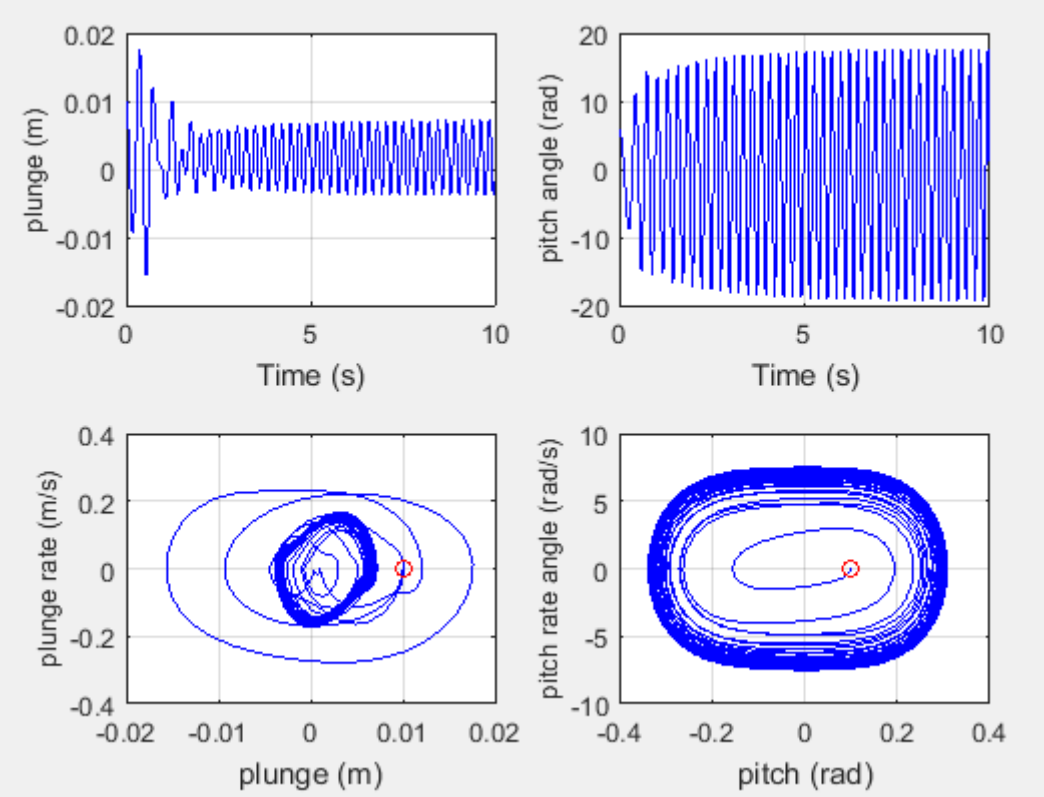

Figure IV.3: The open-loop system response for the vertical displacement h and the pitch angle α and its velocities, at a speed V = 19m/s.

IV.1.1. Interpretation:

The open loop simulation results show that at the speed V = 10m/s (Figure (IV.1)), the system has low amplitude oscillations which slowly converge to zero, they take about 6 seconds for they are amortized. The system has low positive damping coefficients (ε_1=0.0494, ε_2=0.0656), so it is stable but poorly damped.

At the speed V = 13 m/s, the model becomes unstable as shown in the figure (IV.2), the eigenvalues of the matrix are not all with negative real part, they also show that the system has a negative damping coefficient (ε_1=-0.0928) which illustrates the phenomenon of floating.

So, there is a transition velocity called Flutter Velocity, which leads the system from stability to instability. With successive simulations, this speed is identified at:

$$\mathbf{V_{flutter}=10.77\ m/s}$$

IV.2. Closed loop simulation:

In view of the previous results and after analyzing the open loop of the system, it was found that the performance of the system is not satisfactory and requires improvement.

It is planned to determine robust controllers based on sliding mode control, to guarantee stability, improve performance and increase the floating speed of the system.

IV.2.1. Application of sliding mode control:

Sliding mode control (SMC) is a control technique that aims to ensure the system states reach and remain on a specific sliding surface, regardless of model uncertainties or disturbances. The sliding surface defines a desired behavior and acts as a switching condition to control the system dynamics. The robustness of a sliding mode controller is related to its ability to maintain the sliding motion and desired performance in the presence of various uncertainties.

we developed a law of a sliding mode controller based on Lyapunov's equation for our model in the chapter II. The simulation of the system in the presence of the controller is carried out using MATLAB, such that we will simulate for the same values of the speeds used in the previous part and we will choose λ_1=2, λ_2=4, tho_1=3 and tho_2=1, and we will keep the same initial vector x, and we will plot the plunge h, the pitch angle α and its velocities with the command β and γ:

For a flow velocity $V = 10\ m/s$, the system response is shown in Figure (IV.4):

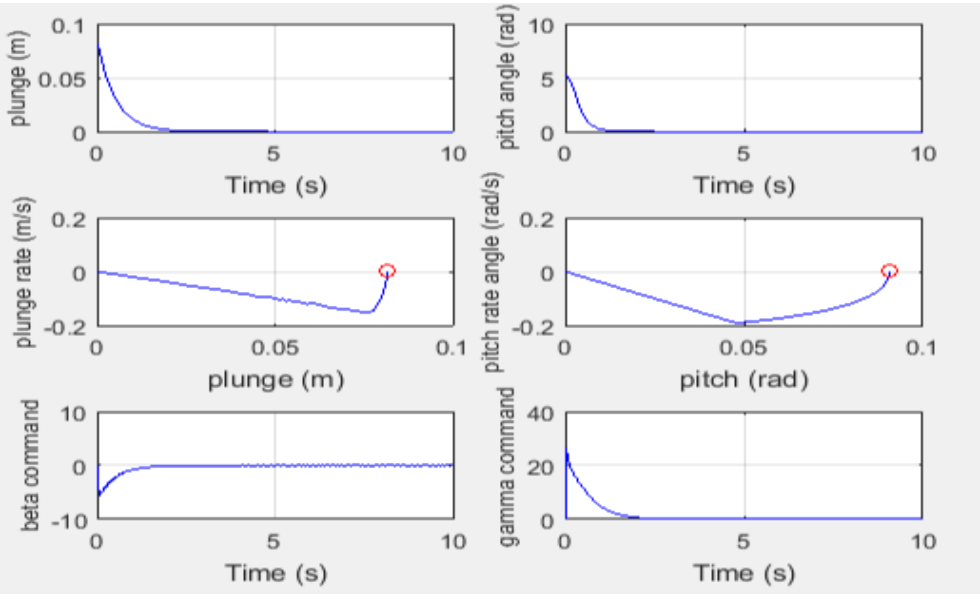

Figure IV.4: The response of the model after applying a SMC with the movement of command surfaces, at a speed V = 10m/s.

For another flow velocity $V = 13\ m/s$, the system response is shown in Figure (IV.5):

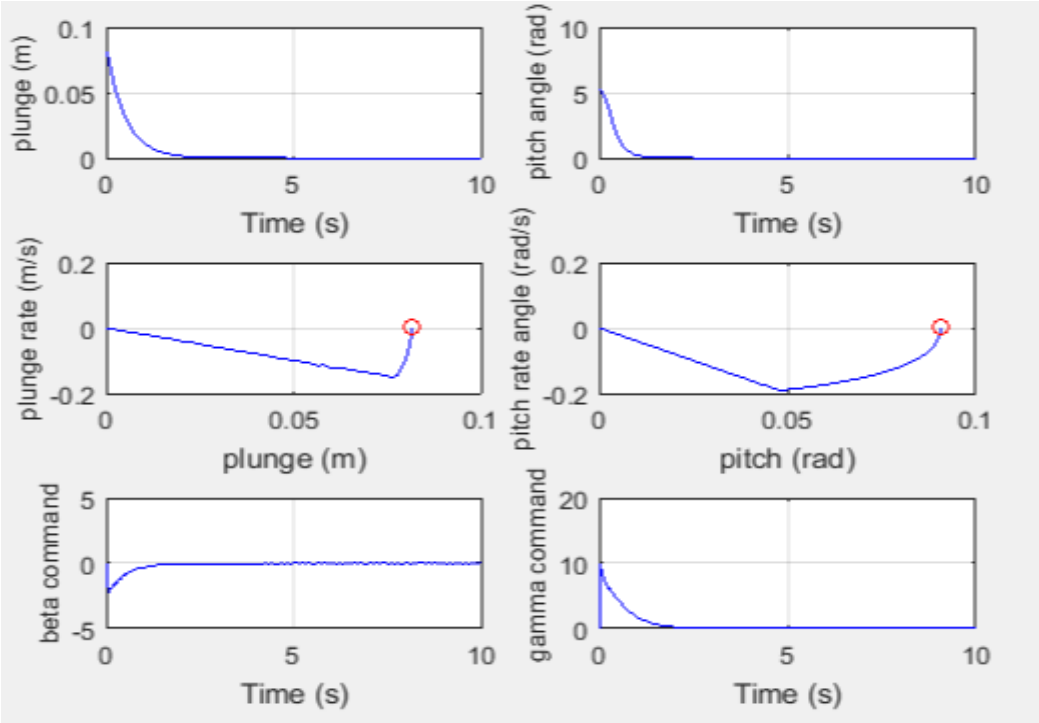

Figure IV.5: The response of the model after applying a SMC with the movement of command surfaces, at a speed V = 13m/s.

For the next flow velocity $V = 19\ m/s$, the system response is shown in Figure (IV.6):

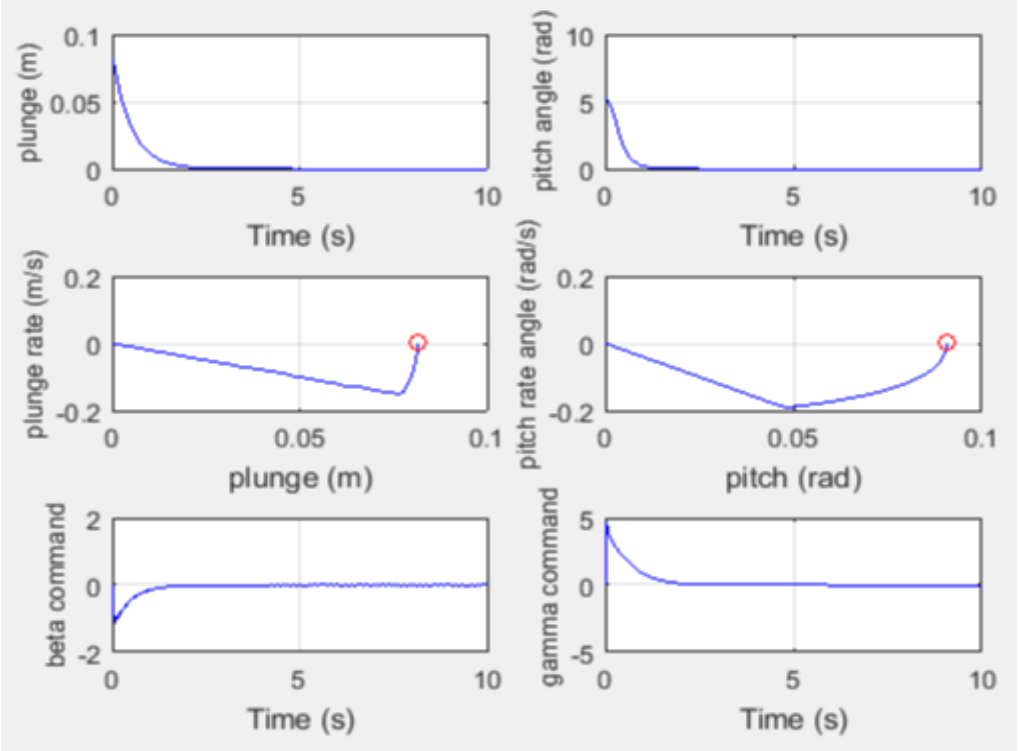

Figure IV.6: The response of the model after applying a SMC with the movement of command surfaces, at a speed V = 19m/s.

IV.2.1. Interpretation:

For V=10m/s:

The open loop system is stable but badly damped, the pitching movement and the vertical displacement of the profile converge slowly with weak oscillations towards the value zero (figure (IV.1)).

By applying the sliding mode command, we obtain a faster convergence of 2 seconds, we also see an improvement in the damping coefficient $\boldsymbol{\varepsilon}$ which makes the system well damped.

For this speed, the deflections of the mobile surfaces are acceptable as shown in the figure (IV.4), and it took a bigger value of deflections than the other two velocities 13 m/s and 18 m/s.

For V=13m/s and V=19m/s:

The open loop system is unstable (figure (IV.2), figure (IV.3)), the movement of pitch and vertical displacement of the profile present significant and divergent oscillations, hence the need to use a controller to stabilize the system.

By applying the sliding mode control law, we obtain a stable system, a little faster than the open loop one with less significant oscillations and an improvement in the damping coefficient $\boldsymbol{\varepsilon}$. We can see here that the β and γ commands decrease the more we increase the speed of flow.

The sliding mode command succeeded in controlling the pitch movement α and the vertical displacement h, it reduced the response time significantly but it did reduce the level of vibrations, and improved the damping coefficient.

IV.3. Adaptive sliding mode controller:

An adaptive sliding mode controller is a control strategy that combines the concepts of sliding mode control and adaptive control. It aims to achieve robust control of uncertain systems by adapting the controller parameters based on online estimation of system dynamics and uncertainties.

To get the online estimation of system dynamics and uncertainties, we train the RBF neural network on multiple inputs to use it later to adaptive our sliding mode controller.

IV.3.1. Training an RBF neural network:

We chose to use an RBF neural network with 3 layers and 10 neurons in every layer and we chose the input vector (x0=0.1*[1;1;0;0].*rand(4,1)) that change in every iteration (1000 iterations) to get the maximum states of our model and to get a powerful neural network that will be able to handle the maximum states, we notice that we chose only two velocities among the last three ones which are 10 m/s and 13 m/s to do the training.

After 25 hours of training for each velocity, we got our trained RBF neural networks that will be used later to build our adaptive sliding mode controller, and these were the results when we did a comparison between the trained neural network (y_{nn}) and our system (y):

For V=10m/s:

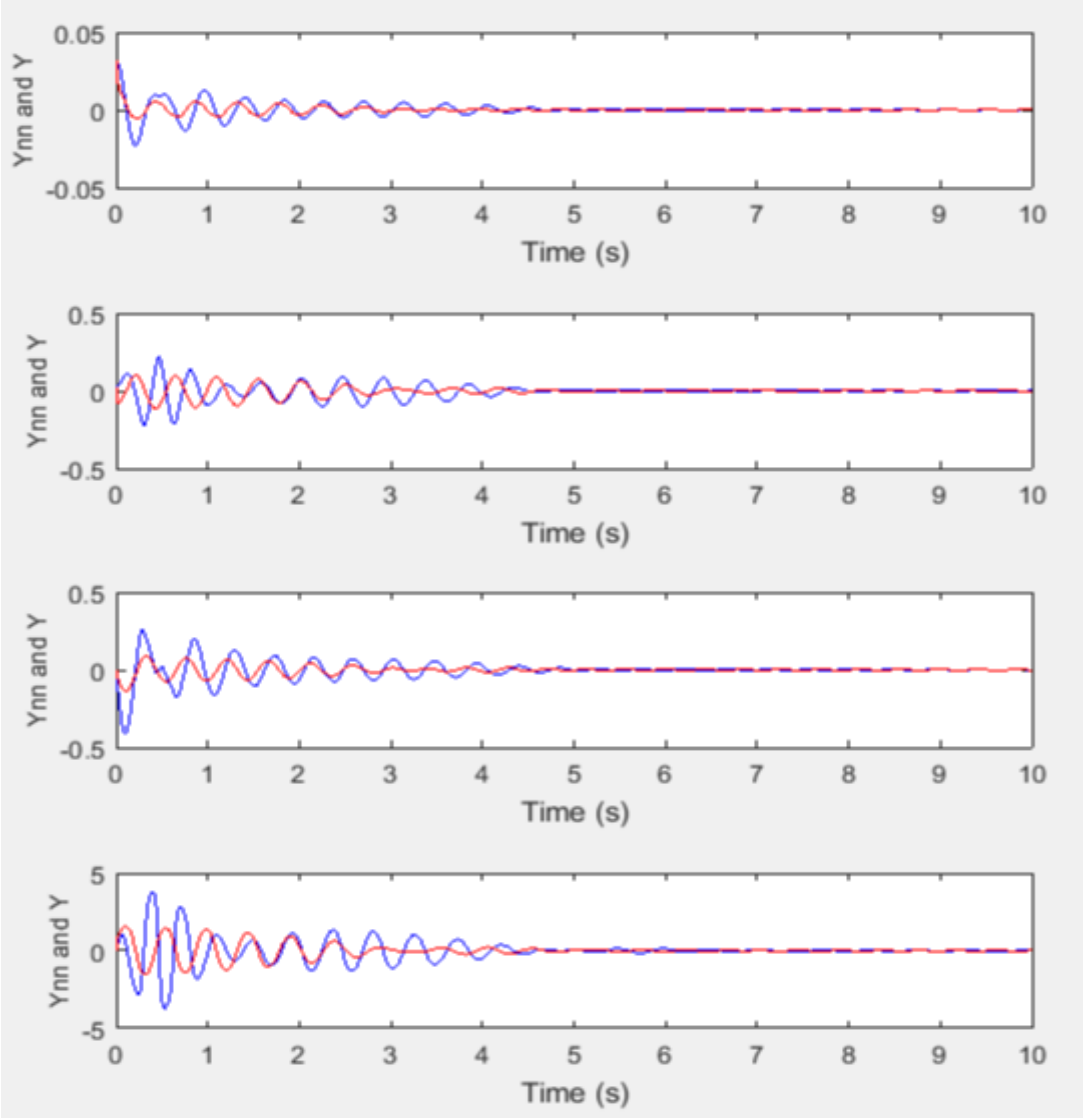

Figure IV.7: a comparison between the trained neural network (blue) and the model (red) at a speed V = 10m/s.

For V=13m/s:

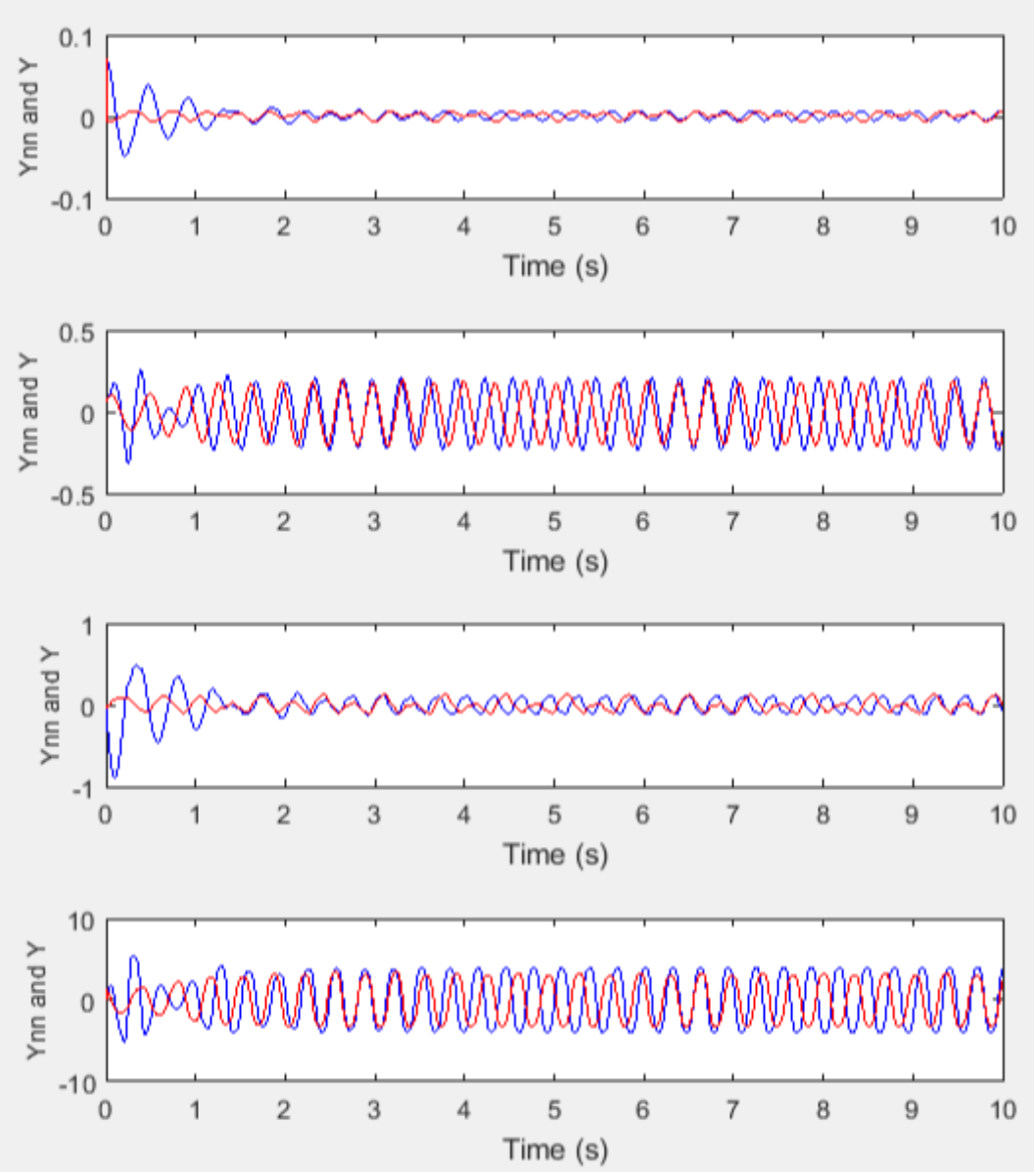

Figure IV.8: a comparison between the trained neural network (blue) and the model (red) at a speed V = 13m/s.

After that, we tried to test our trained neural networks on different velocities to see if they will be able to deal with a different state in different velocity than the one they trained for, and the result was that the first neural network could manage the velocities less than 10 m/s, and the second neural network could handle the differences in velocity between 11 m/s and 19 m/s, we assume that the explanation is because the flutter velocity was 10.77 m/s which means that the first neural network that got a training on our model in the convergence state couldn't deal with the velocities above our flutter velocity, and the opposite for the second neural network.

IV.3.2. Creating adaptive sliding mode controller:

In this part, we used a real-time RBFNN for the velocity 19 m/s to adapt our SMC and then we apply it on our divergent model to see its ability to control it, and the figure below shows the results of the simulation:

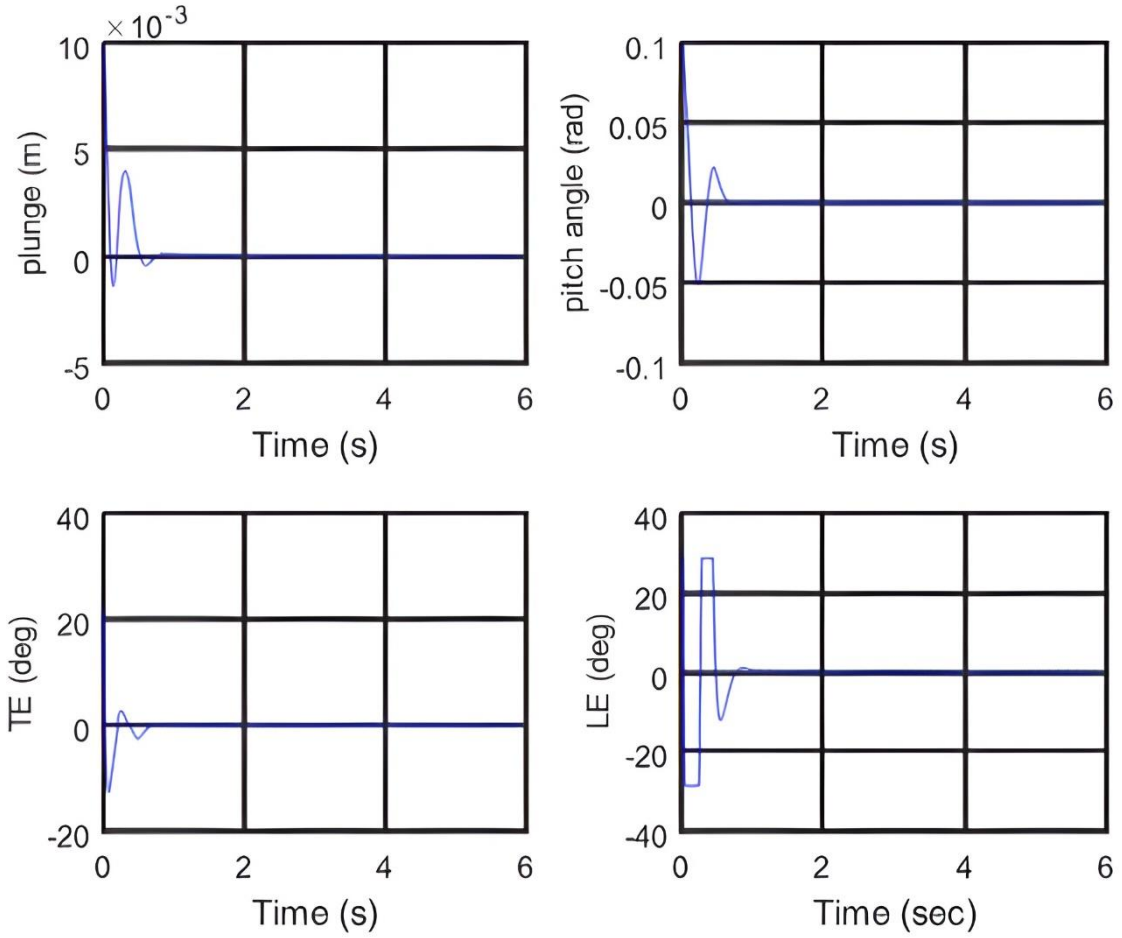

Figure IV.9: The response of the model after applying ARBFNNSM with the movement of command surfaces (TE is β command and LE is γ command), at a speed V = 19m/s.

Notice that the values of λ_1, λ_2, tho_1 and tho_2 remain the same as the previous simulation.

The plunge displacement and pitch angle responses at speed using the proposed ARBFNNSM controller are shown in Fig.IV.9. The associated leading and trailing-edge control surfaces deflection are shown in Fig.IV.9. It is clear to see that the designed control law efficiently derives the plunge displacement and pitch angle to zero within approximately 1s. Also, the control inputs must be constrained to vary between during the initial transient period to remain within the actuator's capabilities.

General conclusion:

In summary, the work in this thesis revolves primarily around the mathematical modeling of a two-dimensional wing section with two degrees of freedom and two control surfaces, the dynamics of which concern the pitch and vertical displacement, which for specific flow velocities present poorly damped or unstable oscillations, and then an application of the reputed "robust" controls was considered to stabilize the system, suppress the unstability. We began by providing a general overview of the phenomenon of aeroelasticity, more specifically flutter, and the best solution to remedy it, "aeroservoelasticity."

The mathematical development of the dynamic model based on the application of the Lagrange method then became an important aspect of our study about the physical system, allowing us to obtain the mathematical model of the system. The TAMU WING II wing model was then used to create an application to retrieve its state representation.

These modelization allowed us to simulate the dynamic behavior of the wing section in terms of pitch movement and vertical displacement, which led us to detect the flutter phenomenon and to predict different corresponding speeds.

To deal with the floating phenomenon, we opted for two types of control, it is the control by sliding mode control and use the RBF neural network to create an adaptive controller.

Through experimental studies and numerical simulations, the research has provided valuable insights into the dynamics of flutter and the design considerations for sliding mode controllers. The development of appropriate sliding surfaces and control laws has been a key focus, ensuring that the system operates within safe boundaries while actively suppressing the flutter oscillations.

The results of the research have shown that sliding mode controllers can effectively suppress the flutter phenomenon and improve the stability and performance of flexible wings. The robustness and adaptability of sliding mode control make it suitable for handling uncertainties in aerodynamic forces, structural parameters, and varying flight conditions, which are crucial factors in flutter control. Furthermore, the research has explored the integration of sliding mode control with other control techniques, RBF neural network to result an adaptive control, to enhance the flutter suppression capabilities. This interdisciplinary approach has the potential to provide even greater effectiveness and robustness in mitigating flutter and ensuring the safety and performance of flexible wings.

References:

[1] T.H.G,MEGSON, «Aircraft structure for engineering students »,ELSEVIER aerospace engineering series,2007.

[2] EARL,H.,CLARK,R.,DAVID,C., «A modern course in aeroelasticity »,Kluwer academic publishers,2004.

[3] Y.,C.FUNG, «An introduction to the theory of aeroelasticity »,DOVER publishers,1983.

[4] A.TEWARI, «Aeroservoelasticity: modeling and control »,Bir Khauser.

[5] P.,WILFRID, «sliding mode control in engineering »,Marcel Dekker,2002.

[6] A.R.,COLLAR, «The first fifty years of aeroelasticity »,1977.

[7] R.,L.,BISPLIGHOFF, «Aeroelasticity » ,1955.

[8] R.,G.,LOEWEY, «Resent developments in smart structures with aeronautical applications »,1977.

[9] J.,T.,XING, «Fluid-solid interaction dynamics »,Academic press,2019.

[10] A.,V.,BALAKRISHNAN « Aeroelasticity: the continuum theory », 2012.

[11] G.,DIMITRIODIS «Introduction to nonlinear aeroelasticity », 2017.

[12] EARL,H.,CLARK,R.,DAVID,C., «Amodern course in aeroelasticity »,Kluwer academic publishers,2004.

[13] H.,ASHLEY, « Aeroelasticity »,Dover publications.INC,1983.

[14] A.TEWARI, «Adaptive aerosevoelasticity control »,Aerospace series,2016.

[15] R.,P.,PELOUBET, «Recent development in the F-16 flutter suppression with active controls program »,Journal of aircrafts,1984.

[16] E.ALBANO, «A doublet-latice method for calculating lift distributions on oscillating surfaces in subsonic flows »,1969.

[17] J.GVID, «Active control of wind-tunnel model aeroelastic response using neural networks »,2000.

[18] M.KRISTC, «Nonlinear and adaptive control design»,1995.

[19] J.,E.,GIBSON, «Nonlinear automatic control »,1963.

[20] F.LOTZ, «Discontinuous automatic control»,1953.

[21] C.EDWARDS, «Sliding mode control: theory and applications »,1998.

[22] Y.ITKIS, «Control systems of variable structure »,1976.

[23] R.,A.,DECARLO, «Variable structure control of nonlinear multivariable systems »,1988.

[24] Y.,SHTESSEL, «Sliding mode control and observation »,Birkauser,2014

[28] SUKUT,T.W., « Nonlinear aeroelastic analysis of UAVs : Deterministic and stochastic Approaches », 2d Lt USAF, RICE University HOUSTON, TEXAS, April 2012.

[29] TENG,Y., « Modeling and simulation of Aeroservoelastic control with multiple control surfaces using µ-method ». these Phd ,Claremont, California and Long beach, California,2005.

[30] DEWEY, HODGES,H., ALVIN PIERCE,G., « Introduction to structural dynamics and aeroelasticity », campridge aerospace series 2001.

[31] ROTELLA, F., « Commande des systèmes linéaires à plusieurs entrées », Ecole Nationale d'Ingénieurs de Tarbes ,2003.

[33] HAYKIN,S.H., « Neural networks and learning machines», 2009.

[34] HOGAN,M.T., « Neural network design », 2014.

[35] JINKIN,L., « Radial basis function NN control for mechanical systems: design, analysis and matlab silmulation », 2013.

[36] Edwards,C., « Sliding mode control: theory and application », CRS press .Inc, 1998.

[37] YANG,L., « Adaptive sliding mode NN control for nonlinear systems », Elsevior .INC, 2019.

Appendix

RBF neural network training methods

In this appendix we will discuss in detail the most used technics in RBFNN training and Lyapunov function that mentioned earlier in chapter III.

1. Linear least squares:

In this section we will assume that the first layer weights and biases of the RBF network are fixed. This can be done by fixing the centers on a grid, or by randomly selecting the centers from the input vectors in the training data set (or by using the clustering method which is described in a later section). When the centers are randomly selected, all of the biases can be computed using the following formula:

$$b_i^1 = \frac{\sqrt{S^1}}{d_{max}} \tag{D.1}$$

where d_{max} is the maximum distance between neighboring centers. This is designed to ensure an appropriate degree of overlap between the basis functions. Using this method, all of the biases have the same value. There are other methods which use different values for each bias. We will discuss one such method later, in the section on clustering.

Once the first layer parameters have been set, the training of the second layer weights and biases is equivalent to training a linear network. For example, consider that we have the following training points:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \tag{D.2}$$

where p_q is an input to the network, and is the corresponding target output. The output of the first layer for each input p_q in the training set can be computed as:

$$n_{i,q}^1 = \left\| \mathbf{p}_q - {}_i\mathbf{w}^1 \right\| b_i^1 \tag{D.3}$$

$$\mathbf{a}_q^1 = \mathbf{radbas}(\mathbf{n}_q^1) \tag{D.4}$$

Since the first layer weights and biases will not be adjusted, the training data set for the second layer then becomes:

$$\{\mathbf{a}_1^1, \mathbf{t}_1\}, \{\mathbf{a}_2^1, \mathbf{t}_2\}, \ldots, \{\mathbf{a}_Q^1, \mathbf{t}_Q\} \tag{D.5}$$

The second layer response is linear:

$$\mathbf{a}^2 = \mathbf{W}^2\mathbf{a}^1 + \mathbf{b}^2 \tag{D.6}$$

We want to select the weights and biases in this layer to minimize the sum square error performance index over the training set:

$$F(\mathbf{x}) = \sum_{q=1}^{Q} (\mathbf{t}_q - \mathbf{a}_q^2)^T (\mathbf{t}_q - \mathbf{a}_q^2) \tag{D.7}$$

Our derivation of the solution to this linear least square problem will follow the linear network derivation. To simplify the discussion, we will assume a scalar target, and we will lump all of the parameters we are adjusting, including the bias, into one vector:

$$\mathbf{x} = \begin{bmatrix} {}_1\mathbf{w}^2 \\ b^2 \end{bmatrix} \tag{D.8}$$

Similarly, we include the bias input "1" as a component of the input vector:

$$\mathbf{z}_q = \begin{bmatrix} \mathbf{a}_q^1 \\ 1 \end{bmatrix} \tag{D.9}$$

Now the network output, which we usually write in the form:

$$a_q^2 = ({}_1\mathbf{w}^2)^T \mathbf{a}_q^1 + b^2 \tag{D.10}$$

can be written as:

$$a_q = \mathbf{x}^T \mathbf{z}_q \tag{D.11}$$

This allows us to conveniently write out an expression for the sum square error:

$$F(\mathbf{x}) = \sum_{q=1}^{Q} (e_q)^2 = \sum_{q=1}^{Q} (t_q - a_q)^2 = \sum_{q=1}^{Q} (t_q - \mathbf{x}^T \mathbf{z}_q)^2 \tag{D.12}$$

To express this in matrix form, we define the following matrices:

$$\mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_Q \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} {}_1\mathbf{u}^T \\ {}_2\mathbf{u}^T \\ \vdots \\ {}_Q\mathbf{u}^T \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_Q^T \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_Q \end{bmatrix} \tag{D.13}$$

The error can now be written:

$$\mathbf{e} = \mathbf{t} - \mathbf{U}\mathbf{x} \tag{D.14}$$

and the performance index become:

$$F(\mathbf{x}) = (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x}) \tag{D.15}$$

If we use regularization to help in preventing overfitting, we obtain the following form for the performance index:

$$F(\mathbf{x}) = (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x}) + \rho \sum_{i=1}^{n} x_i^2 = (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x}) + \rho \mathbf{x}^T\mathbf{x} \tag{D.14}$$

Let's expand this expression to obtain:

$$\begin{aligned} F(\mathbf{x}) &= (\mathbf{t} - \mathbf{U}\mathbf{x})^T(\mathbf{t} - \mathbf{U}\mathbf{x}) + \rho \mathbf{x}^T\mathbf{x} = \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\mathbf{U}\mathbf{x} + \mathbf{x}^T\mathbf{U}^T\mathbf{U}\mathbf{x} + \rho \mathbf{x}^T\mathbf{x} \\ &= \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\mathbf{U}\mathbf{x} + \mathbf{x}^T[\mathbf{U}^T\mathbf{U} + \rho \mathbf{I}]\mathbf{x} \end{aligned} \tag{D.15}$$

Take a close look at Eq. (D.15), and compare it with the general form of the quadratic function, given here:

$$F(\mathbf{x}) = c + \mathbf{d}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} \tag{D.16}$$

Our performance function is a quadratic function, where:

$$c = \mathbf{t}^T\mathbf{t}, \ \mathbf{d} = -2\mathbf{U}^T\mathbf{t} \text{ and } \mathbf{A} = 2[\mathbf{U}^T\mathbf{U} + \rho \mathbf{I}] \tag{D.17}$$

We know that the characteristics of the quadratic function depend primarily on the Hessian matrix. For example, if the eigenvalues of the Hessian are all positive, then the function will have one unique global minimum. In this case the Hessian matrix is:

$$2[\mathbf{U}^T\mathbf{U} + \rho \mathbf{I}]$$

and it can be shown that this matrix is either positive definite or positive semidefinite, which means that it can never have negative eigenvalues. We are left with two possibilities. If the Hessian matrix has only positive eigenvalues, the performance index will have one unique global minimum. If the Hessian matrix has some zero eigenvalues, the performance index will either have a weak minimum, or no minimum, depending on the vector. In this case, it must have a minimum, since is a sum square function, which cannot be negative.

2. Clustering:

There is another approach for selecting the weights and biases in the first layer of the RBF network. This method uses the competitive networks. After training, the rows of the competitive networks contain prototypes, or cluster centers. This provides an approach for locating centers and selecting biases for the first layer of the RBF network. If we take the input vectors from the training set and perform a clustering operation on them, the resulting prototypes (cluster centers) could be used as centers for the RBF network. In addition, we could compute the variance of each individual cluster and use that number to calculate an

appropriate bias to be used for the corresponding neuron. Consider again the following training set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\} \tag{D.18}$$

We want to perform a clustering of the input vectors from this training set:

$$\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_Q\} \tag{D.19}$$

We will train the first layer weights of the RBF network to perform a clustering of these vectors, using the Kohonen learning rule:

$${}_{i^*}\mathbf{w}^1(q) = {}_{i^*}\mathbf{w}^1(q-1) + \alpha(\mathbf{p}(q) - {}_{i^*}\mathbf{w}^1(q-1)) \tag{D.20}$$

where p(q) is one of the input vectors in the training set, and ${}_{i*}w^1(q\text{-}1)$ is the weight vector that was closest to p(q). We could also use other clustering algorithms, such as the Self Organizing Feature Map, or the k-means clustering algorithm. The resulting converged weights will represent cluster centers of the training set input vectors. This will insure that we will have basis functions located in areas where input vectors are most likely to occur. In addition to selecting the first layer weights, the clustering process can provide us with a method for determining the first layer biases. For each neuron (basis function), locate the n_c input vectors from the training set that are closest to the corresponding weight vector (center). Then compute the average distance between the center and its neighbors.

$$dist_i = \frac{1}{n_c}\left(\sum_{j=1}^{n_c} \left\|\mathbf{p}_j^i - {}_i\mathbf{w}^1\right\|^2\right)^{\frac{1}{2}} \tag{D.21}$$

Where p_1^i is the input vector that closest to ${}_iw^1$, and is p_2^i the next closest input vector. From these distances, recommends setting the first layer biases as follows:

$$b_i^1 = \frac{1}{\sqrt{2}dist_i} \tag{D.22}$$

Therefore, when a cluster is wide, the corresponding basis function will be wide as well. Notice that in this case each bias in the first layer will be different. This should provide a network that is more efficient in its use of basis functions than a network with equal biases. After the weights and biases of the first layer are determined, linear least squares is used to find the second layer weights and biases. There is a potential drawback to the clustering method for designing the first layer of the RBF network. The method only takes into account the distribution of the input vectors; it does not consider the targets. It is possible that the function we are trying to approximate is more complex in regions for which there are fewer

inputs. For this case, the clustering method will not distribute the centers appropriately. On the other hand, one would hope that the training data is located in regions where the network will be most used, and therefore the function approximation will be most accurate in those areas.

3.Orthogonal least squares:

In the previous section we assumed that the weights and biases in the first layer were fixed. (For example, the centers could be fixed on a grid, or randomly selected from the input vectors in the training set.) In this section we consider a different approach for selecting the centers. We will assume that there exists a number of potential centers. These centers could include the entire set of input vectors in the training set, vectors chosen in a grid pattern, or vectors chosen by any other procedure one might think of. We will then select vectors one at a time from this set of potential centers, until the network performance is satisfactory. We will build up the network one neuron at a time.

The basic idea behind this method comes from statistics, and it is called subset selection. The general objective of subset selection is to choose an appropriate subset of independent variables to provide the most efficient prediction of a target dependent variable. For example, suppose that we have 10 independent variables, and we want to use them to predict our target dependent variable. We want to create the simplest predictor possible, so we want to use the minimum number of independent variables for the prediction. Which subset of the 10 independent variables should we use? The optimal approach, called an exhaustive search, tries all combinations of subsets and finds the smallest one that provides satisfactory performance. Unfortunately, this strategy is not practical. If we have Q variables in our original set, the following expression gives the number of distinct subsets:

$$\sum_{q=1}^{Q} \frac{Q!}{q!(Q-q)!}$$

If Q=10, this number is 1023. If Q=20, the number is more than 1 million. We need to have a less expensive strategy than the exhaustive search. There are several suboptimal procedures. They are not guaranteed to find the optimal subset, but they require significantly less computation. One procedure is called forward selection. This method begins with an empty model and then adds variables one at a time. At each stage, we add the independent variable that provides the largest reduction in squared error. We stop adding variables when the performance is adequate. Another approach, called backward elimination, starts with all independent variables selected for the model. At each stage we eliminate the variable that

would cause the least increase in the squared error. The process continues until the performance is inadequate. There are other approaches which combine forward selection and backward elimination, so that variables can be added and deleted at each iteration. Any of the standard subset selection techniques can be used for selecting RBF centers. For purposes of illustration, we will consider one specific form of forward selection, called orthogonal least squares. Its main feature is that it efficiently calculates the error reduction provided by the addition of each potential center to the RBF network. We have the Eq. (D.14), repeated here in slightly different form:

$$\mathbf{t} = \mathbf{U}\mathbf{x} + \mathbf{e}. \tag{D.23}$$

We will use our standard notation for matrix rows and columns to individually identify both the rows and the columns of the matrix U:

$$\mathbf{U} = \begin{bmatrix} {}_1\mathbf{u}^T \\ {}_2\mathbf{u}^T \\ \vdots \\ {}_Q\mathbf{u}^T \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_Q^T \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix} \tag{D.24}$$

Here each row of the matrix U represents the output of layer 1 of the RBF network for one input vector from the training set. There will be a column of the matrix U for each neuron (basis function) in layer 1 plus the bias term ($n=S^1+1$). Note that for the OLS algorithm, the potential centers for the basis functions are often chosen to be all of the input vectors in the training set. In this case, n will equal Q+1, since the constant "1" for the bias term is included in z. The matrix U is called the regression matrix, and the columns of U are called the regressor vectors.

The objective of OLS is to determine how many columns of (numbers of neurons or basis functions) should be used. The first step is to calculate how much each potential column would reduce the squared error. The problem is that the columns are generally correlated with each other, and so it is difficult to determine how much each individual column would reduce the error. For this reason, we need to first orthogonalize the columns. Orthogonalizing the columns means that we can decompose U as follows:

$$\mathbf{U} = \mathbf{M}\mathbf{R} \tag{D.25}$$

where is an upper triangular matrix, with ones on the diagonal:

$$\mathbf{R} = \begin{bmatrix} 1 & r_{1,2} & r_{1,3} & \cdots & r_{1,n} \\ 0 & 1 & r_{2,3} & \cdots & r_{2,n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{n-1,n} \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{D.26}$$

And M is a matrix with orthogonal columns m_i. This means that M has the following properties:

$$\mathbf{M}^T\mathbf{M} = \mathbf{V} = \begin{bmatrix} v_{1,1} & 0 & \cdots & 0 \\ 0 & v_{2,2} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & v_{n,n} \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1^T\mathbf{m}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{m}_2^T\mathbf{m}_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \mathbf{m}_n^T\mathbf{m}_n \end{bmatrix} \tag{D.24}$$

Now Eq. (D.23) can be written:

$$\mathbf{t} = \mathbf{MRx} + \mathbf{e} = \mathbf{Mh} + \mathbf{e} \tag{D.25}$$

Where:

$$\mathbf{h} = \mathbf{Rx} \tag{D.26}$$

The least squares solution for Eq. (D.25) is :

$$\mathbf{h}^* = [\mathbf{M}^T\mathbf{M}]^{-1}\mathbf{M}^T\mathbf{t} = [\mathbf{V}]^{-1}\mathbf{M}^T\mathbf{t} \tag{D.27}$$

and because V is diagonal, the elements of can be computed:

$$h_i^* = \frac{\mathbf{m}_i^T\mathbf{t}}{v_{i,i}} = \frac{\mathbf{m}_i^T\mathbf{t}}{\mathbf{m}_i^T\mathbf{m}_i}. \tag{D.28}$$

From h* we can compute x* using Eq. (D.26). Since R is upper-triangular, Eq. (D.26) can be solved by back-substitution and does not require a matrix inversion. There are a number of ways to obtain the orthogonal vectors m_i, but we will use the Gram-Schmidt orthogonalization process, starting with the original columns of U:

$$\mathbf{m}_1 = \mathbf{u}_1 \tag{D.29}$$

$$\mathbf{m}_k = \mathbf{u}_k - \sum_{i=1}^{k-1} r_{i,k}\mathbf{m}_i \tag{D.30}$$

Where:

$$r_{i,k} = \frac{\mathbf{m}_i^T\mathbf{u}_k}{\mathbf{m}_i^T\mathbf{m}_i},\ i = 1, \ldots, k-1 \tag{D.31}$$

Now let's see how orthogonalizing the columns of U enables us to efficiently calculate the squared error contribution of each basis vector. Using Eq. (D.25), the total sum square value of the targets is given by:

$$\mathbf{t}^T\mathbf{t} = [\mathbf{Mh}+\mathbf{e}]^T[\mathbf{Mh}+\mathbf{e}] = \mathbf{h}^T\mathbf{M}^T\mathbf{Mh}+\mathbf{e}^T\mathbf{Mh}+\mathbf{h}^T\mathbf{M}^T\mathbf{e}+\mathbf{e}^T\mathbf{e} \tag{D.32}$$

Consider the second term in the sum:

$$\mathbf{e}^T\mathbf{Mh} = [\mathbf{t}-\mathbf{Mh}]^T\mathbf{Mh} = \mathbf{t}^T\mathbf{Mh}-\mathbf{h}^T\mathbf{M}^T\mathbf{Mh} \tag{D.33}$$

If we use the optimal from Eq. (D.27), we find:

$$\mathbf{e}^T\mathbf{Mh}^* = \mathbf{t}^T\mathbf{Mh}^*-\mathbf{t}^T\mathbf{MV}^{-1}\mathbf{M}^T\mathbf{Mh}^* = \mathbf{t}^T\mathbf{Mh}^*-\mathbf{t}^T\mathbf{Mh}^* = 0 \tag{D.34}$$

Therefore, the total sum square value from Eq. (D.32) becomes:

$$\mathbf{t}^T\mathbf{t} = \mathbf{h}^T\mathbf{M}^T\mathbf{Mh}+\mathbf{e}^T\mathbf{e} = \mathbf{h}^T\mathbf{Vh}+\mathbf{e}^T\mathbf{e} = \sum_{i=1}^{n} h_i^2\mathbf{m}_i^T\mathbf{m}_i+\mathbf{e}^T\mathbf{e} \tag{D.35}$$

The first term on the right of Eq. (D.35) is the contribution to the sum squared value explained by the regressors, and the second term is the remaining sum squared value that is not explained by the regressors. Therefore, regressor (basis function) i contributes

$$h_i^2\mathbf{m}_i^T\mathbf{m}_i \tag{D.36}$$

to the squared value. This also represents how much the squared error can be reduced by including the corresponding basis function in the network. We will use this number, after normalizing by the total squared value, to determine the next basis function to include at each iteration:

$$o_i = \frac{h_i^2\mathbf{m}_i^T\mathbf{m}_i}{\mathbf{t}^T\mathbf{t}} \tag{D.37}$$

This number always falls between zero and one.

4. Lyapunov function :

4.1. Definition:

A Lyapunov function, named after Russian mathematician Aleksandr Mikhailovich Lyapunov, is a mathematical function used to analyze the stability of dynamical systems. It provides a means to assess whether a system will converge to a stable equilibrium point or remain in a bounded region.

The core idea of Lyapunov second method is to set up a function similar to "energy" function, transform the problem of whether the system is stable into analyze the positive

definite problem of the “energy” function and its first derivative, such “energy” function is Lyapunov function.

Lyapunov functions are widely used in control theory and stability analysis to assess the stability of control systems, including linear and nonlinear systems. They provide a mathematical tool for proving stability properties and designing control strategies that ensure system stability.

4.2. Construction of Lyapunov function:

The construction of a Lyapunov function involves selecting a suitable scalar function that satisfies the conditions The construction of Lyapunov function is a complex problem. For a simple system, a class of Lyapunov function plays an important role in the stability analysis, that is, the quadratic form function.

$$V(x) = x^T P x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{12} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Note that x is real vector and P is real symmetric matrix.

$$V(x) = x^T P x = p_{11}x_1^2 + 2p_{12}x_1x_2 + \cdots 2p_{1n}x_1x_n + p_{22}x_2^2 + 2p_{23}x_2x_3 + \cdots 2p_{2n}x_2x_n + \cdots + p_{nn}x_n^2$$

4.3. Lyapunov Stability Theorem and Lyapunov Global Uniform Asymptotic Stability Theorem:

For continuous time, nonlinear, time-varying, autonomous system:

$$\dot{x} = f(x,t), \quad t \in [t_0, \infty).$$

if we can construct a scalar function V(x,t), V(0,t) = 0 that has a continuous first-order derivative to x and t, and all nonzero state x in the state space R^n are satisfied:

1. V(x,t) is positive definite function, which means there are two continuous nonscalar functions α(||x||) and β(||x||), where α(0) = 0 and β(0) = 0, for all t belong to $[t_0,\infty)$ and $x \neq 0$ satisfies:

$$0 < \alpha(\|x\|) \leq V(x,t) \leq \beta(\|x\|)$$

2. V(x,t) to t of the time derivative $\dot{v}$(x,t) negative definite, that is there is a continuous nondecreasing scaler function γ(||x||), where γ (0) = 0, for all t belong to [t0,∞) and x≠0 satisfies:

$$\dot{V}(x,t) \leq \gamma(\|x\|) < 0$$

3. When ||x||→∞, we have α(||x||) →∞, that is V(x,t) →∞. The equilibrium state x = 0 of the system is globally uniformly asymptotically stable.