

Control of Unmanned Aerial Vehicle with Wing Shape Identification using Vision System and Sensor Fusion

by

Julius Osemudiamen Adoghe M.Sc.

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

Carleton University

Ottawa, Ontario

© 2022, Julius Osemudiamen Adoghe

Abstract

This thesis presents a Deflection-Detection-Vision-System (DDVS) for unmanned aerial vehicles (UAV) fixed-wing for control and navigation. This technique allows measurement of the fixed-wing shape, deflection, and identification of the aerodynamic coefficient acting on the system, using information from the stereo camera and strain gauge. It determines specific points to identify the wing's shape and deflection. The model UAV is equipped with a stereo camera fixed at the top rear end of the device and strain gauges placed at eight different points marked on the wing. Both sensors measure the deflection in chosen locations simultaneously. The DDVS performance and dynamic parameters are tested in a wind tunnel at speeds ranging from 10 km/h to 35 km/h, angles of attack (AOA), and roll angles ranging from 0 degrees to 30 degrees, respectively. An image acquisition, feature extraction, matching process, 3D reconstruction, and stereo camera calibration are presented in this thesis as a part of proposed identification procedure. This approach measures the wing deflection at each selected point and identifies the maximum deflection location based on various aerodynamic conditions such as wind speed, AOA, and roll angle. The drag and lift forces were obtained using the wing's surface area, and the experiment shows that less force is required for lifting as the AOA increases. The DDVS was implemented in a UAV and tested in the wind tunnel. Extensive experiments were conducted to determine the deflection of the wing in the function of flight parameters like angle of attack, roll angle, and flow velocity. The experimental results have shown that the integration of strain gauge and vision system sensors identify wing deflections accurately. Extensive simulation results were compared with the experimental results and demonstrated that the proposed method-based sensor fusion could be used even in the most demanding environment.

The proposed control system is designed to control the aircraft's attitude and velocity. A Fuzzy PID controller has been suggested, and the stability of the controller was verified numerically. The performance of the Fuzzy PID controller was compared with Linear Quadratic Regulator (LQR) controller. The control system included a mathematical model of the aircraft that was based on the Cessna 172 aircraft model. In simulation experiments, the system's stability and robustness were checked and verified.

Acknowledgments

I would like to thank Almighty God and all of the people who encouraged and supported me during the undertaking of this research. To begin, I would like to express my gratitude to my Ph.D. program advisor, Professor Jurek Z. Sasiadek, for his continued guidance and support. He provided advice, opportunities, and assistance that greatly enhanced my research experience.

Many thanks to my mother, my father, and my brothers and sisters for their patience and support. In particular, I thank my lovely wife and my children “Marvellous, Katriel, and Hallel” for their company and understanding during this thesis activity.

Last but not certainly least, I would also like to thank my friends and fellow students for the great time spent together and the experiences shared.

Table of Contents

Abstract.....	ii
Acknowledgments	iv
Table of Contents	v
List of Tables	ix
List of Figures.....	x
Nomenclature	xiv
Abbreviation Definitions	xvii
Chapter 1	1
1 Introduction	1
1.1 Motivation	6
1.2 Contributions and Publication.....	7
1.3 Thesis Outline	9
Chapter 2	11
2 Literature Review	11
2.1 Computer Vision for UAV.....	11
2.1.1 Sensors Integration.....	15
2.1.2 Classification of the Wing Shape.....	16

2.1.3	Control System.....	17
Chapter 3	19
3	Deflection Calculation	19
3.1	Machine Vision	19
3.1.1	Stereo Vision Camera	19
3.1.2	Feature Detection and Tracking.....	23
3.1.3	RANdom SAmples Consensus (RANSAC).....	23
3.1.4	Triangulation.....	25
3.1.5	Homography	26
3.1.6	Decomposition of Homography Matrix.....	28
3.1.7	Fast Homography Decomposition Method.....	28
3.1.8	Iterative Closest Point Algorithm (ICP)	29
3.1.9	Horn’s Absolute Orientation Method	30
3.2	Deflection Calculation Using Strain Gauge.....	30
3.3	Standard Deviation Calculations.....	33
Chapter 4	35
4	Control System Design	35
4.1	Introduction	35
4.2	Longitudinal Control.....	35
4.3	Inner Loop Pitch Attitude Tracker Design.....	36

4.4	Outer Loop Altitude Hold Controller.....	37
4.5	Airspeed Hold controller.....	38
4.6	ANFIS Controller.....	38
4.7	Controllability and Observability.....	40
4.8	PID Controller.....	41
4.9	Fuzzy Logic Controller (FLC).....	43
4.10	Controllers.....	46
4.10.1	LQR Controller.....	46
4.10.2	Fuzzy Logic Controller (FLC).....	47
Chapter 5.....		51
5 Experimental Setup and Results.....		51
5.1	Introduction.....	51
5.2	Camera Calibration.....	51
5.3	Feature Matching and Tracking.....	53
5.4	Experiments with Camera Measurement System.....	57
5.5	Experimental result.....	59
5.6	Proposed Control System Design.....	70
Chapter 6.....		77
6 Result Verification.....		77
6.1	Sensor Fusion.....	81

6.2	The Unscented Kalman Filter	82
6.3	Mathematical Modeling of The Fixed-Wing	85
6.4	The Aerodynamic Results	92
6.4.1	The input parameters.....	94
6.4.2	The out parameters.....	94
6.5	Future Work	105
Chapter 7	107
7 Conclusion	107
References	109
Appendix	123

List of Tables

Table 4.1: Rule of KP, KI, and KD.....	50
Table 6.1: Properties of the fixed-wing.	79
Table 6.2. Comparison of Deflection taken from camera/strain gauge and simulation.	79
Table 6.3:Standard deviation calculation for strain gauge measurement	80
Table 6.4 Minimum velocity required to lift the wing (Projected area).	101
Table 6.5 Minimum velocity required to lift the wing (Surface area).	101
Table 6.6. The final output result for the UAV control.	104

List of Figures

Figure 1.1: UAV axes and proposed initial concept design.....	3
Figure 1.2: CAD drawings for the experimental setup.	5
Figure 1.3 Render modelling of UAV experimental setup.	6
Figure 3.1: StereoLabs ZED camera [80].	20
Figure 3.2: Model of the stereo camera.	21
Figure 3.3: The geometry of a homography mapping [88].....	26
Figure 3.4: Deformation due to the external force.....	32
Figure 4.1: Longitudinal Guidance and Control System for pitch hold [80].....	36
Figure 4.2: Longitudinal Guidance and Control System for altitude hold [80].....	37
Figure 4.3: Longitudinal Guidance and Control System for speed hold [98].....	38
Figure 4.4: Adaptive neuro-fuzzy interference system structure [100].	39
Figure 4.5 Block Diagram of PID Controller [100].....	43
Figure 4.6 Block diagram of Fuzzy logic.	44
Figure 4.7: The Platform of Cessna 172 [98].	44
Figure 4.8:Block diagram of the fuzzy-PID controller with LQR.....	47
Figure 4.9:Block diagram of the LQR controller.....	47
Figure 4.10: The blocks diagram of the fuzzy-PID controller.....	48
Figure 4.11: The membership function of the input variables.....	49
Figure 4.12: The membership function of the output variables.....	49
Figure 5.1: Calibrated chessboard.....	52
Figure 5.2: Camera calibration toolbox for MATLAB with a model chessboard.	53
Figure 5.3: Camera extrinsic parameters.	53

Figure 5.4: Points Position (stain gauge and target points).	55
Figure 5.5: Front and rear view of the fixed-wing in the wind tunnel.....	56
Figure 5.6: Matching points.....	56
Figure 5.7: Process flow of live vision camera measurement system.	58
Figure 5.8: Deflection vs. node no., AOA = 10 deg, Roll angle = 10 deg.	60
Figure 5.9: Deflection vs. node no., AOA = 10 deg, Roll angle = 10 deg.	61
Figure 5.10: Deflection vs. node no. at AOA = 20 deg Roll angle = 0 deg.....	61
Figure 5.11: Deflection vs. node no. at AOA = 30 deg, Roll angle = 0 deg.....	62
Figure 5.12: Deflection vs. node no. at AOA = 30 deg, Roll angle = 10 deg.....	62
Figure 5.13: Deflection vs. node no. at AOA = 30 deg, Roll angle = 20 deg.....	63
Figure 5.14: Max. deflection vs. different speed at AOA = 0 deg.....	64
Figure 5.15: Max. deflection vs. different speed at AOA = 20 deg.....	64
Figure 5.16: Max. deflection vs. different speed at AOA = 30 deg.....	65
Figure 5.17: Max. deflection vs. AOA for the fused value.....	65
Figure 5.18: Tracking points.....	70
Figure 5.19: The Proposal Detection-Deflection Vision System (DDVS).....	71
Figure 5.20: Airspeed with step throttle input.	72
Figure 5.21: Altitude with step elevator input.	73
Figure 5.22: Airspeed increases by changing the throttle control.	73
Figure 5.23: Altitude change when throttle change.	74
Figure 5.24: Airspeed fixed with elevator change.	75
Figure 5.25: Altitude change when elevator change.....	75

Figure 6.1: Fixed-Wing Maximum deflection at AOA = 30, Roll angle = 0 and Wind Speed 35 km/h.	78
Figure 6.2: Side and front view of wing setup during flow analysis in the computational domain.	78
Figure 6.3: UKF fusion of camera and strain gauge at AOA 10 deg, Roll angle 0 deg.	87
Figure 6.4: Measurement error, AOA 10 deg, Roll angle 0 deg.	87
Figure 6.5: UKF fusion of camera and strain gauge at AOA 10 deg, Roll angle 10 deg.	88
Figure 6.6: Measurement error, AOA 10 deg, Roll angle 10 deg.	88
Figure 6.7: UKF fusion of camera and strain gauge at AOA 20 deg roll angle 0 deg.	89
Figure 6.8: Measurement error, AOA 20 deg, Roll angle 0 deg.	89
Figure 6.9: UKF fusion of camera and strain gauge at AOA 30 deg roll angle 0 deg.	90
Figure 6.10: Measurement error, AOA 30 deg, Roll angle 0 deg.	90
Figure 6.11: Camera vs. Simulation and Strain Gauge deflection at AOA= 30 deg Roll angle = 0 deg.	91
Figure 6.12: Wing setup for lift force calculation.	93
Figure 6.13: Top view of wing setup during flow analysis in the computational domain.	96
Figure 6.14: Roll Angle vs. Aerodynamic forces for AOA 10 deg.	96
Figure 6.15: Roll angle vs. Aerodynamic forces for AOA 20 deg.	97
Figure 6.16: Roll angle vs. Aerodynamic forces for the AOA 30 deg.	97
Figure 6.17: Roll angle vs. Deflection for AOA 10 deg.	98
Figure 6.18: Roll angle vs. Deflection for AOA 20 deg.	99
Figure 6.19: Roll angle vs. Deflection for AOA 30 deg.	99
Figure 6.20: Angle of attack vs. Deflection.	100

Figure 6.21: Roll angle vs. minimum lift velocity for AOA 10 deg.....	102
Figure 6.22: Roll angle vs. minimum lift velocity for AOA 20 deg.....	102
Figure 6.23: Roll angle vs. minimum lift velocity for AOA 30 deg.....	103
Figure 6.24: Angle of attack vs. minimum velocity required for lift.....	103
Figure 6.25: Technical drawing of the UAV.	105
Figure 6.26: Isometric view of the model UAV.	106
Figure 6.27: The prepared model of the UAV.....	106

Nomenclature

- L Length of the wing (mm)
- b Width of the wing (mm)
- t Thickness of wing (mm)
- α Angle of attack (degree)
- β Roll angle (degree)
- V Velocity of wind (km/h)
- d Deflection of wing (mm)
- ρ Density of air (kg/m^3)
- F_L Lift force (N)
- F_D Drag force (N)
- C_L Coefficient of lift
- C_D Coefficient of drag
- W Weight of UAV (N)
- (X, Y, Z) coordinate system
- H Homography matrix
- $A(k)$ the transition matrix
- $B(k)$ the control matrix
- K_k Kalman filter gain
- $H(k)$ the measurement matrix
- P_x a covariance
- P_{vv} innovation covariance
- P_{xy} cross covariance

- Q system noise covariance matrix
- R observation noise covariance matrix
- $u(k)$ the input vector
- $v(k)$ the measurement noise
- v mean of residuals
- $x(k)$ the state vector
- X the sigma points
- $w(k)$ the process noise
- W weighted points
- $z(k)$ the measurement vector
- L rolling moment (N.m)
- m mass (kg)
- N yawing moment (N.m)
- p roll rate (rad/s)
- q pitch rate (rad/s)
- r yaw rate (rad/s)
- S wing area (m^2)
- u axial velocity (m/s)
- v lateral velocity (m/s)
- V_e reference equilibrium airspeed (m/s)
- w normal velocity (m/s)
- T thrust (N)
- α angle of attack(AOA) (radians)

- β sideslip angle (radians)
- δ longitudinal control angle (radians)
- δ_{th} thrust Control
- ρ air density (kg/m³)
- (θ, ϕ, ψ) euler angles (radians)
- ξ lateral control angle (radians)

Abbreviation Definitions

DDVS Deflection – Detection Vision System

AOA Angle of attack

UKF Unscented Kalman Filter

DLT Direct Linear Transform

DOD Degree of Divergence

EKF Extended Kalman Filter

FLAS Fuzzy Logic Adaptive System

ICP Iterative Closest Point

PID Proportional – Integral – Derivative

RANSAC Random Sample Consensus

SURF Speeded Up Robust Features

UT Unscented Transformation

Chapter 1

1 Introduction

Modeling a fixed-wing aircraft undergoing deformation requires a geometrical structural model combined with a reliable large motion aerodynamic model. The analyses and design of the fixed-wing configuration are important for designing unmanned aerial vehicles (UAV) [1] and its control system. The measurement technique proposed in this thesis is a visual method for determining aeroelastic deformation and behavior of elastic wings. The vision system is coupled with strain gauge sensors for obtaining more accurate results. The measurements from those sensors were integrated using the advanced sensor fusion algorithm. The fusion results are more accurate and reliable than measurements from the single sensor alone, and they could be input to the control and navigation system. This method aims to design an advanced control system for autonomous navigation.

The proposed technique is composed of visual and strain measurements methods for determining aeroelastic deformation and identifying the aerodynamic properties of the fixed-wings system. The vision system output results are integrated with strain gauge sensors for improved accuracy. This methodology is used to determine the shape of the wing and current aerodynamic UAV conditions important for the control system. The wing deformation induced by aerodynamic charge is determined using several spatial coordinates [2]. In this case, deformation refers to the change in the shape of a wing due to aerodynamic loading. Therefore, the measurement of wind tunnel models' deformations is

advantageous for further use, especially in high Reynolds number facilities where dynamic pressure is typically high [3].

Furthermore, it is crucial to fuse both information from the strain gauge and vision system measurements to determine the wing deformation accurately [4]. The system includes a ZED stereo camera for vision system measurement and a set of strain gauges placed on the wing surface to determine shape changes due to aerodynamic loading. A novel Deflection-Detection-Vision-System (DDVS) determines the wing's deformation and controls the wing. Aerodynamic characteristics are substantial in enhancing aircraft performance and capacity [5]. The thesis describes methods for obtaining fixed-wing deflection by comparing a known measurement instrument (strain gauge) to a stereo camera. This method includes testing in the wind tunnel using a model of an airplane wing at various wind speeds and wind directions, as shown in Figure 1.1.

For a system to be fully autonomous, the system should be able to make decisions at all times autonomously. This can be achieved when all the required aerodynamic parameters are identified during flight using a live system like the live vision calculation, which is implemented in this thesis. The DDVS implemented in this thesis measures the deflection for a fixed-wing utilizing a vision system and strain gauge. Figure 1.1 illustrates the model setup.

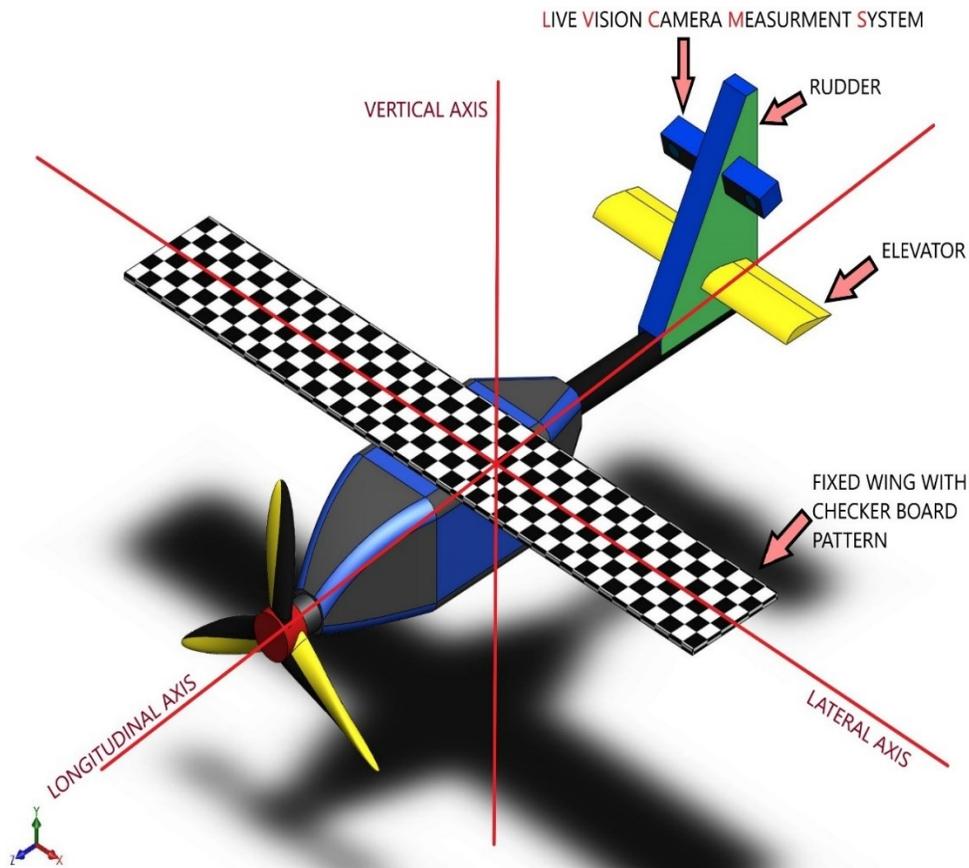


Figure 1.1: UAV axes and proposed initial concept design.

Sensor fusion is crucial for the localization, navigation, and control of an unmanned aerial vehicle [6]. Several sensors have been used in trying to get perfect control for UAVs. Currently, fusing different sensors is now used in the autonomous robot. This integration helps to correct an error in the system and helps to improve the system performance. The camera and strain gauge information will be more efficient in deflection measurement, and it is cheap and easy to calibrate. To achieve an excellent result using camera and strain gauges to calculate deflection, we considered fusing both sensors using Kalmar Filter (K.F.), Extended Kalmar Filter (EKF), and Unscented Kalmar Filter (UKF). K.F. is an

algorithm that only deals with a linear dynamic system with white Gaussian noise. Extended Kalman Filter (EKF) has been previously used in sensor fusion but still has linearization techniques error and has glitching issues with other noise. The decision to use UKF is to perform with nonlinear systems better and deal with white and colour Gaussian noise [7].

UKF is an advanced filtering method used for sensor data fusion, and it is more accurate than the Extended Kalman Filter (EKF) in several applications. This new sensor fusion technology enhances the performance of localization systems, which can assist with navigation. Robots require systems that can detect features and track targets, so precision vision is necessary [7]. A robot must be able to navigate an environment correctly based on three factors: Localizing the environment or getting the exact location of features, mapping the environment or detecting identical features, and making a decision or matching the features based on the information from localization and mapping [8].

The system presented in this thesis is the full integration of measurements from the vision system and strain gauge sensors to calculate the dynamic behaviour of an aircraft fixed-wing, and this information will be used for advanced control and navigation. The state of the error will determine the formation of the UKF model; that is, the state variable of the UKF will result from the difference in camera and strain gauge measurement. The new Deflection-Detection-Vision-System (DDVS) and strain gauges for UAV technology use 3D Digital Image acquisition integrated with strain gauge measurements. It tracks a random pattern on the surface of components to provide an accurate and dynamic measurement of the shape, displacements, and strain map all over the object's surface using the fused data from the camera and strain gauge [9]. Finally, by solving the static analysis

model in Fusion 360, the deflection at each node, strain, stress, and many more parameters of the fixed-wing setup was obtained using the CAD drawing as shown in Figure 1.2 and the render modelling in the wind tunnel as in Figure 1.3.

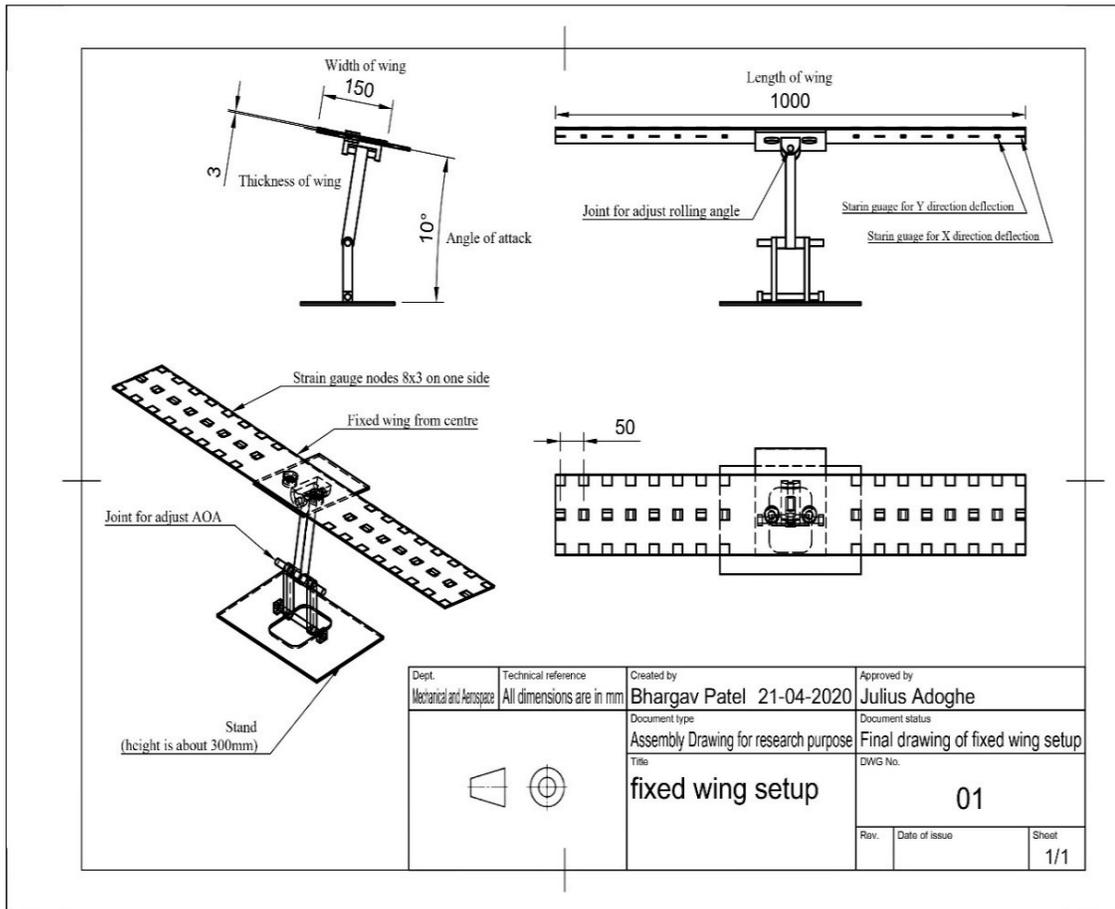


Figure 1.2: CAD drawings for the experimental setup.

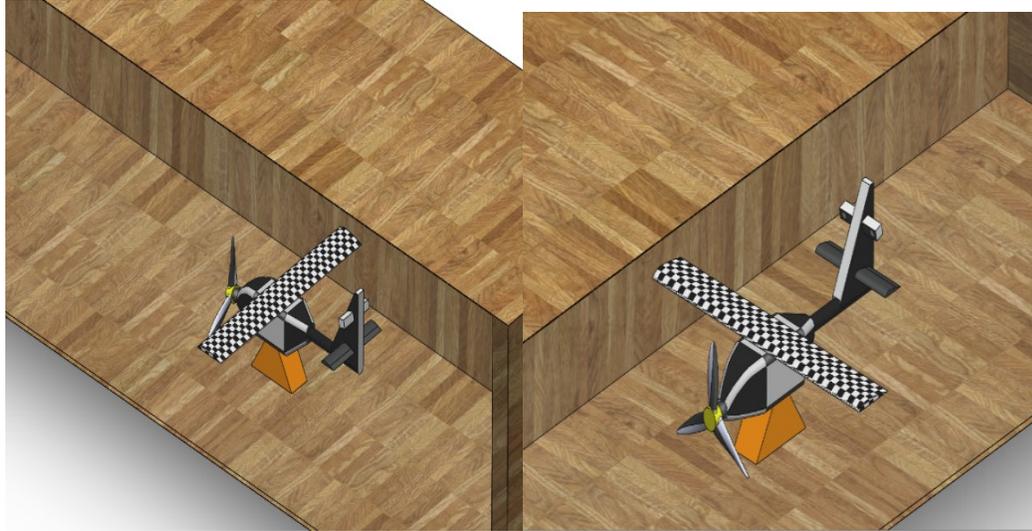


Figure 1.3 Render modelling of UAV experimental setup.

This thesis focuses on combining get information from the vision system and strain gauge, and the information will be used to design a controller for the UAV, taking both advantages of fuzzy control and PID control so that the dynamic performance, far away from the design point, can be improved [10]. The unmanned aerial vehicle (UAV) longitudinal model is established and controlled using Fuzzy PID [11].

1.1 Motivation

The focal of this thesis is to invent a system that will visually control UAVs. Most early UAVs are not autonomous when viewed from this perspective. Autonomous air vehicles are still a developing field, but the military is driving the development of this technology for the warfighter. Compared to the manufacturing of UAV flight hardware, the market for autonomy technology is immature and undeveloped. Regarding this, autonomy has been and may continue to be the bottleneck for future UAV developments. Advances in

autonomy may significantly impact the future UAV market's overall value and growth rate. Commonly, autonomy refers to the ability to make decisions without human intervention. To that end, the goal of autonomy is to teach machines to be "smart" and act more like humans. The keen observer may associate this with the development in artificial intelligence made popular in the 1980s and 1990s, such as expert systems, neural networks, machine learning, natural language processing, and vision. However, the mode of technological development in autonomy has followed a bottom-up approach, and the practitioners have driven mainly recent advances in control science, not computer science. Additionally, autonomy has been and likely will continue to be considered a branch of the control field. However, it is possible that in the near future, the two fields will overlap more, and practitioners and researchers will work together to foster rapid technological development.

1.2 Contributions and Publication

This thesis presents the theory, methods, and applications of using a strain gauge and camera to ensure accurate wind tunnel testing. The proposed method will identify the UAV wing's shape using two independent sensors, and the simultaneous measurement from those sensors is fused with the advanced sensor fusion method. The fusion results are more accurate and reliable than the single sensor alone, and this is done to design an advanced control system. The technique can determine static and dynamic aeroelastic deformation, structural deformation subject to strain, and a wind-tunnel model's attitude. The system's hardware includes a ZED stereo camera, sets of strain gauges, amplifier, computer, Arduino, illumination lights, and painted targets on a wind-tunnel model. Custom software includes routines for image acquisition, target tracking and identification, target centroid

calculation, camera calibration, and deformation calculations. Experiments were carried out at the Mechanical and Aerospace Department wind tunnel.

The major contributions of this thesis are summarized below:

- Design a system that uses vision system and strain gauge information to measure the deflection of the shapes of a fixed-wing.
- Using the information from the fused data to calculate the aerodynamic coefficient of fixed-wing UAV, calculating aerodynamic parameters from a vision system, and using the information to control UAV autonomously.
- 3D pose estimation transforms an object into a 2D image to give the 3D image. The need for 3D pose estimation arises from the limitations of feature-based pose estimation. There exist environments where it is difficult to extract corners or edges from an image. To circumvent these issues, the object is dealt with as a single body using free-form contours.

The publications related to this thesis work are listed below:

- J. O. Adoghe, J. Z. Sasiadek; "A Novel Approach to the Design of an Unmanned Aerial Vehicle for Humanitarian Initiatives," MITACS Annual Report on Innovation, Sciences and Development Economic Canada, pp 151, 2016-2017.
- J. O. Adoghe, J. Z. Sasiadek, M. A. Al-Isawi; "Control of Unmanned Aerial Vehicle Using Vision System and Sensor Fusion for Wing Shape Deflection Measurement" Journal of Complex Systems: Spanning Control and Computational Cybernetics: Vol. 1 Foundations; vol. 2 Applications Dedicated to Professor Georgi M.

Dimirovski on his anniversary Editors: Peng Shi, Jovan Stefanovski, Janusz KacprzykGT-C-CTP-07/2013, Springer 2022.

- J. O. Adoghe, J. Z. Sasiadek, M. A. Al-Isawi; “Unmanned Aerial Vehicle Fixed Wing Shape, and Deflection Measurement Using Vision System,” IEEE 25th International Conference on Methods and Models in Automation and Robotics (MMAR), 2021.
- J. O. Adoghe, J. Z. Sasiadek, M. A. Al-Isawi, H.A Bitlmal; “UKF Estimating the Deflection of Fixed Wing Measurement by Vision System,” The 17th International Symposium on Experimental Robotics, Malta, Mar. 22–25, 2021
- J. O. Adoghe, J. Z. Sasiadek, M. A. Al-Isawi, H.A Bitlmal; “Real-Time Localization and Mapping of a Mobile Robot using Visual Features,” Global Society for Research and Development 125 International Conference, Ottawa, 2020/9/28.

1.3 Thesis Outline

Seven chapters make up this thesis: Chapter 1 provides a concise introduction; Chapter 2 presents a literature review of computer vision, sensor integration, and autopilot design. Chapter 3 discusses the computer vision processing algorithms used for visual matching, tracking and strain gauge measurement. It also explains how the robust projective transformations estimation employ algorithms used to obtain the pose estimation. Chapter 4 presents the control system of the fixed-wing UAV. Chapter 5 analyzes the experimental setup and results. Also, discuss in details camera calibration and camera measurement for the system. Chapter 6 presents result verification and sensor fusion to estimate and integrate different data types using UKF. It also discusses the experimental tests and results

using the stereo camera and a strain gauge. It shows results for the DDVS system that classifies the wing shape to obtain the state vector of the UAV. An autopilot for both longitudinal and lateral motion control is designed and tested. This chapter also introduces future research work. Finally, chapter 7 concludes this work by reviewing its main findings and highlighting the contributions of the thesis.

Chapter 2

2 Literature Review

2.1 Computer Vision for UAV

A vision system (camera as a sensor) has a lot to play in the robot control system. This research proposes determining the aeroelastic deformation and behaviour of the wing of an unmanned aerial vehicle (UAV) using information from a stereo camera and stain gauge. The robot vision system is a fast-growing robotics section with applications in numerous fields used in any state that deploys human vision. The vision-based pose estimation technique compares and calculates the relative rotations and translations of two pictures of the same target. In [12] and [13], the authors present methods for reconstructing a camera's posture in relation to an object using an accurate geometric model of the item. Reference [14] shows pose estimates from homography and measurement error research, while [15] uses the broad homography decomposition to analyze robotic motion. The camera symbolizes the eye of a robotic system intended to capture an image and send it to a computer. The camera varies from the webcam on a laptop to a cell phone camera or digital still camera to a high-end industrial camera. This field presently has many limitations, but improvements are being made on a daily basis. Object recognition, material handling systems, surveillance, and component inspection in manufacturing are limited areas where cameras and vision technology are being combined in processing. Reference [16] developed a new technique for estimating relative posture between two pictures based on new decomposition. Reference [17] employed the Iterative Closest Point (ICP) algorithm to estimate position and attitude based on 3D data, while [18] showed the mathematical

preliminaries for obtaining the closest point. Three different focus lengths were used to attain the performance. Reference [19] proposed a closed-form solution for three or more points using unit quaternions. Reference [20] shows the posture estimation for UAV pose estimation. A lot of the work being done in surveillance is focused on reducing the processing power needed to do high-quality object tracking and recognition, which helps to reduce the power consumption needed for autonomous mobile robots. The problem with this system is mostly around mechanical setup, electrical control, and software required to implement and successfully develop these systems. My thesis research is based on fusing a calibrated Stereo camera and strain gauge to monitor the dynamic behavior of the wing of an unmanned aerial vehicle (powered hand glider). The camera's requirements to identify wing shape during a flight are calibration, central perspective image model, perspective transformation, homography, and using a stereo camera with a convex lens, which allows more light to pass and leads to much brighter images. In general, the camera is to have an arbitrary pose with respect to the world coordinate frame. The pose estimation problem is to determine the pose of the wing's coordinate frame with respect to the camera. Knowing the wing's geometry will help determine the position of several points (X, Y, Z) on the wing with respect to its reference frame. The camera's intrinsic parameters must also be known, and how the image is captured and the corresponding images are processed. To get the pixel coordinate of the different images from the camera, the feature correspondence will be determined by using the Harris corner feature, Speeded Up Robust Features (SURF), which will show the scale or consider Random sample consensus (RANSAC), which help to match which point belongs to a particular plane [20]. The

information from the vision system and strain gauge is used to control the UAV taking advantage of fuzzy control and PID control.

Unmanned aerial vehicles (UAVs) have been a popular research subject over the last few years. These vehicles have been used in civilian applications and military missions, including search, rescue, and atmospheric research [21, 22]. Studies have been carried out on small UAVs because of the low costs of the system and their effectiveness in special missions. Autonomous UAVs, like helicopters, are among the most popular robotic platforms because they are easy to control for vertical take-off and landing and stationary flight [23, 24]. UAVs can employ many possible methodologies in control. Sliding mode control is an efficient, robust control method widely used to control many complex systems [25]. However, chattering is a big drawback of sliding mode control [26]. The most popular controller in the past is a proportional-integral-derivative controller (PID). Analytical methods are used to tune the conventional structure of the PID controller [27]. An approach to control, stabilization, and disturbance rejection of the attitude subsystem of a quadrotor. A self-tuned PID is used to design the controller of the autopilot based on the longitudinal motion (altitude and speed) and lateral motion (heading angle) of the Aerosonde UAV [28].

The authors in [29] and [30] presented a method to guide and control a system-based vision system. They introduced a Fuzzy logic controller is designed to be compared with the self-tuned PID controller. In [31], a stable gain scheduling PID controller is developed based on nonlinear systems' grid point concept. The autopilot controller has a high accuracy of the tracking path and robustness regarding environmental disturbances, mainly winds. The small UAVs are sensitive to wind disturbance since their magnitude may be comparable to the UAV's speed [32]. Inventors can improve the performance of

the PID controller by using soft computing techniques such as fuzzy logic [33]. The Fuzzy PID controller is developed to enhance the performance of the longitudinal motion (pitch control) of the aircraft [34,35]. The authors [36] and [37] present a comparison between different controllers used with a dynamic model of a quadcopter platform. These controllers are an ITAE tuned PID, a classic LQR controller, and a PID tuned with an LQR loop. The conventional LQR method presented in [38] has effectively solved the problem that fixed-wing UAVs are disturbed by air current conveniently and have poor flight stability. The longitudinal control for the flying wing UAV based on LQR can make the flying wing UAV achieves satisfactory longitudinal flying qualities [39]. The authors presented a studying the control of the vertical moving for UAVs under PID, LQR, fuzzy control, and self-tuning fuzzy PID. They have investigated the controller's effectiveness on the state of vertical motion UAV [40].

This controller is explained and compared with PID and LQR controllers [41]. The robust control is presented in [42]. A CMMAC (Classical Multiple Model Adaptive Control) has been implemented to track quadrotor helicopters' performance against the linearization model error and uncertainties. A robust controller on an aerial manipulator is presented [43]. This controller can track the error in the external disturbance, and also, the stability is proved. This thesis mainly introduces a more robust controller schema and a more distinctive gain tuning procedure. The controller must provide a reference signal for altitude and airspeed. A dual fuzzy PID and LQR controllers are proposed, in this work, to analyze the effectiveness of the control designs, and the corresponding evaluation of the system performance is presented.

There has been limited research on nonlinear aeroelasticity and flight dynamics of flying-wing configurations. Weisshaar and Ashley [44] conducted some of the earliest work and were focused on the static aeroelasticity of flying wings with application to oblique wing aircraft [45]. There has been recent interest in flying-wing aircraft's aeroelasticity and flight dynamics. Banerjee [46] conducted an aeroelastic analysis of a tailless sailplane and compared it with a tailed airplane. The tailless configuration showed a considerable decrease in flutter speed and had a dominant contribution of the rigid-body modes making it a coupled flight-dynamic and aeroelastic instability. Love et al. [47] have also presented the coupled flight-dynamic/aeroelastic instability of a swept flying-wing aircraft and showed the existence of body-freedom flutter. The present paper describes a theoretical basis for the flight dynamic response estimation of a highly flexible flying wing. Various realistic design space requirements are considered, including concentrated payload pods, multiple engines, multiple control surfaces, vertical surfaces, discrete dihedral, and continuous pretwist. Researchers can use the code based on the theoretical development presented here in preliminary design and control synthesis. This work is a continuation of work conducted by the authors over the past decade in the area of nonlinear aeroelasticity of conventional wing-tail configurations [48,49]. The focus of the present work is on flight dynamics of flying-wing design.

2.1.1 Sensors Integration

Many researchers have focused their efforts on the guidance and navigation aspects of this problem. A navigation technique based on a visual servoing system combined with a six-degree freedom (DOF) robotic manipulator was used to capture a moving target [50]. The novel concept for moving identification objects in space and the trajectory design was

presented in [51]. In [52] and [53], they were focused on capturing the autonomous satellite using a visual servoing system. The pose of a large non-cooperative target was determined in [54] using two cameras were chosen to make the recognition of a selected rectangular feature object easier. The Unscented Kalman filter (UKF) was developed based on a neural network (NN) algorithm in [55]. The UKF and NN are integrated for nonlinear estimation states to enhance the nonlinear dynamic model. A novel adaptive UKF (AUKF) was developed by [56], and they are applied nonlinear joint estimation for time-varying states and parameters. This algorithm is based on a cost function generated from the error between the covariance matrices of innovation and their corresponding computations. The authors [57] presented a newly robust adaptive UKF that performs correction of the process noise covariance or measurement noise covariance depending on the type of the fault, and the algorithm is applied to the attitude estimation for a picosatellite. This paper presents the aircraft and UAV orientations analysis based on the AUKF in [58]. The proposed filter estimates the accelerations generated over steady cruising flights to accurately predict the roll and pitch orientation angles.

2.1.2 Classification of the Wing Shape

The Fuzzy Logic Control (FLC) has done preliminary work classifying deflections and controlling a flexible wing with a vision system. The authors of [59] concentrated on the static aeroelasticity of flying wings, with some work including aerodynamic coefficients and flying aeroelasticity. [60] describes a theoretical investigation of dynamic response estimation of a very flexible flying wing. [61] presents aerodynamic coefficients and structural displacements from wind tunnel studies and wing flexibility on Micro Air Vehicles (MAV). Non-contact videogrammetry techniques examined the dynamic

behaviour of flexible mini-UAV wings in wind tunnel experiments [62]. In contrast, videogrammetry techniques studied the deformation and attitude of static and dynamic wings in a wind tunnel model [63]. In [64], the authors suggested a wing twist technique based on camera photogrammetric, investigated structure deformation, and collected measurement data in-flight testing using digital image correlation [65]. [66] also measured and discussed static aeroelasticity. In addition, the uncertainty of videogrammetric techniques was applied to wind tunnel models, and fuzzy logic [67] was utilized to determine the surface deflections of the wind tunnel models. Furthermore, videogrammetric techniques' uncertainty was applied to wind tunnel models, and fuzzy logic [67] was utilized to discover the surface deflections of a flexible wing to accomplish the necessary roll angle.

2.1.3 Control System

Today, UAVs are becoming more beneficial due to their critical missions in civil and military applications [68]. Different control techniques have been developed and verified successfully for UAV flight control systems, such as optimal control, predictive control, adaptive control, and robust control [69]. The most popular controller in the past is a proportional-integral-derivative controller (PID). Analytical methods are used to tune the conventional structure of the PID controller [70]. The author in [71] presents a new approach to control, stabilization, and disturbance rejection of the quadrotor's attitude subsystem. A self-tuned PID is used to design the autopilot controller based on the longitudinal motion (altitude and speed) and lateral motion (heading angle) of the Aerosonde UAV [71].

Moreover, the fuzzy logic approach can compensate for the effects of nonlinearity and uncertainties and also has strong robustness independent of a mathematical model of the system. In addition, designing a fuzzy logic controller is fast, simple, and easily maintained because the rules can be understood and explained linguistically by human experts [72]. A Fuzzy logic controller is designed to be compared with the self-tuned PID controller. Autonomous UAVs, like helicopters, are among the most popular robotic platforms because they are easy to control for vertical take-off and landing and stationary flight [73], [74]. In [75], a stable gain scheduling PID controller is developed based on nonlinear systems' grid point concept. The autopilot controller has a high accuracy of the tracking path and robustness regarding environmental disturbances, especially winds. The small UAVs are sensitive to wind disturbance since their magnitude may be comparable to the UAV's speed [76]. The Fuzzy PID controller is developed to improve the longitudinal motion's performance (pitch control) [77]. This paper focuses on how to combine both advantages of fuzzy control and PID control so that the dynamic performance, far away from the design point, can be improved [78]. The unmanned aerial vehicle (UAV) longitudinal model is established and controlled using Fuzzy PID [79].

Chapter 3

3 Deflection Calculation

3.1 Machine Vision

Most of the image processing techniques presented in this thesis are for various applications. First, a description of how the camera is modelled as a pinhole camera and as a stereo camera, then the feature detection and tracking method is introduced to find relevant interest feature points helpful in identifying and tracking regions of interest in image sequences. Finally, different techniques are infixed to determine the 3D pose [80].

3.1.1 Stereo Vision Camera

Stereo vision is the process of extracting a depth of a 3-D scene from different points of view. The stereo vision technique uses two cameras with a horizontally aligned and fixed distance known as the baseline. The ZED stereo camera was chosen, as shown in Figure 3.1, to provide vision data and depth data. One of the reasons to choose this camera is that it can capture a high-resolution side-by-side video on USB3 [80]. Furthermore, this camera is already calibrated and comes with known intrinsic and extrinsic parameters. The epipolar constraint reduces the time of searching for corresponding points in two images. This can be reduced by searching over a full-size image to searching across just a line of the image [81]. This method can be computed by rectifying homography between images to make their epipolar lines axis-aligned and parallel, thus making the search for corresponding points very efficient. By using a vision system measurement technique, information from the plain image is transformed into a three-dimensional (3D) image. The relationship

between the 3D image in object space and the corresponding 2D image is achieved using a stereo camera as a sensor.



Figure 3.1: StereoLabs ZED camera [80].

The collinearity equations involve the target location (X, Y, Z) in object space to the resultant point (x,y) in the image plane. Considering the right and left eye retinal image, a perception of depth arises from the disparity of the 3D point of view. When projected under the perspective of the stereo camera, the difference in the 3D image location is expressed as [80];

$$d = X_l - X_r \quad (3.1)$$

where X_l is the position of the image from the left camera, X_r is the position of the image from the right camera, and d is the disparity.

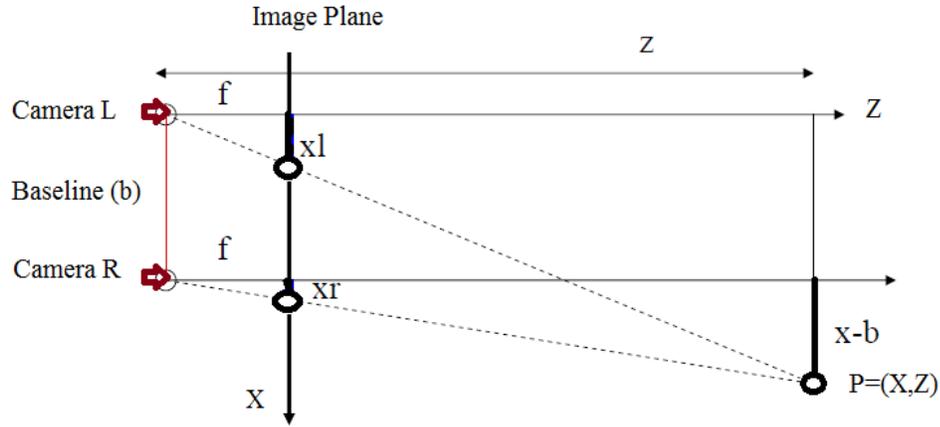


Figure 3.2: Model of the stereo camera.

Figure 3.2 shows the stereo camera optic axes. Y – axis is the axis perpendicular to the page, from a similar triangle. In the triangulation equation below, z is the image depth, f is the focal length, and b is the baseline [80].

$$\left[\frac{z}{f} = \frac{X}{x_l} \right], \left[\frac{z}{f} = \frac{X-b}{x_r} \right], \left[\frac{z}{f} = \frac{y}{y_l} = \frac{y}{y_r} \right] \quad (3.2)$$

Because the camera's intrinsic and extrinsic parameters have already been calibrated, it is possible to get the corresponding points quickly using the epipolar constraint. This approach is calculated by correcting the homography between images so that their epipolar lines are parallel and aligned, resulting in a highly efficient search for related points. The following is the stereo vision mathematical model: Assume a feature point, F , which corresponds to the scene's pixel points, and F_l , the picture created on the left camera C_l . The image formed on the right camera is C_r , and F_r is depicted in Figure 3.2. The projection matrix is as follows [80]:

- The matrix for the left camera

$$Z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \beta & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}_l \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_l \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_w \quad (3.3)$$

It can also be expressed as

$$Z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = \begin{bmatrix} n_{11}^l & n_{12}^l & n_{13}^l & n_{14}^l \\ n_{21}^l & n_{22}^l & n_{23}^l & n_{24}^l \\ n_{31}^l & n_{32}^l & n_{33}^l & n_{34}^l \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_w \quad (3.4)$$

Rearranging Eq. 3.4

$$\begin{bmatrix} u_l n_{31}^l - n_{11}^l & u_l n_{32}^l - n_{12}^l & u_l n_{33}^l - n_{13}^l \\ v_l n_{31}^l - n_{21}^l & v_l n_{32}^l - n_{22}^l & v_l n_{33}^l - n_{23}^l \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -u_l n_{34}^l + n_{14}^l \\ -v_l n_{34}^l + n_{24}^l \end{bmatrix} \quad (3.5)$$

- The matrix for the right camera

$$Z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \beta & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}_r \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_r \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_w \quad (3.6)$$

It can also be express as

$$Z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} n_{11}^r & n_{12}^r & n_{13}^r & n_{14}^r \\ n_{21}^r & n_{22}^r & n_{23}^r & n_{24}^r \\ n_{31}^r & n_{32}^r & n_{33}^r & n_{34}^r \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_w \quad (3.7)$$

Rearranging Eq. 3.7

$$\begin{bmatrix} u_r n_{31}^r - n_{11}^r & u_r n_{32}^r - n_{12}^r & u_r n_{33}^r - n_{13}^r \\ v_r n_{31}^r - n_{21}^r & v_r n_{32}^r - n_{22}^r & v_r n_{33}^r - n_{23}^r \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -u_r n_{34}^r + n_{14}^r \\ -u_r n_{34}^r + n_{24}^r \end{bmatrix} \quad (3.8)$$

3.1.2 Feature Detection and Tracking

The initial step in machine vision is feature point detection. It locates feature locations of interest in each frame, whereas feature tracking refers to correlating features from frame to frame [82]. As a result, it's critical to extract visual features like corners, blobs, and borders. Bay et al. [83] developed the Speeded Up Robust Features (SURF) detection method based on a Hessian matrix, which can be utilized to evaluate local picture structures due to its robustness and computing speed. It takes the following form: [84]:

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (3.9)$$

The convolution of the second-order image derivatives computed using the Gaussian function is I_{xx} , I_{xy} , and I_{yy} . In order to identify features of interest, the detector looks for a subset of sites where the derivative responses are high in two orthogonal directions or points where the Hessian matrix's determinant has a local maximum.

3.1.3 RANdom SAMple Consensus (RANSAC)

RANSAC is an iterative approach for estimating mathematical model parameters from a set of observed data that include outliers. Fischler and Bolles [85] first proposed it using a set of data with a large number of outliers. RANSAC is the most widely used robust estimation method for homography, according to [86]. The algorithm's concept is

straightforward. A random sample of four correspondences is chosen for multiple iterations, and the homography (H) is computed from them. According to whether or not they agree with H, other correspondence is categorized as an inlier or an outlier. After all the iterations have been completed, the iteration with the most inliers is chosen. The value of H can then be recalculated using all of the correspondence inliers in that iteration.

The RANSAC method can be summarized as [80]:

1. Initialize the number of estimation, threshold, and maximum inliers.
2. Calculate the number of iterations using the following equation [85], [87].

$$T = \frac{\log \epsilon}{\log (1-q)} \quad (3.10)$$

where: ϵ is the probability of incorrect identification of the model, and q is calculated based on the following equation:

$$q = \left(\frac{N_i}{N} \right) k \quad (3.11)$$

where N_i is the number of points belonging to the consensus set, N is the total number of points, and k is the minimal number of data necessary to define the model clearly.

3. Randomly select four feature pairs.
4. Check if these points are collinear; if so, go to step 3.
5. Compute the homography H.
6. Compute the Euclidean distance between two points $d(H P_i, P_i^*)$

7. If $d \leq$ the threshold, select the pairs that agree with the homography and record all the inliers.
8. Repeat steps 3 to 7 until the number of iterations has been reached.
9. Recompute the H estimate using all the inliers.

3.1.4 Triangulation

Triangulation helps determine the depth from disparity d and the coordinate of the 3D location for a stereo camera with parallel optical axes. Direct linear transformation (DLT) is one of the more common algorithms in machine vision. It helps determine the depth from disparity d and the coordinates of the 3D location for a stereo camera with parallel optical axes. The depth is inversely proportional to the disparity. From camera calibration, the focal length and the baseline can be determined. The corresponding point (x_r, y_r) for each (x_l, y_l) is achieved from the corresponding point distance of the images [87].

$$\text{Depth } Z = \left[\frac{f*b}{(x_l - x_r)} \right] = \left[\frac{f*b}{d} \right] \quad (3.9)$$

$$X = \left[\frac{x_l * Z}{f} \right] \text{ or } b + \left[\frac{x_r * Z}{f} \right] \quad (3.10)$$

$$Y = \left[\frac{y_l * Z}{f} \right] \text{ or } \left[\frac{y_r * Z}{f} \right] \quad (3.11)$$

Rearrange the eq. 3.11 and get:

$$\text{Depth } Z = \left[\frac{f^*b}{(Xl - Xr)} \right] = \frac{fb}{d} \quad (3.12)$$

The deflection can be found by the difference between the current depth (with load) and the depth without load.

3.1.5 Homography

Assuming that the camera moves in the $xi yi$ plane while the z - axis is orthogonal to $xi yi$. The camera views four or more planar and non-collinear feature points lying in a plane in front of the camera. These points expressed in the frame (F and F^*) are denoted by $(xi yi zi)$ and $(x^* y^* z^*) \in R$ respectively [88], and the normalized Euclidean coordinate vector can be expressed in terms of m and $m^* \in R$

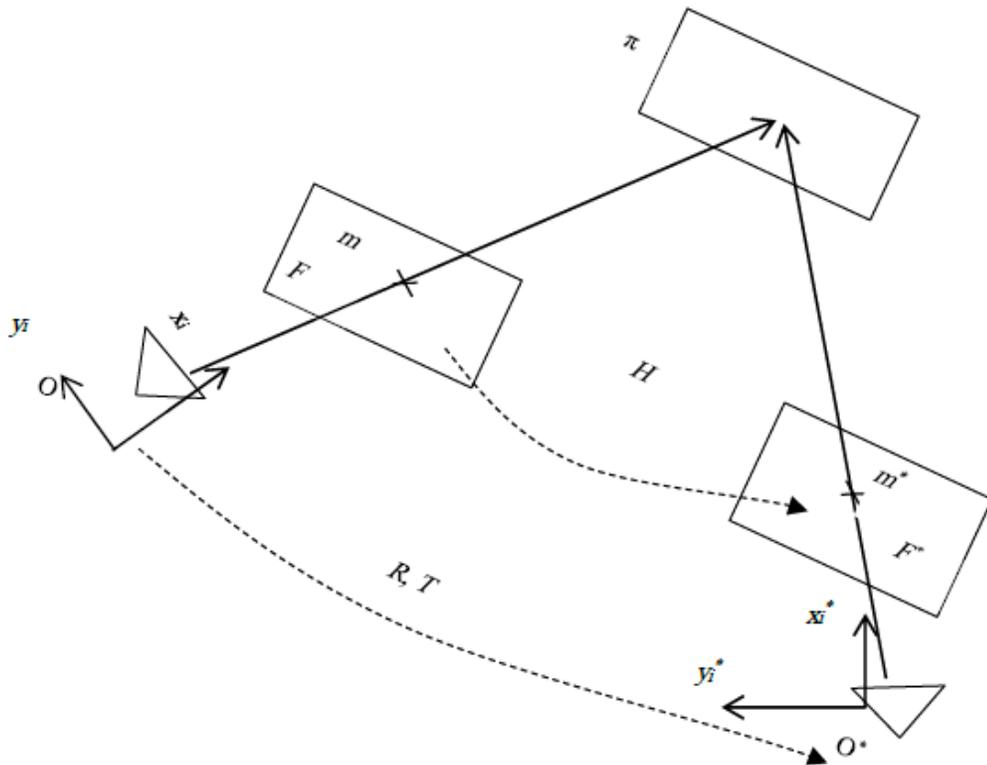


Figure 3.3: The geometry of a homography mapping [88].

$$m = \begin{bmatrix} x_i & y_i & 1 \\ z_i & z_i & 1 \end{bmatrix}^T \quad m^* = \begin{bmatrix} x_i^* & y_i^* & 1 \\ z_i^* & z_i^* & 1 \end{bmatrix}^T \quad (3.13)$$

Feature points m and m^* in the frame (F and F^*), respectively shown in Figure 3.3, are related by the homography matrix $H \in R^{3 \times 3}$ as:

$$m = \alpha_i H m^*$$

From projective geometry, the Euclidean coordinates for (m and m^*) can be expressed in image coordinates (pixel coordinates) as [88]

$$P = Km \quad \text{and} \quad p^* = Km^* \quad (3.14)$$

where $K \in R^{3 \times 3}$ are the constant calculated from camera calibration.

$$K = \begin{bmatrix} f_x & \beta & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

Where K is the intrinsic camera matrix, (f_x, f_y) is the focal length in the x and y -direction, respectively. (u_0 and v_0) are the principal point coordinate relating to the pixel dimension. β is the skew coefficient between the x and y -axes of the camera, which is often zero. P_i and P_i^* are the image coordinate represented by [88]:

$$P = [u_i \ v_i \ 1]^T \quad \text{and} \quad P^* = [u_i^* \ v_i^* \ 1]^T \quad (3.16)$$

From the above equation

$$P_i = \alpha_i K H K^{-1} P_i^* = \alpha_i G P_i^* \quad (3.17)$$

$$H = K^{-1} G K \quad (3.18)$$

where:

$$G = [G_{11}, G_{12}, G_{13} \dots \dots \dots G_{33}]^T \quad (3.19)$$

3.1.6 Decomposition of Homography Matrix

The main idea of this section is to extract information from the environment using the camera image and then determine the rotation and translation vector from O to O^* from the homography matrix. There are several methods for decomposition of the homography matrix, and three different algorithms are used to explore the direct method for the vision system. Those methods are a fast Homography decomposition technique, Iterative Closest Point (ICP), and Horn's Absolute Orientation [89].

3.1.7 Fast Homography Decomposition Method

The homography matrix H can be decomposed as [90]:

$$H = [R + \frac{t}{d} n^T] \quad (3.20)$$

where $H \in \mathbb{R}^{3 \times 3}$ is the homography matrix, the $R \in \mathbb{R}^{3 \times 3}$ and $t \in \mathbb{R}^3$ are the rotation matrix and translation vector from O to O^* , respectively. $d \in \mathbb{R}$ is the distance between the origin camera and the plane π , and $n \in \mathbb{R}^3$ is the plane's surface normal vector to the plane π [90].

According to the planner motion of the camera, the rotation R and translation t can be written as follows [90]:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

$$T = \frac{t}{d} = \left[\frac{t_x}{d} \quad \frac{t_y}{d} \quad 0 \right]^T = [T_1 \quad T_2 \quad 0]^T \quad (3.22)$$

where $\theta \in \mathbb{R}$ is the rotational angle around the z -axis between O to O^* , t_x and t_y the translate along x_i and y_i respectively, and d is the distance from the origin of O to the reference plane π . the H matrix can be calculated by substituting (3.21 and 3.22) into (3.20) [90]:

$$H = \begin{bmatrix} \cos(\theta) + T_1 n_1 & -\sin(\theta) + T_1 n_2 & T_1 n_3 \\ \sin(\theta) + T_2 n_1 & \cos(\theta) + T_2 n_2 & T_2 n_3 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{12} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

It is clear from (3.23) that the homography decomposition algorithm will be divided into three cases depending on the h_{13} and h_{23} [91].

3.1.8 Iterative Closest Point Algorithm (ICP)

An ICP algorithm tries to find a match between two collections of points. One of these sets could be a reference image, while the other could be a collection of data points that describe the ranges to specific locations on an item. For example, the ICP algorithm estimates camera translation and rotation [92]. A point set P of the source image point with Np and the destination points become X with Nx are given.

3.1.9 Horn's Absolute Orientation Method

The transformation between two Cartesian coordinate systems can be thought of as the result of a rigid-body motion and can thus be decomposed into a rotation and a translation.

Let the coordinates of the four points in each of the two coordinate systems be a $P_{A,i}$ and $R_{B,i}$ for two frames. The mathematical relationship between these sets of points is [93]:

$$P_{A,i} = sR_{A/B}R_{B,i} + T_{A/B} \quad (3.24)$$

where s is a scale vector, and $R_{A/B}$ and $T_{A/B}$ are rotation and translation matrices between frame A and frame B.

3.2 Deflection Calculation Using Strain Gauge

Strain (ϵ) is the fractional difference in the length of a substance due to the force applied. More specifically, the strain is characterized as a slight change in length. Strain can be either positive (tensile) or negative (compressive) and without dimension. The size of the measured strain is minimal and is often expressed as a microstrain [94]. The electrical resistance strain gauge is simply a length of wire or foil shaped into a continuous grid. The gauge is joined securely to the surface of the component under examination; hence the strain gauge will measure any change in the surface. Since the central equation for the electrical resistance R of a length of wire is [94],

$$R = \frac{\rho L}{A} \quad (3.25)$$

where ρ is the resistivity, L is the length, and A is the cross-sectional area. It follows that any change in length, and hence sectional area, will result in a change of resistance. The direct reading of the linear strain is acquired by the measurement of this resistance change with suitably calibrated equipment, and strain which may be expressed: [95]

$$\text{Strain } (\varepsilon) = \frac{\Delta L}{L} \quad (3.26)$$

For calibration, the equation of concentrated load at the free end is used;

$$\delta_{max} = \frac{Pl^3}{3EI} \quad (3.27)$$

In this thesis, we consider the tensile force acting on a fixed-wing. The load is uniformly distributed on a fixed-wing, which can also be seen as a flat cantilever beam. Since it is a uniform loading on the wing, the equation used in calculating the deflection of the wing during various airspeed, angle of attack, and roll angle is shown below [95];

$$\delta_{max} = \frac{\omega l^4}{8EI} \quad (3.28)$$

where E is the modulus of elasticity, I is inertia, l is the length of the wing, ω load per unit length, and P is the point load. The best way to calculate strain is using the resistance method is a Wheatstone bridge circuit. We can get the total voltage difference and difference resistance using this circuit.

$$V_O = \left[\frac{R_3}{R_3+R_4} - \frac{R_2}{R_1+R_2} \right] * V_{EX} \quad (3.29)$$

Null condition is satisfied when: $R_1 R_3 = R_2 R_4$. Gauge Factor $GF = (\Delta R/R)$. Knowing the GF, we can get the value of ΔR , which is known as the deflection in length [94]. Now

we are interested in determining the deflection and strain ratio for the cantilever beam to apply it on aircraft wings to find out deflection, as shown in Figure 3.4.

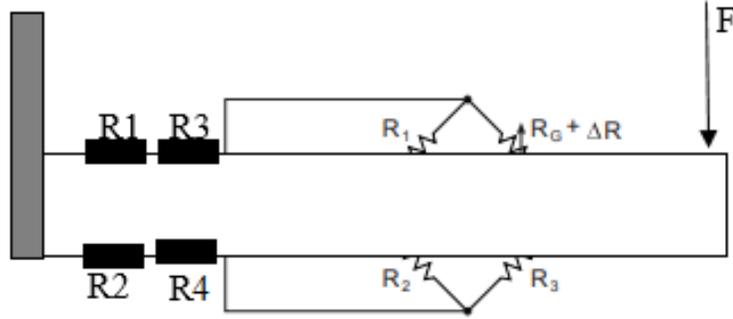


Figure 3.4: Deformation due to the external force.

$$\frac{V_O}{V_{EX}} = -\frac{GF*\epsilon}{4} \left[\frac{1}{1+GF*\frac{\epsilon}{2}} \right] \quad (3.30)$$

When the four strain gauges have equal nominal resistance, i.e., $R1 = R2 = R3 = R4 = R$, the deflection method Wheatstone bridge equation reduces to the linear equation

$$(V_0 / V_{ex}) = [\Delta R / R] \alpha (F) \quad (3.31)$$

$$V_0 = a_0 + a_1 W \quad (3.32)$$

where a_0, a_1 are some constants related to weight. Once an unknown weight is known, an unknown mass or density easily follows [95].

When a strain gauge is attached to a vibrating member, the inertia of its parts causes deformations, which can result in substantial errors in the strain readings. In the past, it was

common practice to reduce the strain gauge's mass while increasing its rigidity in order to reduce deformations and adjust for the small error that resulted. However, it is frequently impossible to build a strain gauge that is both light and rigid enough to achieve this [95]. When compared to the vision system measurement, the strain gauge error was negligible in the wind tunnel. The problem with strain gauges wiring could be resolved by using single Multi-stranded wires with tin coatings for the strain gauge applications instead of connecting several wires to each strain gauge.

3.3 Standard Deviation Calculations

Standard Deviation calculations were taken into account while measuring the repeatability and range of the measurements. The Range method is simple: for each part, subtract the smallest value from the largest value, and the difference is the range. We need to pause for a moment before computing the standard deviation based on the average range. To compute the average range, we must ensure that all of the range values are homogeneous. This necessitates the absence of any specific causes in the range values. This is usually examined by utilizing a Range Chart to look at the range values. The range values, average range, and upper control limit for the ranges are displayed in this chart. It would be unusual for a range value to exceed the top control limit for ranges. This indicates that one of the part's two values is incorrect. The individual performing the research should locate the problematic part and have it re-measured. We employ the new values if the range value goes below the upper control for ranges. The procedure of determining the ranges' suitability is crucial. If this is not done, we risk having an inflated average range. This will improve gauge repeatability, and we may incorrectly dismiss the gauge's utility [95]. The standard deviation formula used;

$$\sigma = \sqrt{\frac{\sum(x-\text{mean})^2}{n-1}} \quad (3.33)$$

where

x is a set of numbers

mean is the average of the set of numbers

n is the size of the set

σ is the standard deviation.

Chapter 4

4 Control System Design

4.1 Introduction

The challenge in designing a flight control system is developing a robust nonlinear control technique that can nullify the effect of the uncertainties and external disturbances, especially the wind on the UAV. The autopilot is most important in the autonomous system, which guides the UAV to complete missions. This chapter presents the proposed design of the whole autopilot (longitudinal and lateral) of UAVs for both the longitudinal dynamics motion (forward speed, pitching, and altitude) and the lateral dynamics motion (rolling and heading) [96], [97]. The autopilot is applied to the hang glider model depending on state-space linear models (longitudinal and lateral). In reference [98], the Linear Quadratic Regulator (LQR) controller was compared with the conventional PID controller for performance. The longitudinal and lateral motion control designs were also considered. The design was in two stages. The first is the inner loop (roll tracker), and the second is the heading related to guidance and control (outer loop).

4.2 Longitudinal Control

The longitudinal controller controls the velocity, pitch angle, and altitude. This is accomplished with two loops. The first loop is the inner loop or the pitch tracker, while the second loop is the outer loop, the altitude (h) and airspeed (Va) control. The longitudinal controller produces a proper input value ($\delta long$), and these parameters control the altitude and airspeed. These stages are illustrated in Figure 4.1 and Figure 4.2. The control bar and

the throttle are the inputs for the longitudinal motion controller. The control bar is used to control inner loops (pitch and pitch rate q) and outer loop height (h), while the throttle (δt) is used in the outer loop to control airspeed [90]. The altitude response loop is simulated by climbing the scenario to the desired altitude (h) and then executing the straight and levelling flight scenario. The velocity response loop is simulated in the same way.

4.3 Inner Loop Pitch Attitude Tracker Design

The longitudinal inner loop is the pitch attitude controller, which is the pitch damper (stability augmentation system (SAS)) [73]. This controller can be used when the plane is in a wings-level flight or climbing; the main job of the pitch tracker or pitch attitude hold is to maintain the value of pitch attitude (θ) matched with the reference commanded pitch. The block diagram of a pitch attitude holds control system is shown in Figure 4.1. The design can be performed using the short period approximation for aircraft dynamics, then adding an integrator to obtain pitch from pitch rate approximately [80].

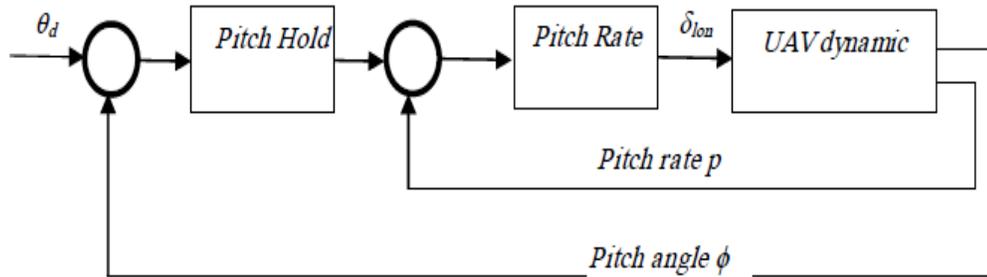


Figure 4.1: Longitudinal Guidance and Control System for pitch hold [80].

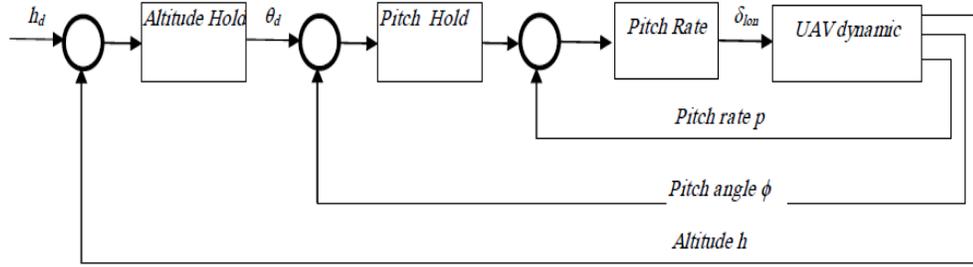


Figure 4.2: Longitudinal Guidance and Control System for altitude hold [80].

4.4 Outer Loop Altitude Hold Controller.

The outer loops were designed to achieve the tracking command requirements (altitude and cruise speed). Basically, the autopilot is constructed to achieve the tracking command requirements between the actual and desired altitude. The flight altitude $h(t)$ is not part of the mathematical model's state; firstly, it should be modelled. In straight and level flight conditions, the change of altitude can be written as the function of flight path angle and the velocity [98], so that,

$$\dot{h} = v_a \gamma = v_a (\theta - \alpha) \quad (4.1)$$

The longitudinal motion controller's performance is checked in straight, levelling, and level climbing flights [98]. The block diagram of the altitude hold control system is shown in Figure 4.2.

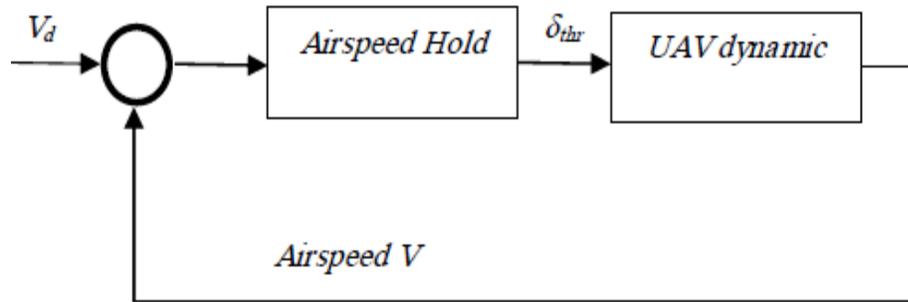


Figure 4.3: Longitudinal Guidance and Control System for speed hold [98].

4.5 Airspeed Hold controller

The airspeed hold control loop aims to keep the aircraft moving at a constant speed along a flight path. This loop was created with the cruising mode in mind. As seen in Figure 4.3, the loop regulates airspeed using the throttle, receives data from the airspeed sensor and controls the throttle. The throttle deflection depends on the current speed inaccuracy for speed changes. This instruction sends data straight to the aircraft model, bypassing the inner loop. The throttle command has a range of 0 to 1 only [80].

4.6 ANFIS Controller

A Neuro-Fuzzy system has been proposed by Jang [99]. Adaptive Neuro-Fuzzy Inference Systems [100] contain fuzzy logic and an artificial neural network. Fuzzy logic control deals with uncertainty existing in complex problems and is excellent for making a decision. It depends on their knowledge and database that were given for the plant, but it is an inability to learn. At the same time, The artificial neural network can minimize the error and optimize the controller parameters but cannot make a decision. Furthermore, the hybrid learning algorithm is used for training the membership function by trial and error and the

fuzzy rules at any number or shape of membership functions with fewer training times [100].

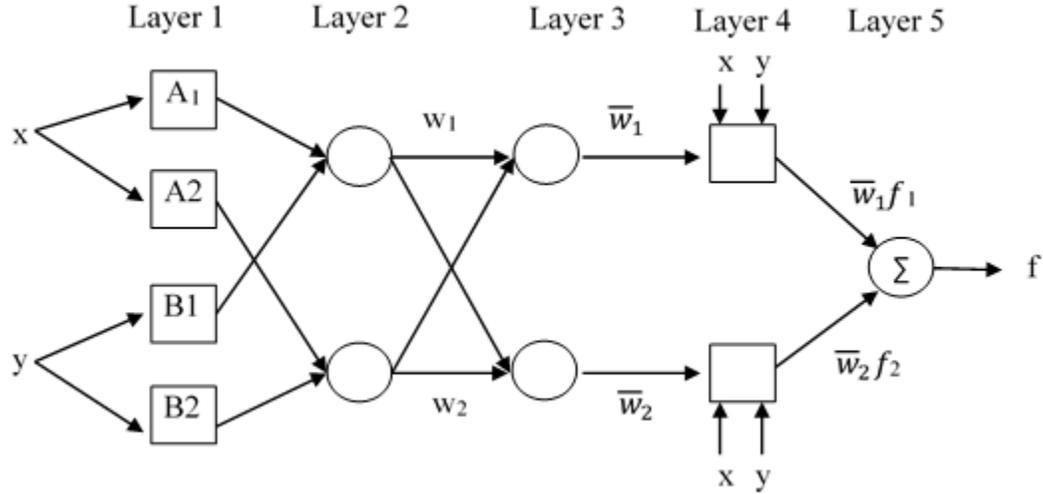


Figure 4.4: Adaptive neuro-fuzzy inference system structure [100].

Layer 1: The output for this layer (L_i^1) is a membership function (μ_{Ai}) of each input variable (x and y).

$$L_{Ai}^1 = \mu_{Ai}(x)L_{Bi}^1 = \mu_{Bi}(y) \quad i = 1, \dots, n \quad (4.2)$$

where n is the number of membership functions.

$$\mu_{Ai}(x) = \frac{1}{1 + \left[\frac{(x-c_i)}{a_i} \right]^{2b_i}} \quad (4.3)$$

where (a_i, b_i, c_i) is the parameter set that changes the shapes of the membership function. Parameters in this layer are referred to as the premise parameters.

Layer 2 is the product of the two inputs.

$$L_i^2 = w_i = \mu_{Ai}(x) \cdot \mu_{Bi}(y) \quad i = 1, \dots, n \quad (4.4)$$

Layer 3: Each node in this layer defines the normalized firing strength of the system

$$L_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1, \dots, n \quad (4.5)$$

Layer 4: The output is a summation of the previous outputs, such as:

$$L_i^4 = \bar{w}_i(p_i x + q_i y + r_i) \quad i = 1, \dots, n \quad (4.5)$$

where p_i , q_i and r_i are the parameters from the training process.

Layer 5: This is the overall output and the final total of all the layers.

$$L_i^5 = \sum_i \bar{w}_i (p_i x + q_i y + r_i) \quad i = 1, \dots, n \quad (4.6)$$

4.7 Controllability and Observability

R. Kalman established the notions of controllability and observability in 1960. The following is a list of terms to describe them.

- Controllability refers to the ability to control every state of a dynamic system when given a command; the system must be controllable.
- Observability: This refers to the ability to observe every state of a dynamic system; the system must be observable.

The segment investigates the system's controllability and observability based on the following conditions: If there exists a control for all vectors x , a system is termed controllable. The ability to measure the state from an observed output over a finite interval is characterized by observability. The controllability matrix is defined as [101]:

$$C_{ntr} = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (4.7)$$

The system is controlled if the matrix C_{ntr} has full rank, and the observability matrix has full rank if it is observable.

$$O_{bse} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4.8)$$

Controllability study identifies the longitudinal UAV system as controllable if the rank of the matrix is $C_{ntr} = n$, and the appropriate controller is built for the state feedback mechanism. The system is also observable if the matrix's rank is $O_{bse} = n$ [80].

4.8 PID Controller

Based on a frequency domain analysis of system dynamics, proportional-integral-derivative (PID) control is one of UAV's fundamental controller design methodologies. A closed-loop controller uses feedback to control a dynamical system's states or outputs. Disturbances and noise in the system are included in the feedback control utilized to lessen the influence of uncertainty [102]. There are numerous benefits to using feedback controllers over open-loop controllers:

1. Improve the response characteristics of the system.
2. Reject the effect of disturbance and noise.
3. Make the system stable.

Mathematically, a PID controller can be expressed as [99]

$$g = K_p e(t) + K_i \int_0^t e(t) dt + k_d \frac{d}{dt} e(t) \quad (4.9)$$

where K_p , K_i , and K_d stand for proportional, integral, and derivative gains. T is the signal transmitted to the process as a control signal. PID controllers are widely utilized in industrial applications because they are simple to design and implement. The controller tries to minimize the disparity between the desired and measurement signals, as shown in Figure 4.5. In comparison to another sort of controller, the controller is considerably easier to develop and execute [100].

The proportional gain minimizes the rising time, steady-state error, and noise at high frequencies. On the other hand, increasing the controller gain renders the system unstable. The integral gain corrects a steady-state inaccuracy by speeding up the process's progress toward a set point. It might contribute to overshoot because it responds to an accumulated mistake from the past, which is corrected by adding the derivative term. The derivative gain reduces the overshoot by slowing down the rate change of the control signal; it does not influence the steady-state error. The derivative term improves the combined controller process stability, but it also renders the process unstable and susceptible to error signal noise [103].

The Ziegler-Nichols rule is used to estimate the interval change of PID parameters in the controller gains technique [104], [105]. As a result, the PID controller delivers stable and reliable performance when the settings are correctly tuned. However, before the parameters are correctly optimized, a PID controller may produce a protracted settling period and constant overshoot until the error tends to zero. The PID controller is a simple mathematical tuning system that lacks direct knowledge of the control process.

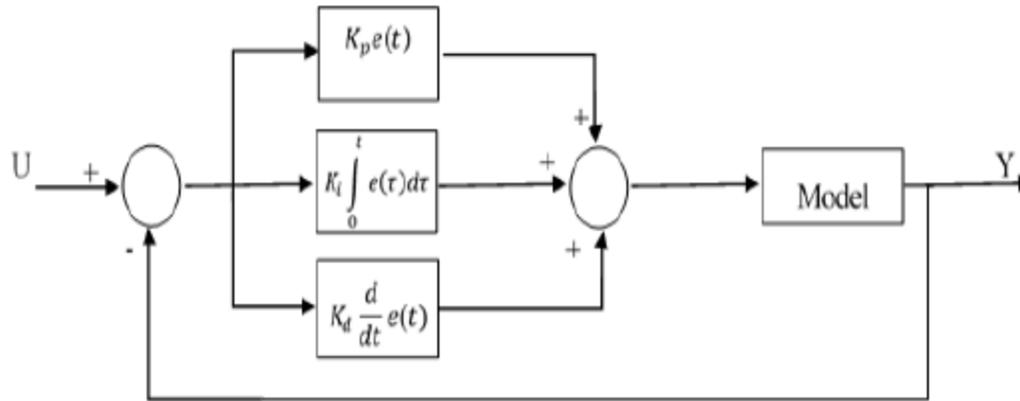


Figure 4.5 Block Diagram of PID Controller [100].

4.9 Fuzzy Logic Controller (FLC)

Fuzzy logic tools are a mathematical method to deal with the complexity and uncertainty in systems and were developed by Lofti Zadeh in 1965. The block diagram of Fuzzy logic is shown in Figure 4.6. As a simplification, the Fuzzy input and output variable sets are defined as {negative large, negative middle, negative small, zero, positive small, positive middle, positive large} or {NL, NM, NS, Z, PS, PM, PL}. The membership functions are, as seen in Figure 4.11 and Figure 4.12. It is noted that these functions are built around the value of Zero to ensure a minor variance in the control sensitivity. The mathematical model used for UAV control can be divided into longitudinal and lateral movement, considering the platform of Cessna 172, as shown in Figure 4.7.

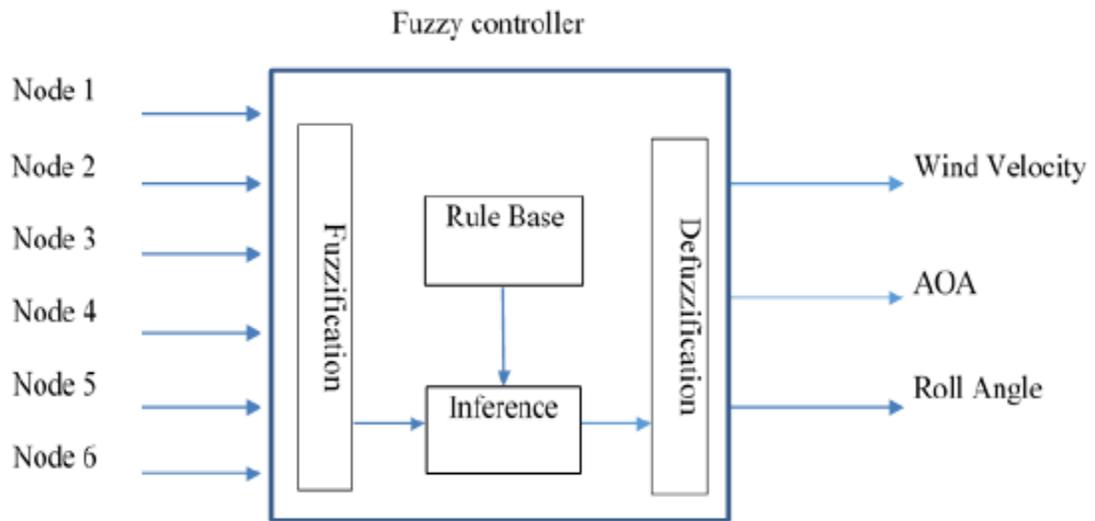


Figure 4.6 Block diagram of Fuzzy logic.

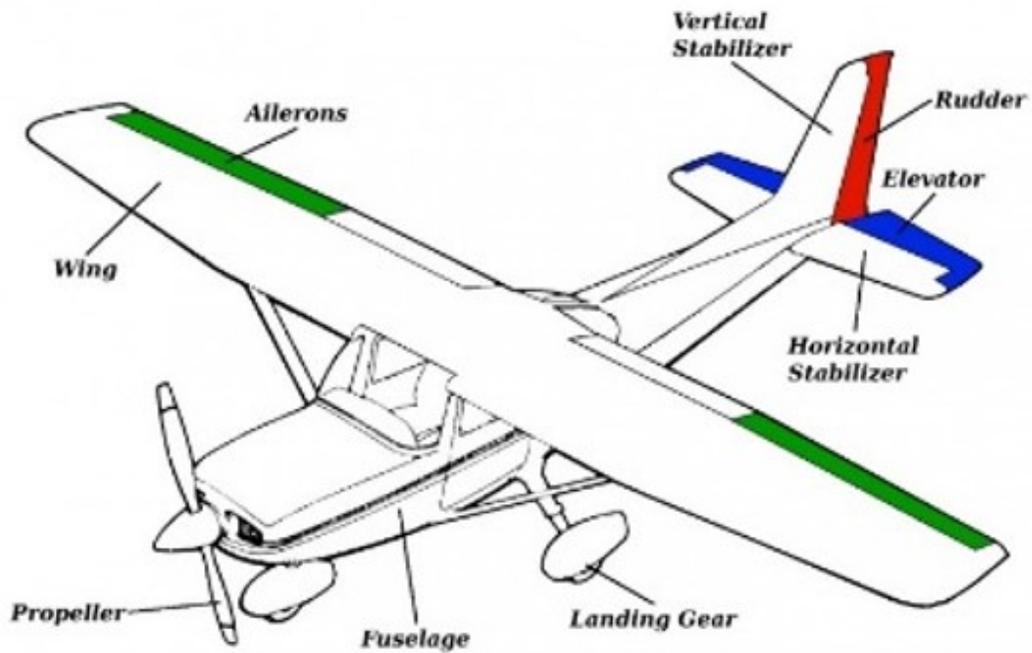


Figure 4.7: The Platform of Cessna 172 [98].

We can obtain this model based on the linearization of the equations in steady-level flight. The longitudinal dynamics are the aircraft's response along the pitch axis, as shown in Equation 4.10 [98].

$$\mathbf{M}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (4.10)$$

where

$$\mathbf{M} = \begin{bmatrix} m & -X_{\dot{w}} & 0 & 0 & 0 \\ 0 & (m - Z_{\dot{w}}) & 0 & 0 & 0 \\ 0 & -M_{\dot{w}} & I_{yy} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} X_u & X_w & (X_q - mW_e) & -mg \cos \theta_e & 0 \\ Z_u & Z_w & (Z_q + mU_e) & -mg \sin \theta_e & 0 \\ M_u & M_w & M_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\sin \theta_e & -\cos \theta_e & 0 & U_e \sin(\theta_e) + W_e \cos(\theta_e) & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} X_{de} & X_{\tau} \\ Z_{de} & Z_{\tau} \\ M_{de} & M_{\tau} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and C is the identity 4×4 matrix.

$\mathbf{x}^T = [u \ w \ q \ \theta]$ is the state vector of the longitudinal motion, M is a mass matrix, A is the system transition matrix, and B is the control matrix. $\mathbf{u}^T = [de \ \tau]$ is the vector of control input, where de and τ are the elevator and throttle control inputs. h is the altitude, q is the pitch rate, θ is the pitch angle, u is the forward velocity, and w is the vertical velocity.

4.10 Controllers

The thrust force and control surface considerably impact the aircraft's airspeed and altitude controls. The longitudinal control angle adjusts the thrust force generated by the throttle control and the control surfaces (Elevator). The proposed control techniques are executed, and the resulting results are reported for a full Cessna aircraft model. Fuzzy-PID with LQR controllers was designed in Figure 4.8 to ensure that the aircraft stabilized within a particular threshold.

4.10.1 LQR Controller

LQR is an optimal control approach using closed-loop control and linear state feedback [98]. The state feedback controller LQR, shown in Figure 4.9

In LQR, a cost function for optimal control performance is:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (4.11)$$

The vector u in eq. (4.11) is important to know that it minimizes the quadratic cost function, leading to optimal feedback control law represented in eq. (4.12) [98].

$$u = -Kx \quad (4.12)$$

The steady-state optimal gain is determined by using the Riccati equation as below [98]

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (4.13)$$

where Q and R are the state-cost and performance index matrices, respectively. The weight matrices of Q and R are significant in the LQR method and are symmetric and nonnegative matrices. They affect the control performance. These matrices are determined by the experience of engineers who are familiar with the controlled system [98].

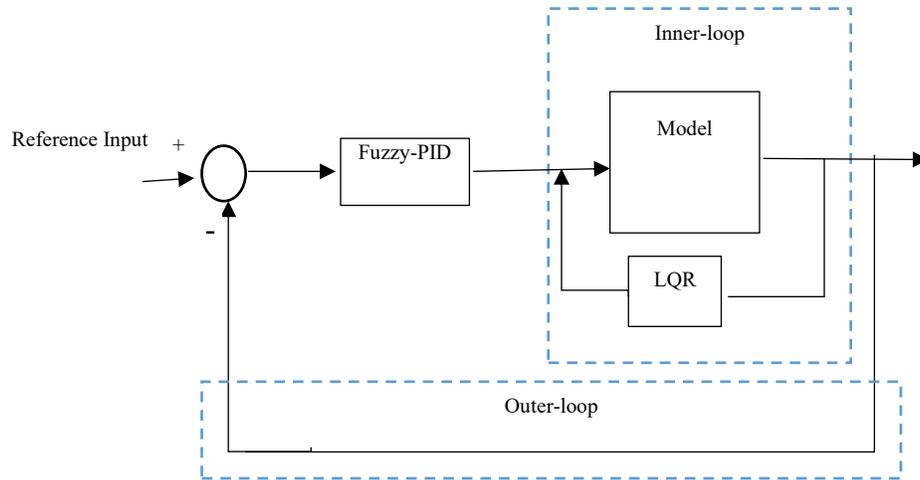


Figure 4.8:Block diagram of the fuzzy-PID controller with LQR.

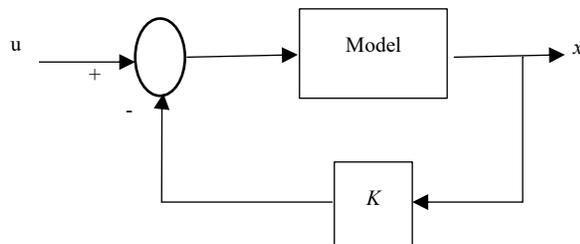


Figure 4.9:Block diagram of the LQR controller.

4.10.2 Fuzzy Logic Controller (FLC)

Lofti Zadeh invented fuzzy logic tools in 1965 as a mathematical tool for dealing with ambiguity and uncertainty in complex problems. It gives a method for dealing with information imprecision and details. A Fuzzy control system comprises three fundamental segments, as shown in Figure 4.10 [109].

1. Fuzzification – convert data into fuzzy or Membership Functions (MFs) data.
2. Fuzzy inference process – derive fuzzy output by combining membership functions and control rules.

- Defuzzification – determine each output and set them into a table using a different method.

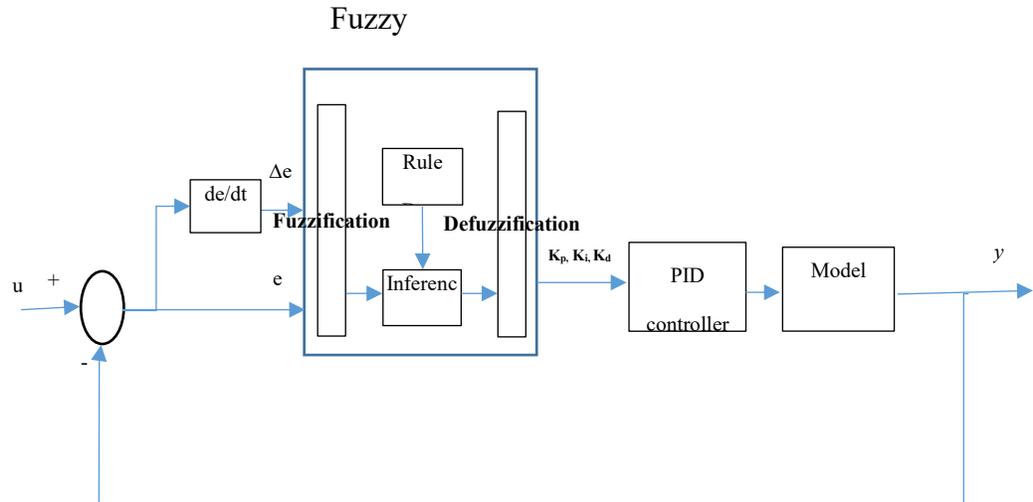


Figure 4.10: The blocks diagram of the fuzzy-PID controller

From a design standpoint, fuzzy-PID control techniques can increase dynamic performance. The intrinsic benefit of using fuzzy logic to self-tune PID gains is the ability to create adaptive controllers. The error, $e(t)$, and the error gradient, Δe , are the inputs of Fuzzy PID, and the outputs are PID gains. The input signals are converted to fuzzy sets in self-tuning for fuzzy, and then nonlinear mapping converts the inputs to PID gains. The block diagram of this control method is shown in Figure 4.10 [98].

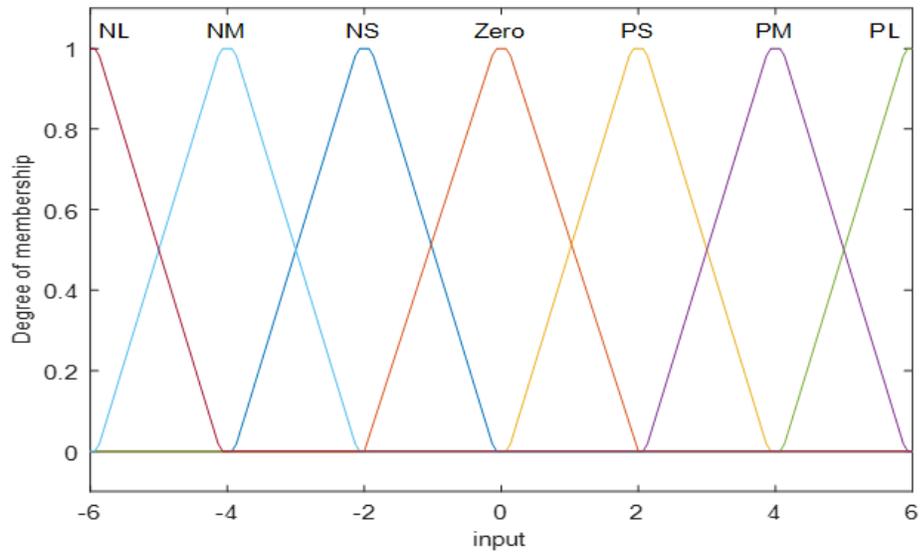


Figure 4.11: The membership function of the input variables

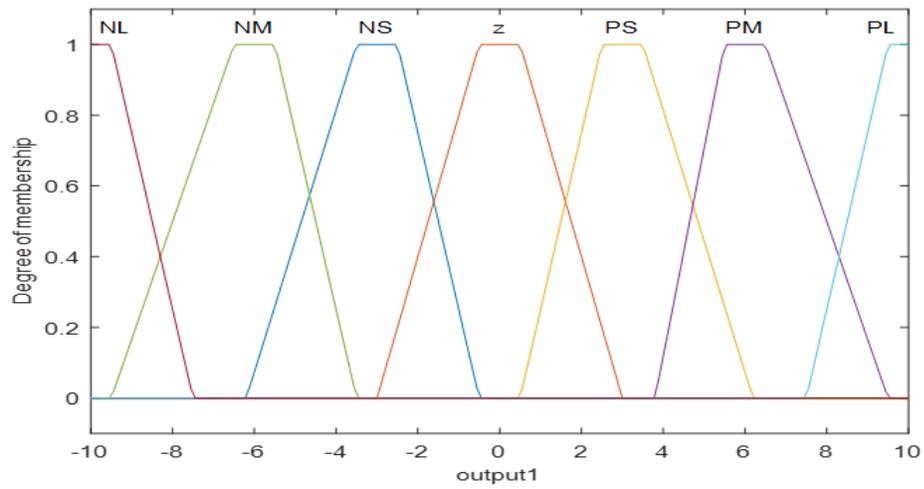


Figure 4.12: The membership function of the output variables

Table 4.1: Rule of KP, KI, and KD.

K_P K_I		Δe						
		NL	NM	NS	Z	PS	PM	PL
K_D								
e	NL	PL,NL,NS	PL,NL,NM	PL,NM,NL	PL,NM,NL	PM,NS,NL	PS,Z,NL	Z,Z,Z
	NM	PM,NL,NS	PM,NM,NS	PS,NM,NM	PS,NS,NM	PS,NS,NM	Z,Z,NS	Z,Z,Z
	NS	PM,NB,Z	PS,NM,NS	Z,NS,NM	Z,NS,NM	Z,Z,NS	NS,PS,NS	NM,PS,Z
	Z	PM,NM,Z	PM,NM,NS	PS,NS,NS	Z,Z,NS	NS,PS,NS	NM,PS,NS	NM,PM,Z
	PS	PM,NM,Z	PS,NS,Z	Z,Z,Z	Z,PS,Z	NS,PS,Z	NS,PM,Z	NM,PM,Z
	PM	PS,Z,PM	Z,Z,PS	NS,PS,PS	NS,PS,PS	NM,PM,PS	NM,PL,PS	NB,PL,PL
	PL	Z,Z,PL	Z,Z,PM	NS,PS,PM	NM,PM,PM	NM,PM,PS	NL,PL,PS	NL,PL,PL

Based on the different e and Δe taken from the experiences, which summarized the UAV system response [97], it can be inferred in Table 4.1.

Chapter 5

5 Experimental Setup and Results

5.1 Introduction

In this thesis, the experiment is to verify the Deflection-Detection-Vision-System (DDVS) that measures the deflection of a fixed-wing UAV model using the vision system, strain gauge, and simulation. The experiment took place in a static setup in the wind tunnel with various positioning of the camera, and we plan to mount the camera on the vertical stabilizer of the UAV in real-time. The platform being used in the wind tunnel is a 100 cm span of acrylic material, which is fixed at the midpoint to allow deflection at both ends. The experiment was performed at the Carleton University wind tunnel to validate the result of the vision system with that of the strain gauge and simulation. The result at various wind speeds, 10 km/h, 20 km/h, and 30 km/h, with different angles of attack and roll angles ranging from 0 degrees to 30 degrees, respectively. Both camera and strain gauge measurements were done simultaneously, which helps to limit error with the time difference.

5.2 Camera Calibration

Calibration is to determine the accuracy of an instrument to eliminate or reduce bias in an instrument's readings over a period. We calibrated both camera and strain gauges in this experiment. Calibration is a critical factor in vision systems as it helps determine the relationship between the object and its location in the global coordinate system. The MATLAB Toolbox shows the various methods of calibrating a camera to determine the

intrinsic parameters referring to the focal length and extrinsic parameters: the camera's 3D position regarding the world coordinate system, lens distortion, and principal point. This toolbox's inputs are a series of images of a model chessboard plane covering the calibration points. Calibrated chessboard edges are used as the reference points, as shown in Figure 5.1. Using the measured homography matrix, one may use the error in the following image to evaluate the error function after transferring points from the first image. All surveyed procedures were configured and checked under the same picture noise conditions. MATLAB, with a model chessboard, a camera calibration toolbox is used to get intrinsic and extrinsic parameters for the sensor, as shown in Figures 5.2 and 5.3.

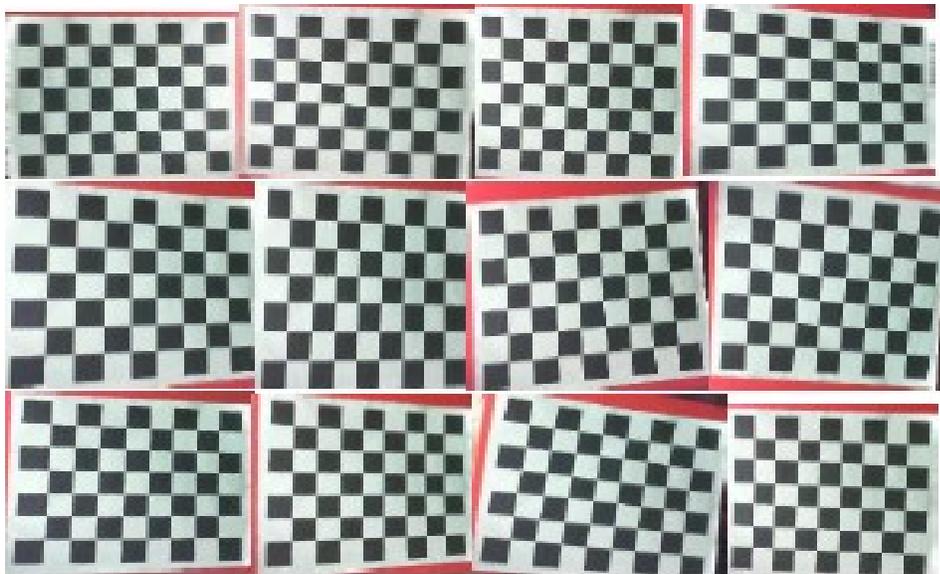


Figure 5.1: Calibrated chessboard.

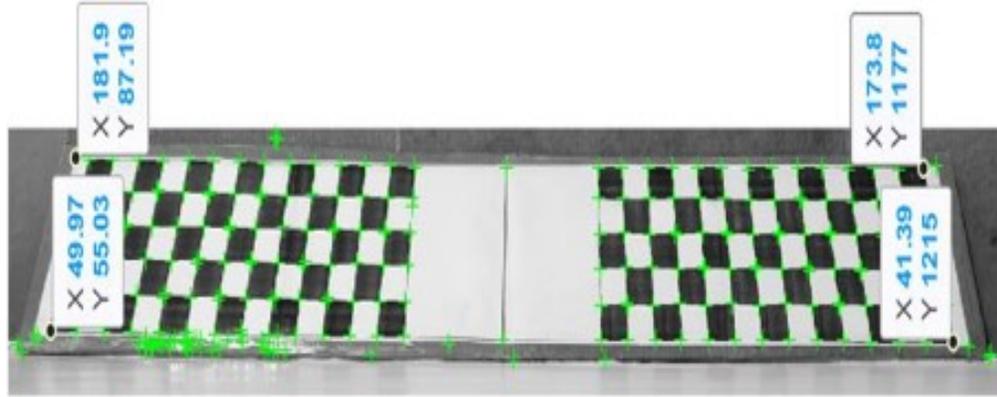


Figure 5.2: Camera calibration toolbox for MATLAB with a model chessboard.

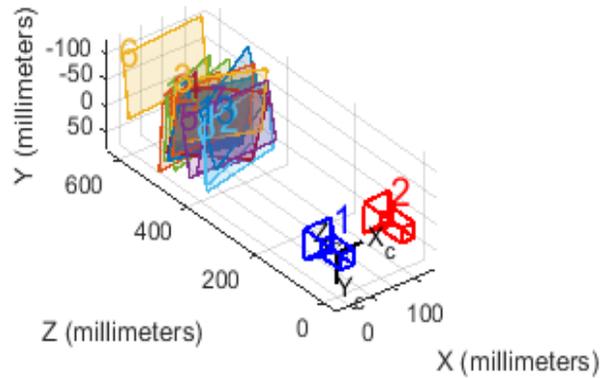


Figure 5.3: Camera extrinsic parameters.

The extrinsic matrix K from the camera calibration is shown as:

$$K = \begin{bmatrix} 3380.8 & 0 & 1374 \\ 0 & 3371.6 & 978.4 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

5.3 Feature Matching and Tracking

Finding corresponding features from two images consists of two phases. First, using Speeded-Up Robust Features (SURF) methods to extract features from the various images,

and then matching key points between the two images using Random Sample Consensus (RANSAC). The RANSAC algorithm was used to sort out the SURF algorithm results and delete the outliers. The choice for combining the RANSAC and SURF methods is to eliminate flaws in the existing matching techniques, such as high mismatch and low computational efficiency. The wing shape was identified using the deflection points, and the shape of the wind was classified using roll angle, pitch angle, and airspeed. SURF and RANSAC algorithms are methods used for object recognition, with sets of real-life objects using their 2-D images in each case. The features are invariant to image scaling, translation, and rotation and somewhat invariant to lighting and 3D projection variations.

Subtracting the target's distance from the camera at different conditions helps to measure the deflection at a selected location on a fixed-wing. The target needs to be visible enough to acquire a perfect image. Figure 5.4 shows 48 targets or checkmarks marked on fixed-wing black surfaces; a stereo camera will track these points. The fixed-wing used in the wind tunnel, as shown in Figure 5.5, has twenty-four checkmarks calibrated on both sides. These checkmarks serve as the features camera used to measure the deflection by matching image features, as shown in Figure 5.6. The checkmarks serve as targets placed on the model's surface, where there is a high possibility of deflection. Since the model's surface is dark, a white target is preferable indoors for the camera to easily detect the marks. The strain gauge is placed at every checkmark of the middle row to measure the deflection, and a total number of ten strain gauges were used. The marked target points are 7 cm apart, each from the tip to the wing's fixed point. The Wheatstone bridge circuit connected to the Arduino board converted the strain gauge information into deflection.

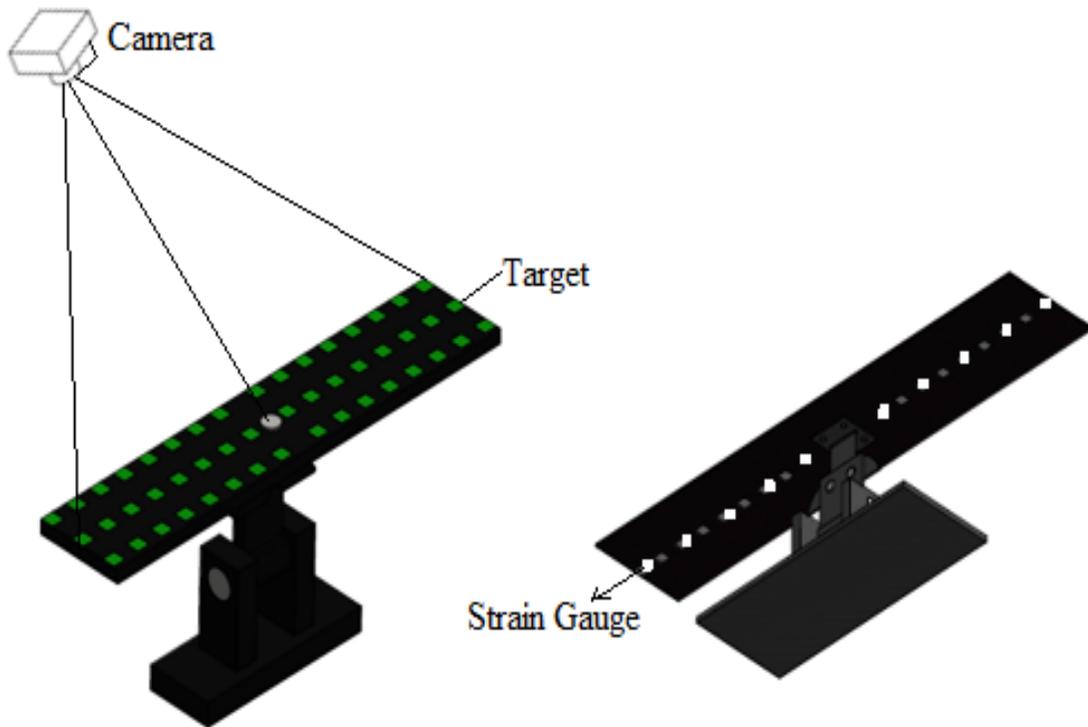


Figure 5.4: Points Position (stain gauge and target points).

The points' position in the fixed-wing is shown in Figure 5.4. The first step was to capture the image of the wing using the ZED stereo camera. Then, we calculated the deflection for each point after extracting and matching the features by using SURF and RANSAC approaches, as shown in Figure 5.6.



Figure 5.5: Front and rear view of the fixed-wing in the wind tunnel.

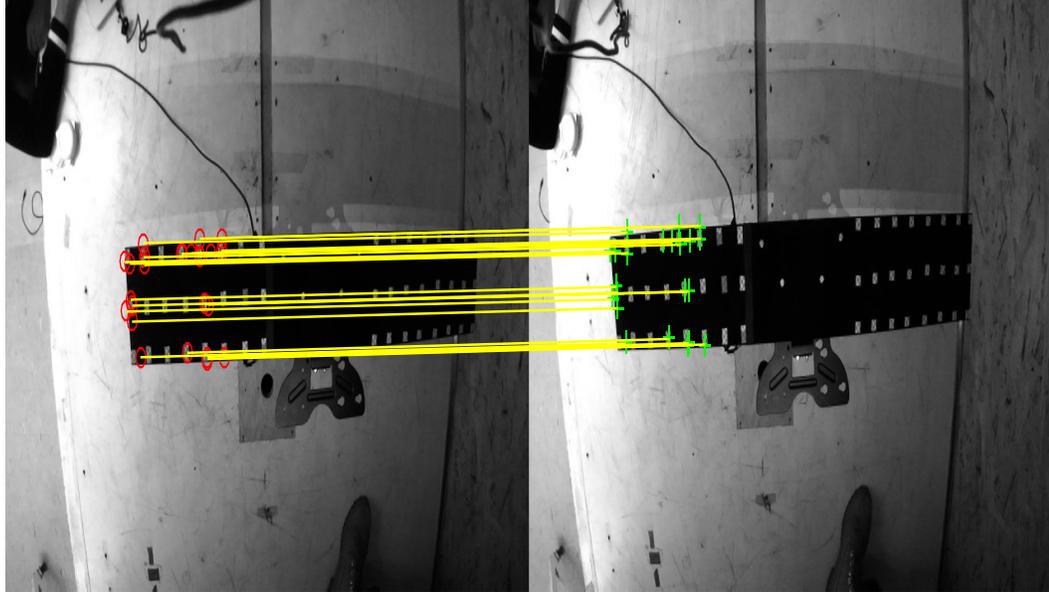


Figure 5.6: Matching points.

5.4 Experiments with Camera Measurement System

The vision system captures images of the initial condition when there is no load on the wing, and the memory is saved. In capturing images from a live camera, the algorithm for snapping photos in the loop is needed, and the limit of that loop ranges from 1 to N , where N is the number for the last image. Every time, the N^{th} image will be saved in the system when under the application of wind-load. The wing's deflection measurement needs to apply the match-point algorithm, but we need to process the images first. These processes include a colour image's conversation into a gray colour scale and match each checkerboard's cross point.

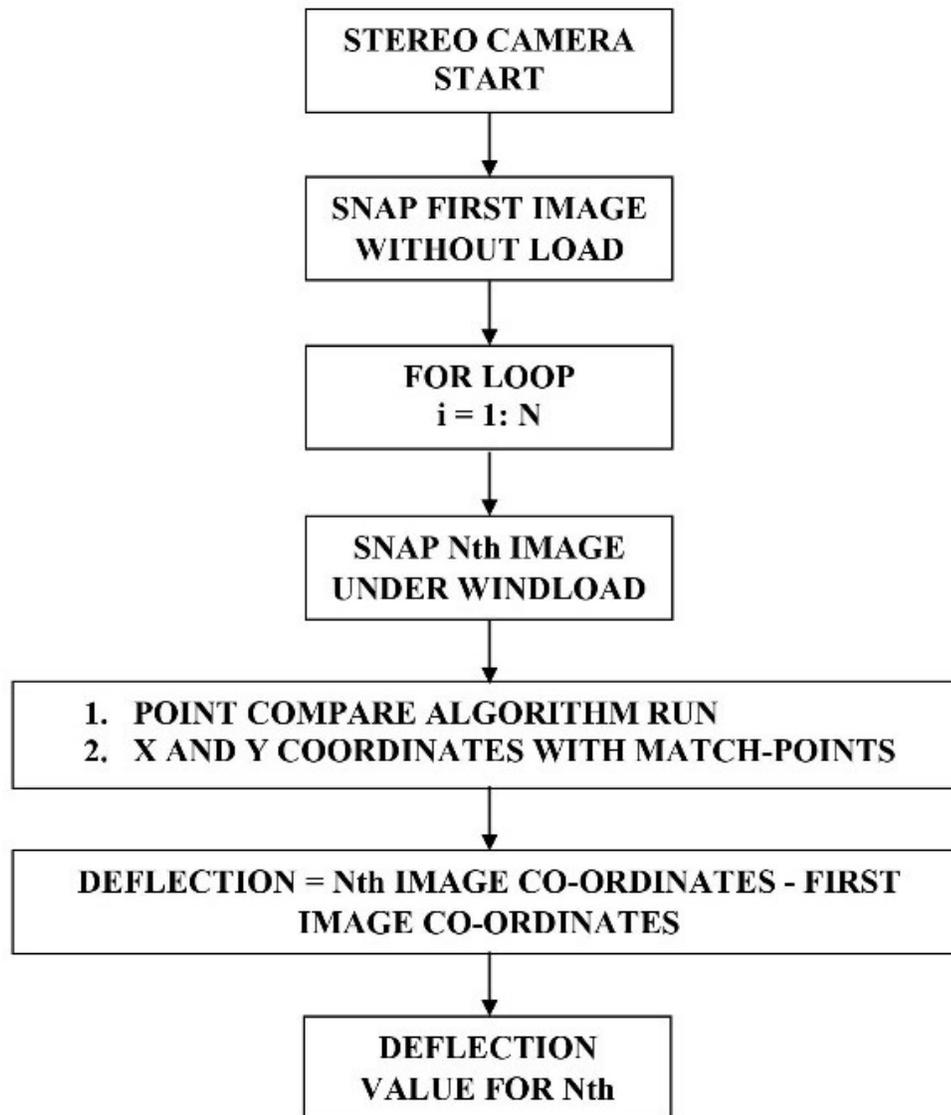


Figure 5.7: Process flow of live vision camera measurement system.

The deflection measurement of the wing needs to apply the match-point algorithm, but first, we need to process the images. These processes include the conversion of a colour image into a gray colour scale, matching each cross point of the checkerboard. To calculate the deflection, subtract the cartesian coordinate of all points for image number 1 to the N^{th} image from the initial image. These images are considered for the calculation of deflection

in live conditions. For this experiment, the N value is 13, and the values were chosen based on the number of images captured by the camera per ten seconds. There would be some pause between capturing images because the deflection measuring algorithm will take time for the solution. So, at the end of the loop need to pause the capturing algorithm so it can capture images lively and with full control. The end loop value deflection will show on display, and those algorithms will run in a loop until its limit and show the deflection values. As a result, deflection values will show in the live conditions of cameras. The procedure for the live camera-vision measurement system works by the camera, so the camera position is the first important step in the process and setting the camera in perfect steady condition from where the camera can cover the entire image/video. The vision system feeds the live video into the system. The system will convert the video into images at an interval of time, then use the image to calculate the area using a corner detection algorithm.

5.5 Experimental result

The DDVS and strain gauge starts to measure the deflection of all points in the wing at a speed starting from 10 km/h to 35 km/h, AOA of 10 degrees, and roll angle of 0 degrees, as shown in Figure 5.8. Then the deflection is measured as the roll angle is increased to 10 degrees, as shown in Figure 5.9, and there was a decrease in deflection. In Figures 5.10 and 5.11, the deflection was measured when the roll angle was 0 degrees, and AOA became 20 and 30 degrees, and there was an increase in the deflection, respectively. The test was repeated at speeds ranging from 10 km/h to 35 km/h, and the roll angle was 10 degrees at AOA 30 degrees and 20 degrees roll angle at 30 degrees AOA as shown in Figures 5.12

and 5.13, respectively, and both graphs show the deflection decreases with an increase in roll angle.

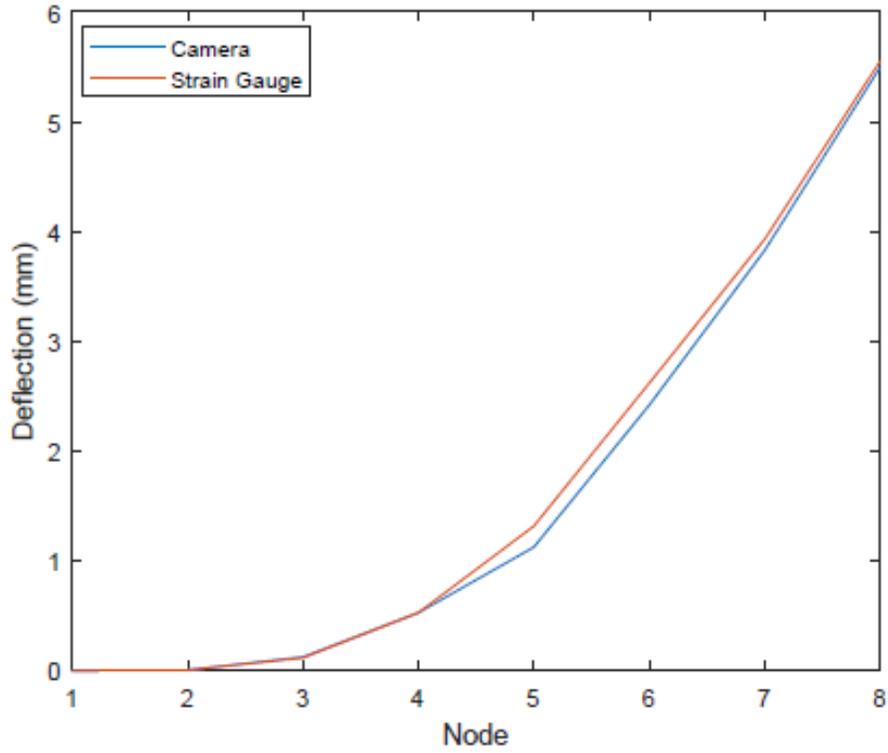


Figure 5.8: Deflection vs. node no., AOA = 10 deg, Roll angle = 10 deg.

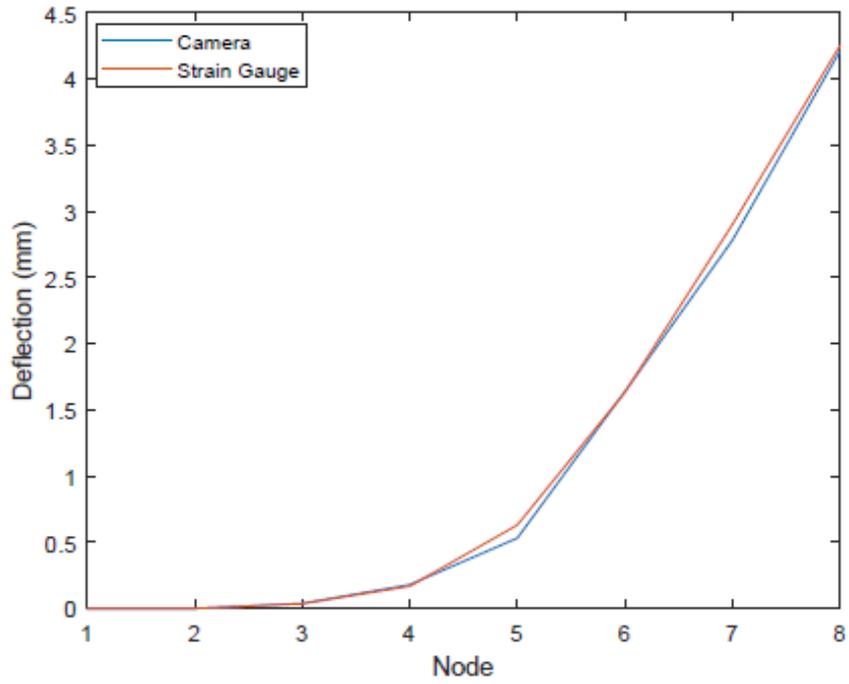


Figure 5.9: Deflection vs. node no., AOA = 10 deg, Roll angle = 10 deg.

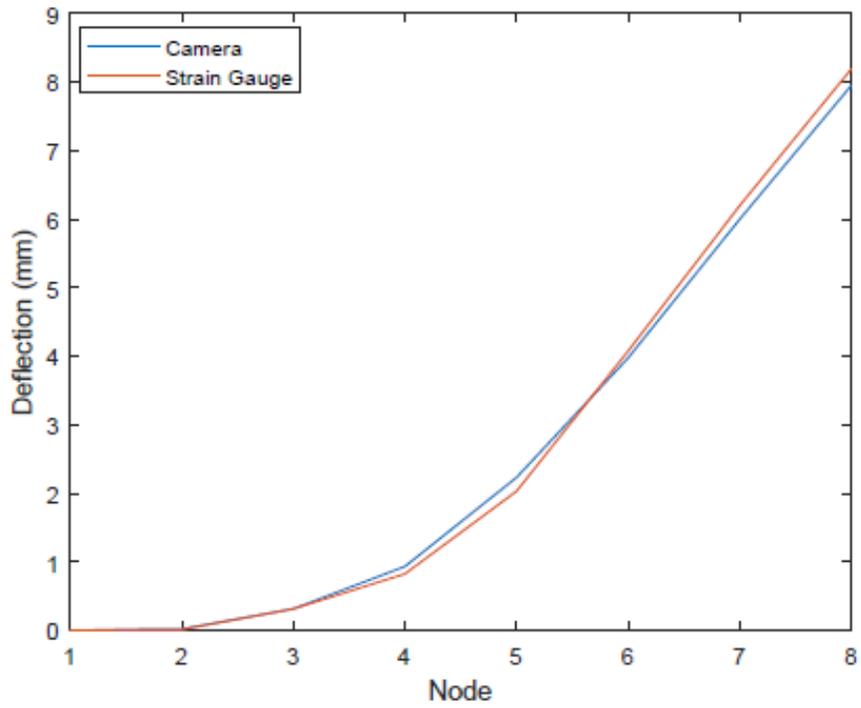


Figure 5.10: Deflection vs. node no. at AOA = 20 deg Roll angle = 0 deg.

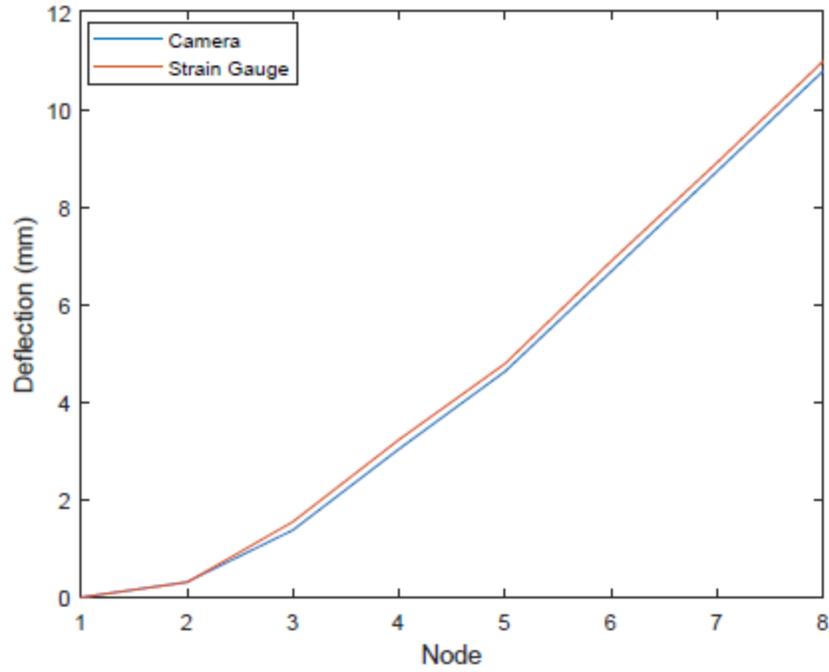


Figure 5.11: Deflection vs. node no. at AOA = 30 deg, Roll angle = 0 deg.

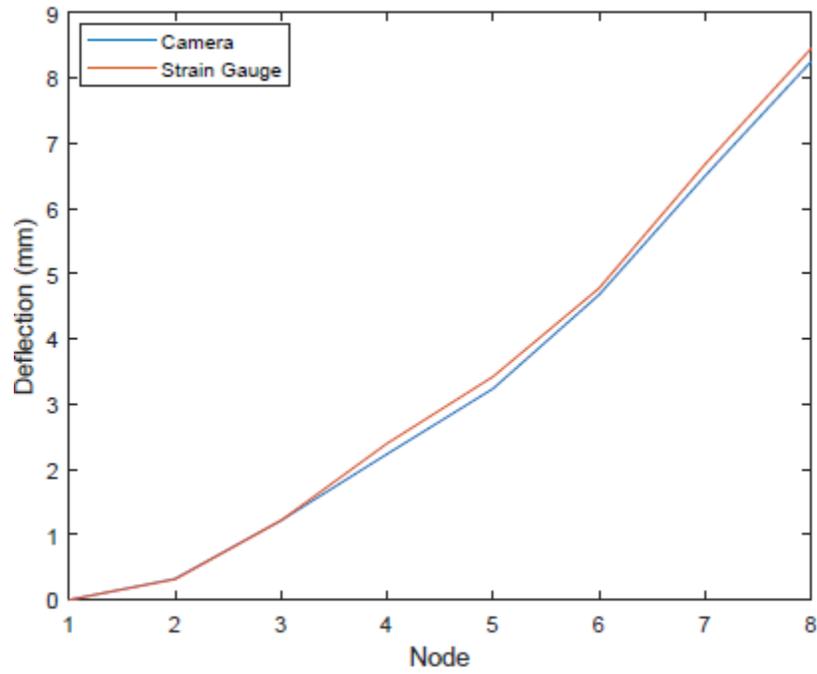


Figure 5.12: Deflection vs. node no. at AOA = 30 deg, Roll angle = 10 deg.

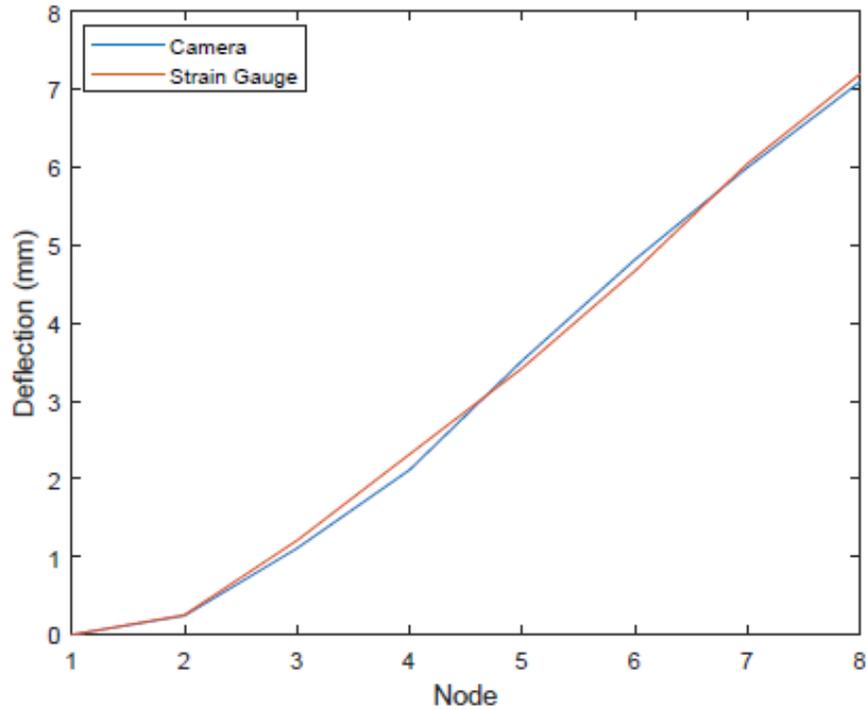


Figure 5.13: Deflection vs. node no. at AOA = 30 deg, Roll angle = 20 deg.

Figures 5.14, 5.15, and 5.16 show the deflection measurements by the camera and the strain gauge. It is noticed that the deflection was increased when the speed increased at an AOA of 0 to 30 degrees, while the maximum deflection occurs at a speed of 35 km/h and an AOA of 30 degrees. The relationship between maximum Deflection and AOA at different speeds is shown in Figure 5.17. The experiment has four phases depending on time, with the first phase measuring the deflection for 60 seconds with the wind speed only. The second phase starts from 60 seconds to 120 seconds. Finally, the deflection is measured when the roll angle is increased to 10 degrees.

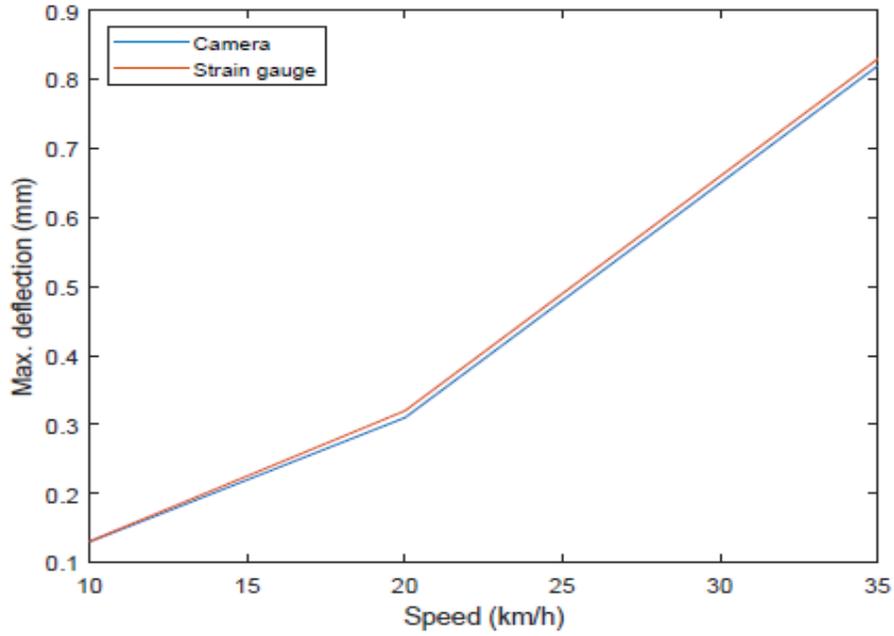


Figure 5.14: Max. deflection vs. different speed at AOA = 0 deg.

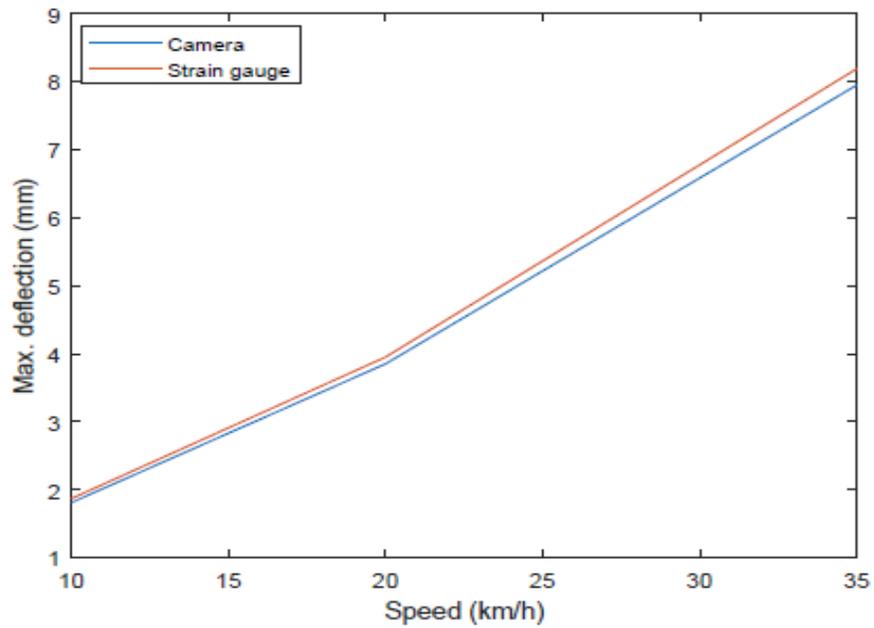


Figure 5.15: Max. deflection vs. different speed at AOA = 20 deg.

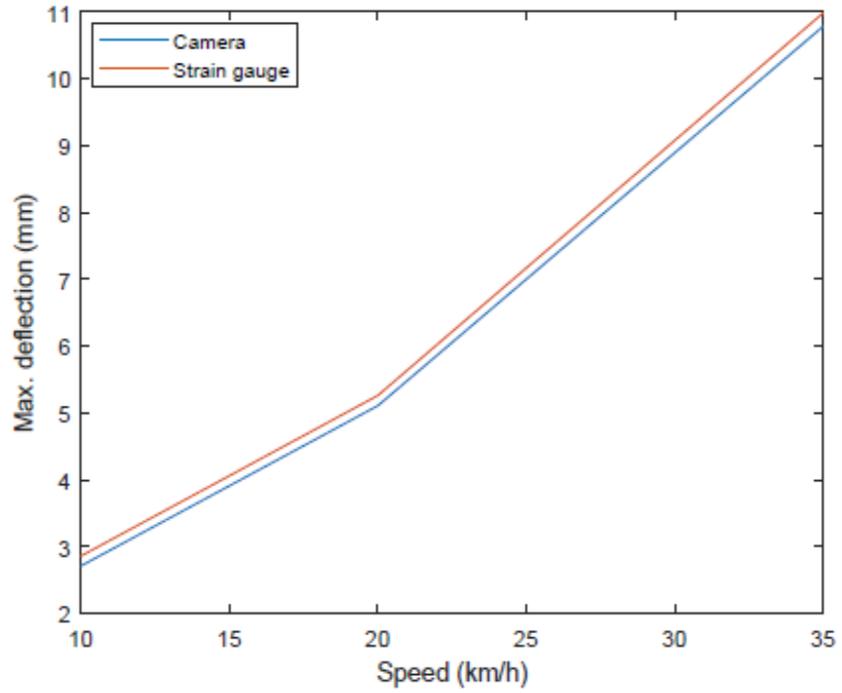


Figure 5.16: Max. deflection vs. different speed at AOA = 30 deg.

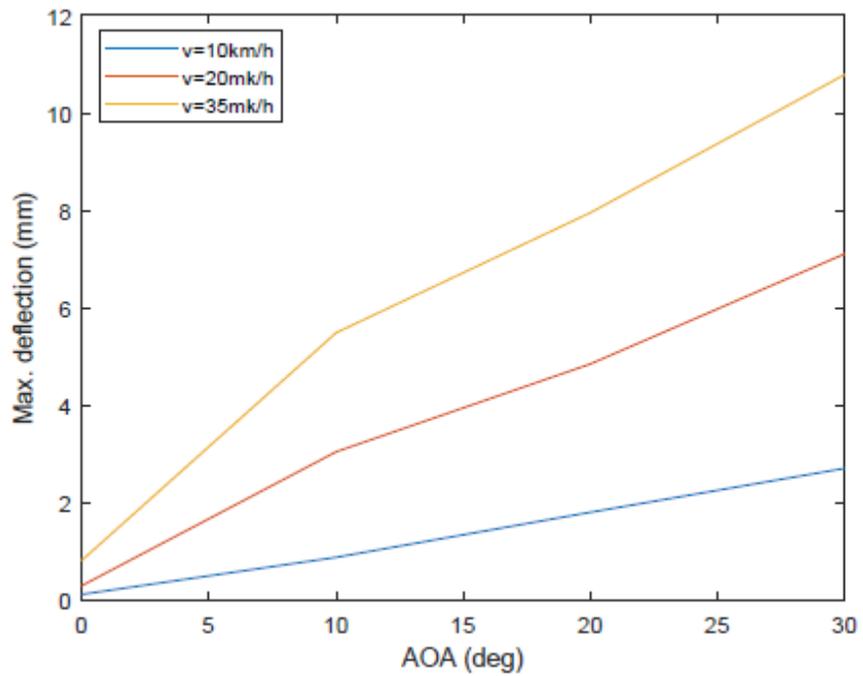
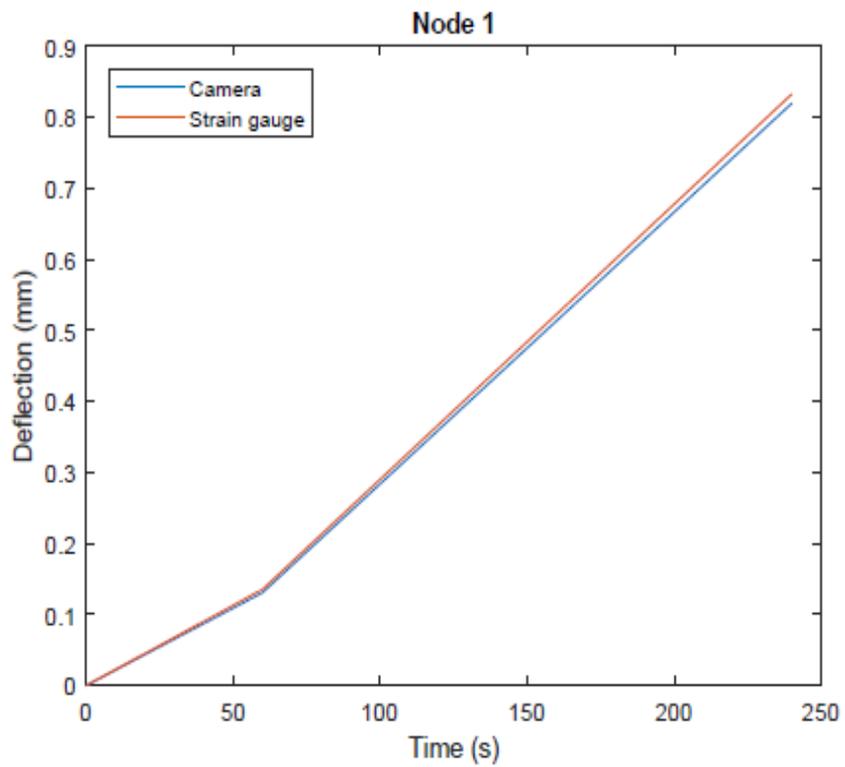
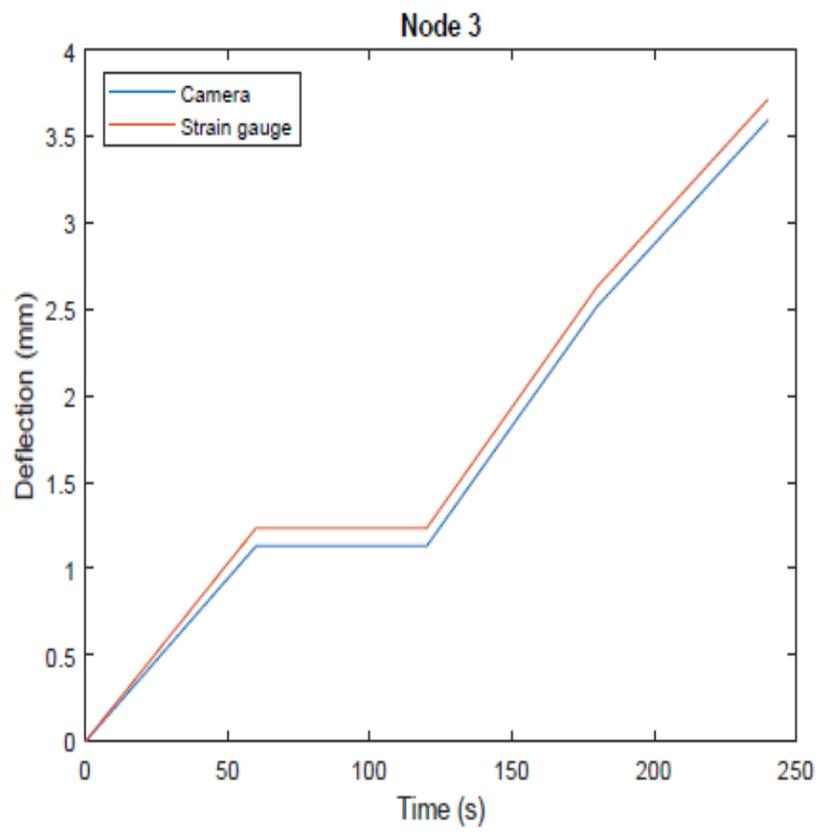
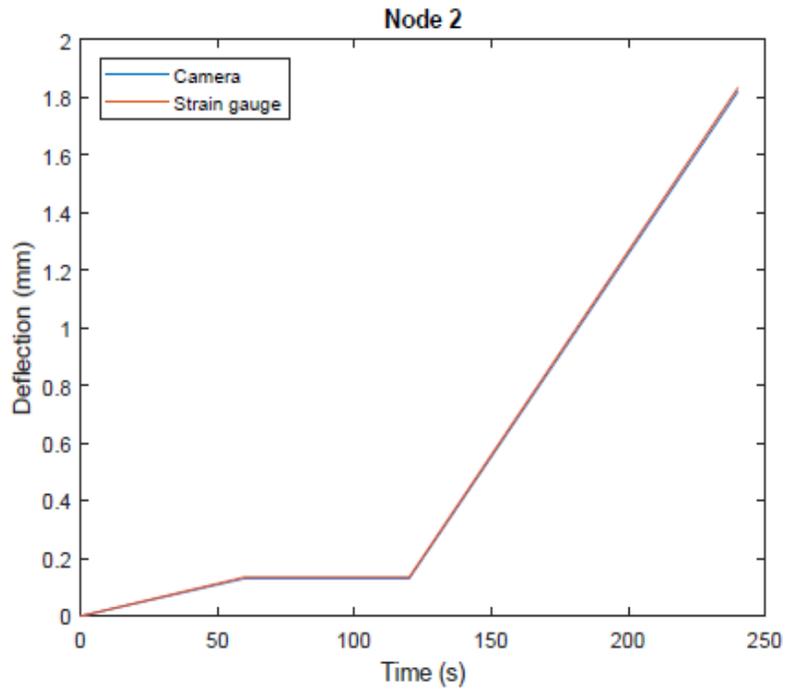
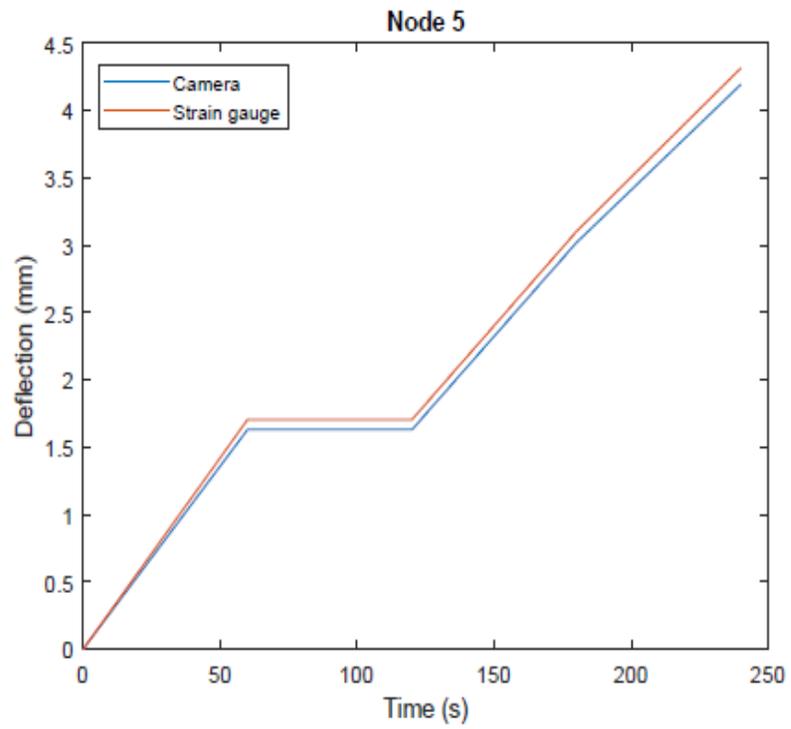
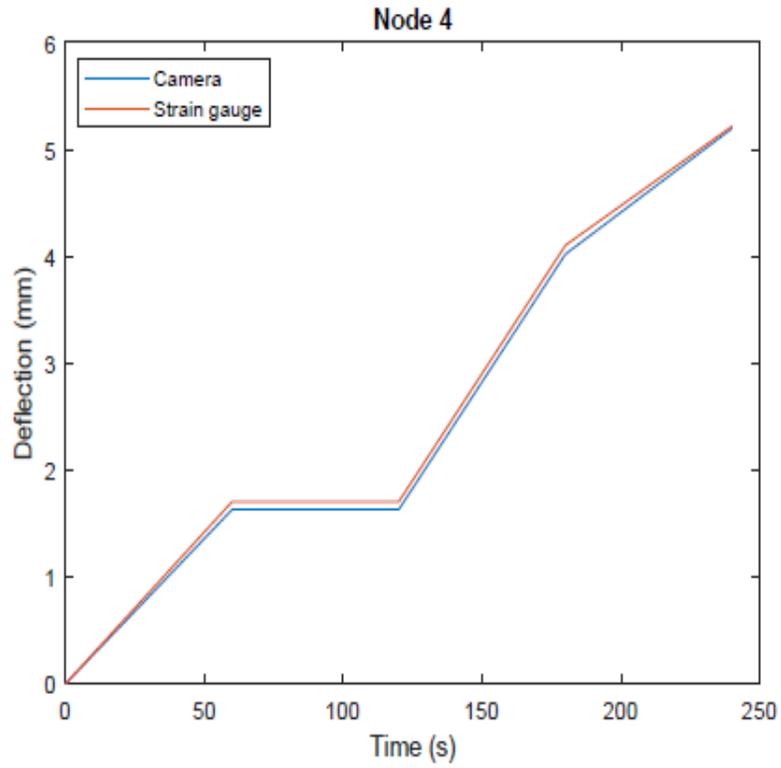


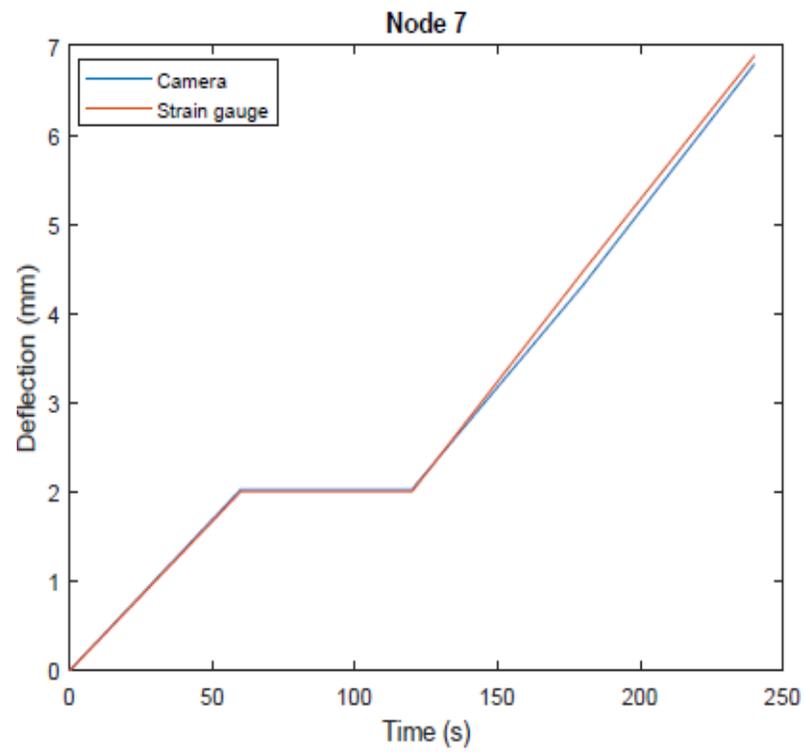
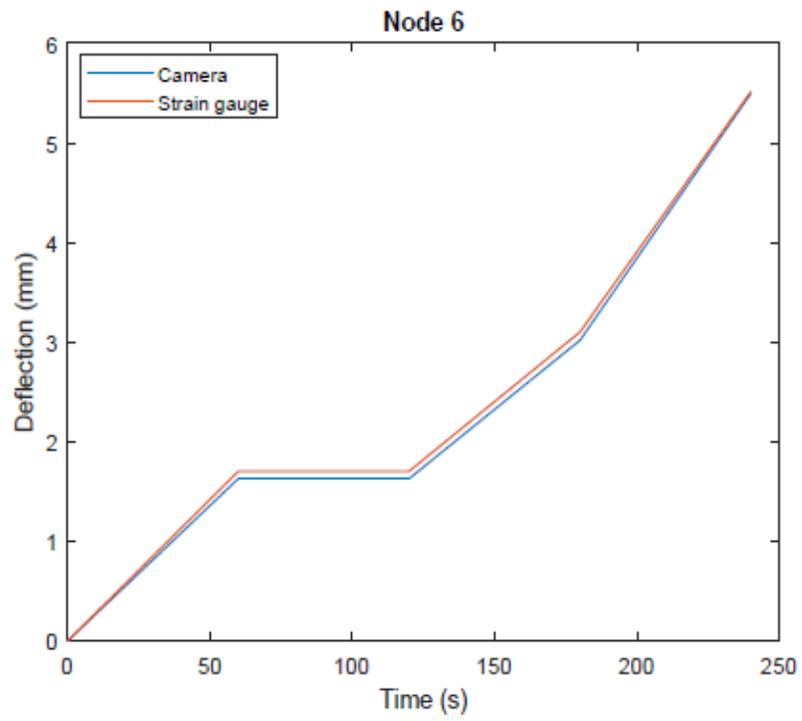
Figure 5.17: Max. deflection vs. AOA for the fused value.

The third phase starts from 120 seconds to 180 seconds, the deflection is determined when the speed is 35 km/s, and the roll angle is 10 degrees and 10 degrees AOA. The final phase starts from 180 seconds to 240 seconds, and the deflection is measured when AOA increases to 20 degrees, as shown with nodes 1 to 8 in Figure 5.18.









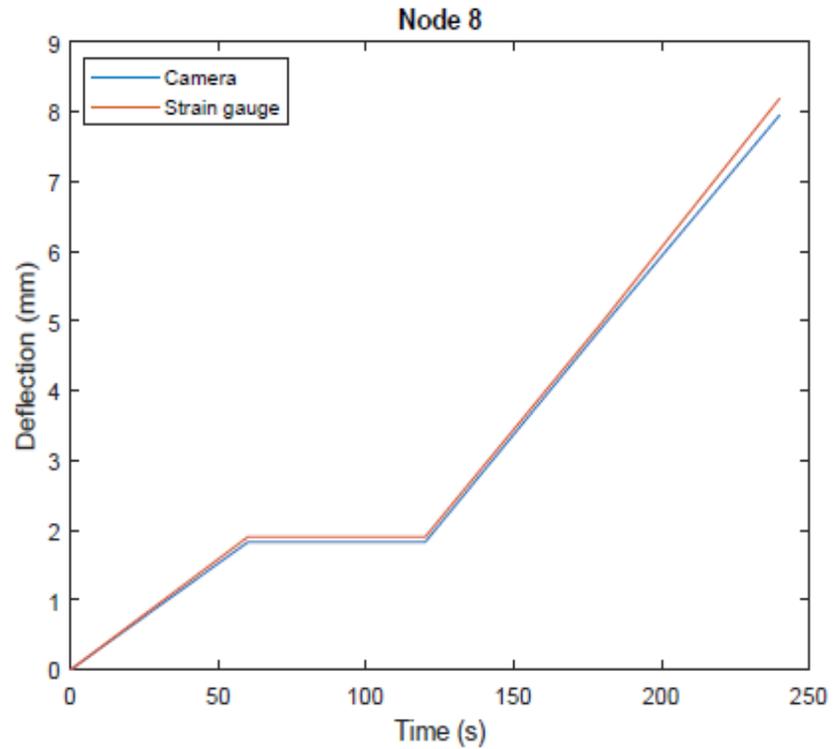


Figure 5.18: Tracking points.

5.6 Proposed Control System Design

The control system demonstrated in this thesis will use the information from the vision system and strain gauge to control the UAV. The preliminary work on control design is shown in the schematic diagram in Figure 5.19.

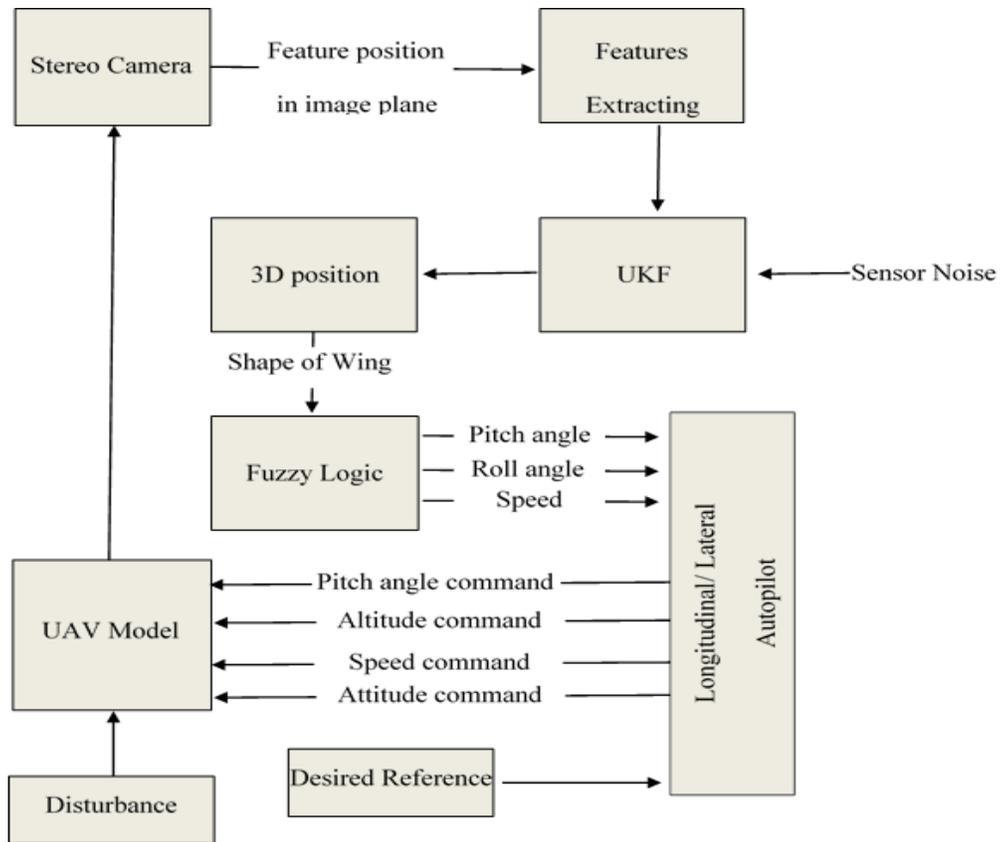


Figure 5.19: The Proposal Detection-Deflection Vision System (DDVS)

The equation of motion for a UAV is divided into two parts: longitudinal and lateral motion. The only motion control considered in this thesis is longitudinal since the model is a fixed-wing aircraft. This model can be obtained based on the linearization of the equations in steady-level flight. The longitudinal dynamics are the aircraft's response along the pitch axis and can be expressed in the following eq. (5.2) [98].

$$\dot{x} = Ax + Bu \quad (5.2)$$

$$y = Cx \quad (5.3)$$

where

$$A = \begin{bmatrix} -0.029 & -0.02 & -2645 & -32.7 \\ -0.029 & -0.021 & 212.5 & 2.78 \\ 0.002 & -0.1 & -2.6 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -6.248 & -9.684 \\ -44.376 & 0 \\ 13.757 & 0 \\ 0 & 0 \end{bmatrix}$$

and C is the identity 4×4 matrix. The values of A and B were calculated using the equations in Appendix B.

A unit step command is used to follow the desired value. In longitudinal control, the altitude and airspeed are controlled using the elevator and throttle combination. Figure 5.20 shows that the Fuzzy-PID controller was given a step of 200 ft/s as a reference for the airspeed input. The controller should track that reference airspeed while other Fuzzy-PID controllers regulate the attitude, as shown in Figure 5.21.

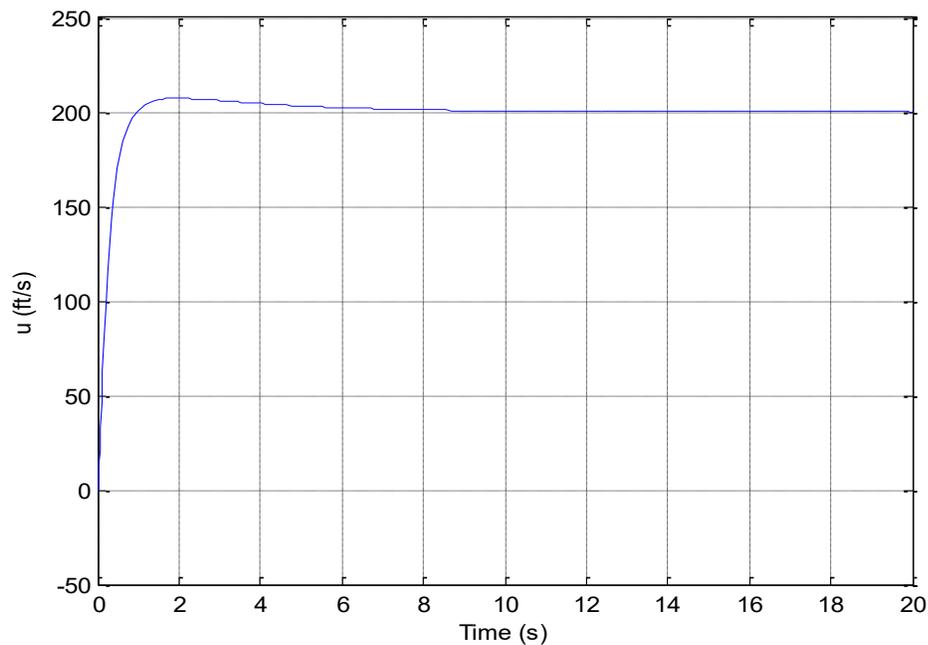


Figure 5.20: Airspeed with step throttle input.

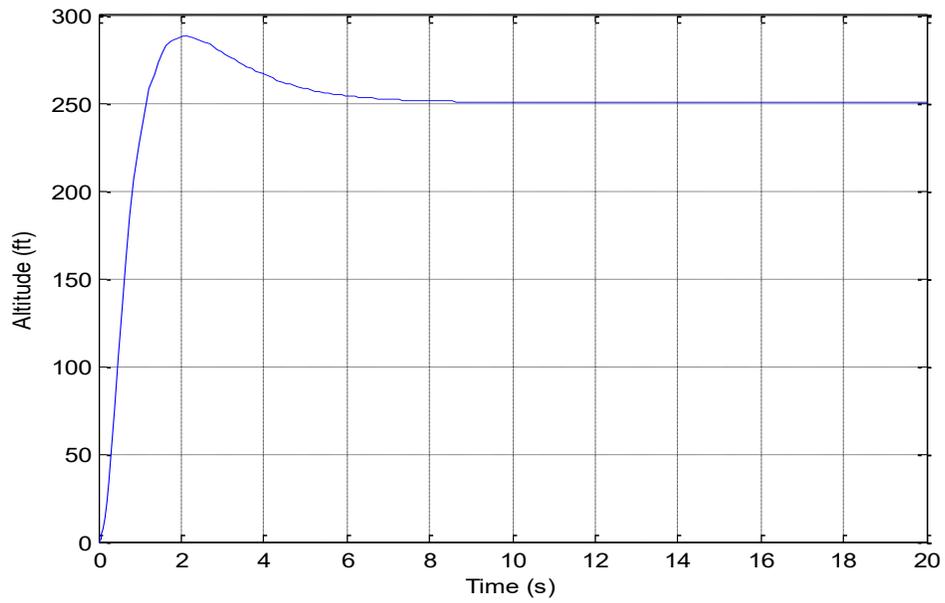


Figure 5.21: Altitude with step elevator input.

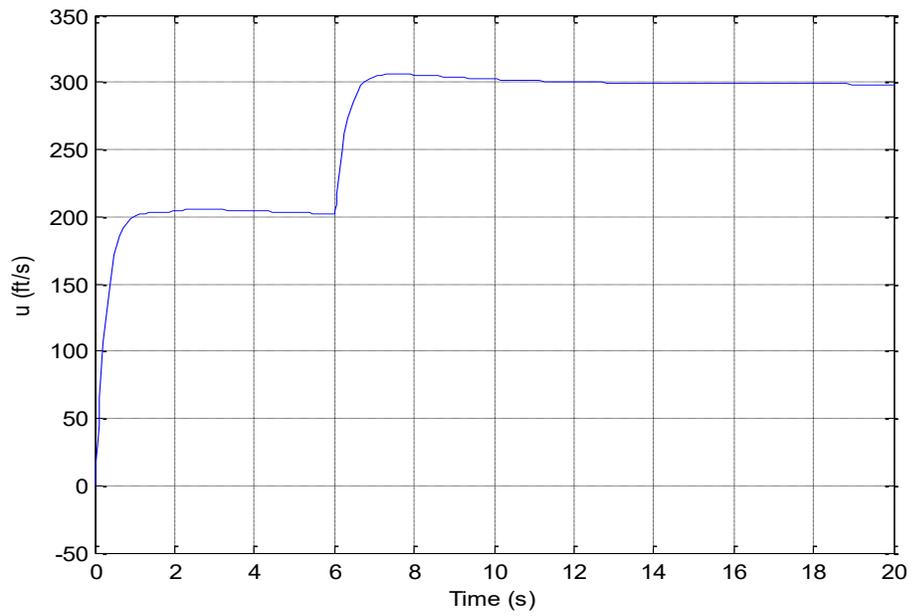


Figure 5.22: Airspeed increases by changing the throttle control.

To ensure that the controller works properly, the change in step inputs was studied in both throttle control and elevator input.

a) Throttle control increased while the elevator was fixed. Figure 5.22 shows that the airspeed increased from 200 ft/s to 300 ft/s at 6s while the altitude increased slowly to 210 ft, as shown in Figure 5.23.

b) Throttle control was fixed while the elevator increased. Figure 5.24 shows that the airspeed was not affected while the altitude was increased, as shown in Figure 5.25

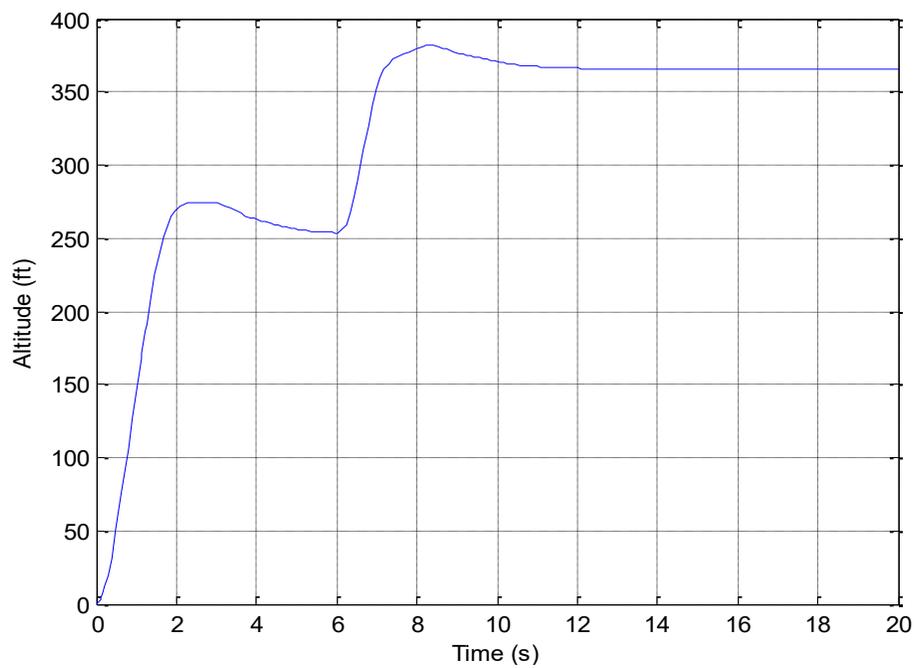


Figure 5.23: Altitude change when throttle change.

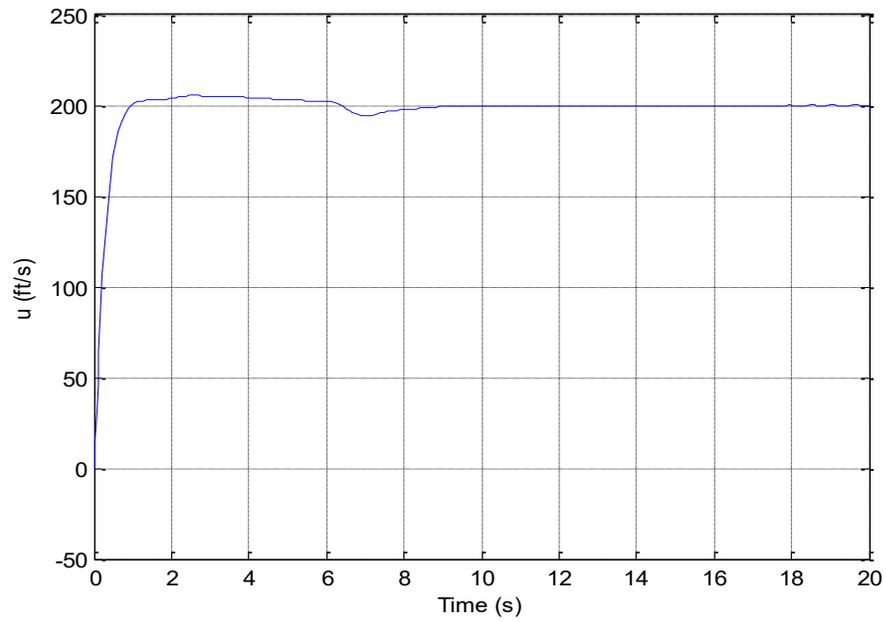


Figure 5.24: Airspeed fixed with elevator change.

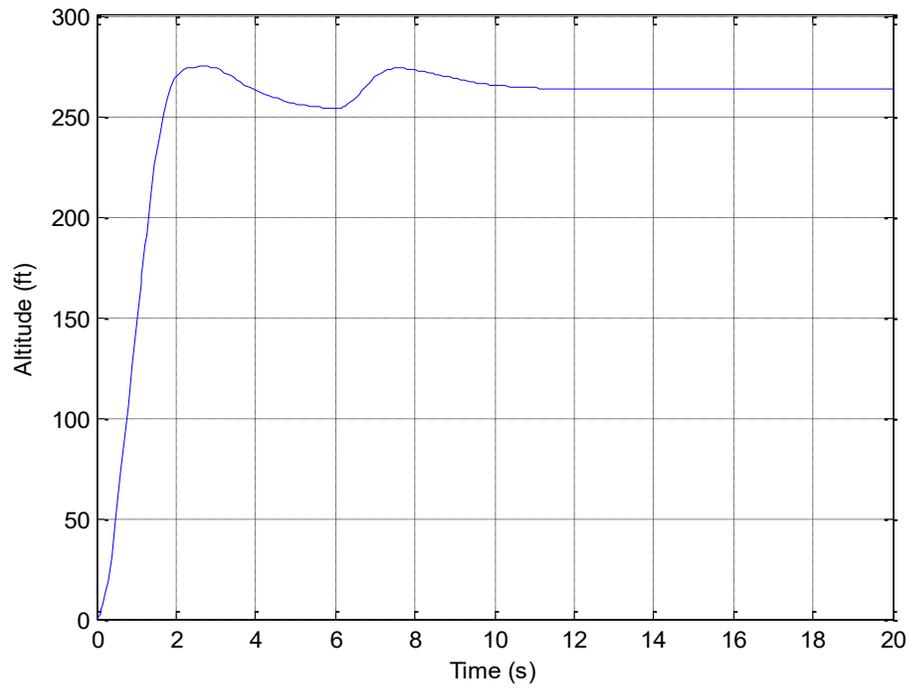


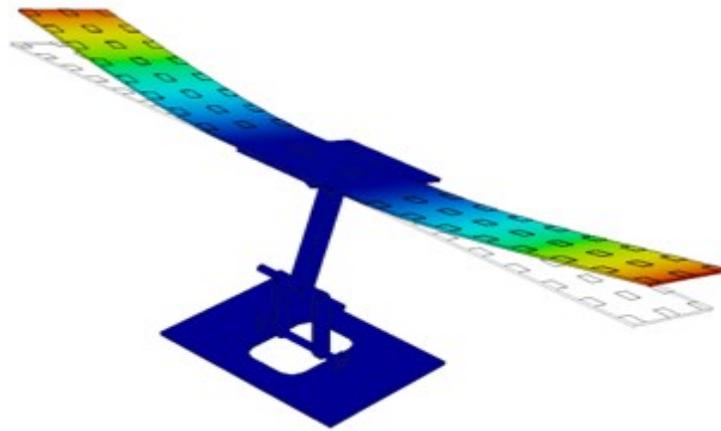
Figure 5.25: Altitude change when elevator change.

We compared the performance control and the output response of the control system relative to the airspeed and altitude of the step input signal to the LQR controller under the same condition. The result for the Fuzzy-PID controller was almost the same as shown in Appendix A1, A2, and A3, and there was no need for combining both controllers. It is noticed that the proposed Fuzzy-PID controller performance and response are excellent, and it indicates settling time, faster rising time, and overshoot to have been improved. The control algorithm will be designed and experimentally verified for the Cessna 172 aircraft, and the control system will be based on Fuzzy-PID.

Chapter 6

6 Result Verification

The information from the camera and strain gauge measurement compared with numerical value was used to calculate the wind load on the fixed-wing for the Autodesk Fusion 360 model. Then developed CAD assembly model in Autodesk Fusion 360, which was converted into a .step file in 12 different cases, i.e., for the angle of attack (AOA) 10, 20, 30 degrees, and for each AOA, there are four different rolling angles 0, 10, 20, 30 degrees. These files were input into SolidWorks for flow analysis and to obtain the aerodynamic forces. The required forces were calculated from the flow analysis and then inputted these data to Autodesk Fusion 360 to calculate the constraints, joints, materials, and forces by generating the mesh. These allowed solving the static analysis model in Fusion 360 and the deflection at each node, strain, stress, and many more aerodynamic parameters. During the inflow analysis, the assumption was made that the tunnel walls (computational domain) are adiabatic and with zero roughness. As a result, the inlet velocity of the wind in the wind tunnel is in the Z direction, as shown in Figure 6.2. The fixed-wing deflection is shown in Figure 6.1, and the result from the camera, strain gauge, and software measurement at different AOA and roll angles are presented in Table 6.1.



[mm] 0  10.7849

Figure 6.1: Fixed-Wing Maximum deflection at AOA = 30, Roll angle = 0 and Wind Speed 35 km/h.

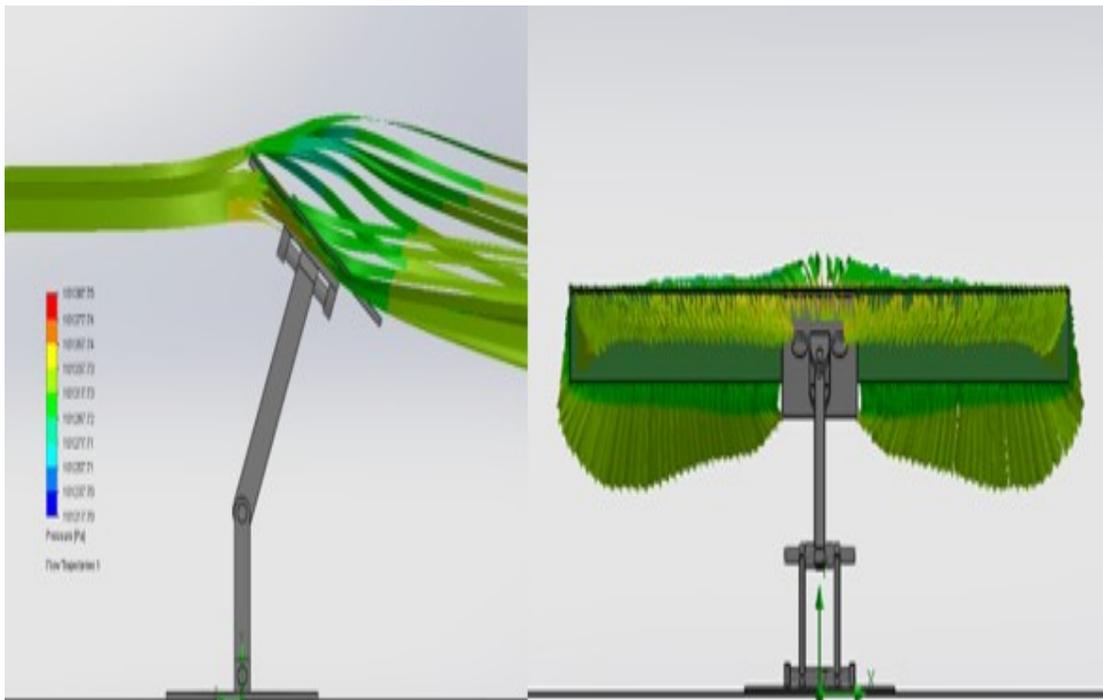


Figure 6.2: Side and front view of wing setup during flow analysis in the computational domain.

Table 6.1: Properties of the fixed-wing.

Parameter (unit)	Symbol	Numerical Values
Length of the wing (m)	L_w	1.00
Width of the wing (m)	W_w	0.15
Thickness of the wing (m)	T_w	0.003
Young's modulus (GPa)	E	3.2
Mass of wing (kg)	M	1.25
Density (g/cm^3)	ρ	1.18
Damping constant	β	0.0001

Table 6.2. Comparison of Deflection taken from camera/strain gauge and simulation.

Wind Velocity (km/h)	Angles		Simulation	Fused Data	Error (%)
	Angle of attack	Rolling angle	Deflection (mm)	Deflection (mm)	
35	0	0	0.8202	0.8201	0.0100
35	10	0	5.4660	5.4941	0.5115
35	10	10	4.1810	4.2121	0.7384
35	10	20	3.6971	3.6921	0.1327
35	10	30	2.9553	3.0037	1.6213
35	20	0	8.1422	7.9549	2.3520
35	20	10	6.5320	6.7004	2.5133
35	20	20	5.8001	5.9537	2.5816
35	20	30	4.4661	4.5401	1.6321
35	30	0	10.622	10.7849	1.5290
35	30	10	8.1933	8.2549	0.7499
35	30	20	7.0072	7.1003	1.3140
35	30	30	5.2091	5.3027	1.7670

The Standard Deviation value of 2.5717 for the strain gauge measurement was obtained using equation (3.32) in section 3.3 above.

Table 6.3: Standard deviation calculation for strain gauge measurement

x	$(x - \bar{x})^2$
0.8201	23.5965
5.4932	0.0340
4.2122	2.1474
3.6901	3.9502
3.0028	7.1546
7.9549	5.1861
6.7004	1.0461
5.9524	0.0755
4.5401	1.2939
10.7838	26.0733
8.2549	6.6423
7.1003	2.0241
5.3031	0.1403

The set of numbers $x = 13$

The mean values $\bar{x} = 5.6776$

The Standard Deviation

$$\sqrt{\frac{\sum(x-\bar{x})^2}{n-1}} = \sqrt{\frac{79.3643}{13-1}} = 2.5717$$

The Population Standard Deviation

$$\sqrt{\frac{\sum(x-\bar{x})^2}{n}} = \sqrt{\frac{79.3643}{13}} = 2.4708$$

6.1 Sensor Fusion

Sensor fusion is crucial for the localization, navigation, and control of an unmanned aerial vehicle. Several sensors have been used in trying to get perfect control for UAVs. Currently, fusing different sensors is now used in the autonomous robot. This integration helps to correct an error in the system and helps to improve the system performance. The camera and strain gauge information will be more efficient in deflection measurement, and it is cheap and easy to calibrate. To achieve an excellent result using camera and strain gauges to calculate deflection, we considered fusing both sensors using Kalman Filter (K.F.), Extended Kalman Filter (EKF), and Unscented Kalman Filter (UKF). K.F. is an algorithm that only deals with a linear dynamic system with white Gaussian noise. Extended Kalman Filter (EKF) has been previously used in sensor fusion but still has errors with linearization techniques and has glitching issues with other noise. The decision to use UKF is to give better performance in dealing with nonlinear systems and deal with white and colour Gaussian noise [106].

UKF is an advanced filtering method used for sensor data fusion, and it is more accurate than the Extended Kalman Filter (EKF) in several applications. This sensor fusion technology is used to advance the performance of localization systems, which helps in the navigation system. Precise vision systems are needed for robots, which involve feature detection and target tracking [107]. For a robot to navigate an environment properly, three crucial factors must be considered: Localizing the environment or getting the exact location of features, mapping the environment or detecting identical features, and making a decision or matching the features based on the information from localization and mapping. The system proposed in this thesis is the full integration of measurements from the vision

system and strain gauge sensors to calculate the dynamic behaviour of an aircraft fixed-wing and then use this information for control and navigation. The state of the error will determine the formation of the UKF model; that is, the state variable of the UKF will result from the difference in camera and strain gauge measurement. The new Deflection-Detection-Vision-System (DDVS) and strain gauges for UAV technology are based on 3D Digital Image acquisition fused with strain gauge information. It tracks a random pattern on the surface of components to provide an accurate and dynamic measurement of the shape, displacements, and strain map all over the object's surface using the fused data from the camera and strain gauge.

6.2 The Unscented Kalman Filter

The first step in the UKF is to evaluate a minimum set of sample points by the unscented transformation (UT.), called the sigma points around the mean. UT. is a tool used to measure statistics for a random variable undergoing a nonlinear transformation. As with the EKF method, UT. provides an alternative method to the analytic linearization approach, referred to as Jacobian matrices. [106],[107]

UKF addresses EKF's approximation issues. A Gaussian Random Variables (GRV) again represents the state distribution but is now defined using a limited set of carefully selected sample points. Such sample points fully capture the GRV's true mean and covariance. When propagated through the real nonlinear system, it reliably captures the posterior mean and covariance to the third-order Taylor series expansion for all nonlinearity cases. The unscented transformation (UT.) is a statistical tool used to measure a random variable that undergoes a nonlinear transformation [107].

Assume that a random variable x with n -dimensions has a mean \hat{x} and a covariance P_x and can be propagated by a nonlinear function. The following can be determined:

The sigma points X and the weighted points W [106].

$$X = [\hat{x} \quad \hat{x} \mp (\sqrt{(n + \lambda)P_x})_i]; \quad i = 1, 2, \dots, 2n \quad (6.1)$$

$$W_0^{(m)} = \frac{\lambda}{n + \lambda} \quad (6.2)$$

$$W_0^{(c)} = W_0^{(m)} + (1 - \alpha^2 + \beta) \quad (6.3)$$

$$W_i^{(c)} = W_i^{(m)} = \frac{1}{2(n + \lambda)} \quad i = 1, 2, \dots, 2n \quad (6.4)$$

where

$$\lambda = \alpha^2(n + k) - n \quad (6.5)$$

the scaling parameter; α is around sigma points \hat{x} and α is typically set to a small positive value (e.g. $1e - 4 \leq \alpha \leq 1$); k is a secondary scaling parameter that is generally set to 0, and β is used to incorporate prior knowledge of the distribution of \hat{x} . The $W_i^{(m)}$ and $W_i^{(c)}$ are the weight for mean and covariance, respectively [107].

The UKF algorithm can be summarized as follows [106]:

- 1- Calculate sigma points with mean and covariance weight using eq. (6.1) to (6.5).
- 2- Transfer each sigma point is passed through the nonlinear state prediction function f .

$$(\hat{X}_{k/k-1})_i = f((X)_i) \quad i = 0, 1, \dots, 2n. \quad (6.6)$$

- 3- Predicted the mean using weighted averages of the transformed sigma points is calculated by

$$\hat{x}_{k/k-1} = \sum_{i=0}^{2n} W_i^m (\hat{X}_{k/k-1})_i \quad (6.7)$$

- 4- Predicted covariance is calculated by $P_{k/k-1} = \sum_{i=0}^{2n} W_i^c [\hat{X}_{k/k-1} - \hat{x}_{k/k-1}] [\hat{X}_{k/k-1} - \hat{x}_{k/k-1}]^T + Q$

- 5- Each prediction points are calculated instead through an observation model:

$$(\hat{Z}_{k/k-1})_i = h((\hat{X}_{k/k-1})_i) \quad (6.9)$$

- 6- Predicted observation

$$\hat{z}_{k/k-1} = \sum_{i=0}^{2n} W_i^m (\hat{Z}_{k/k-1})_i \quad (6.10)$$

- 7- Innovation covariance

$$P_{vv} = \sum_{i=0}^{2n} W_i^c [\hat{Z}_{k/k-1} - z_{k/k-1}] [\hat{Z}_{k/k-1} - \hat{z}_{k/k-1}]^T + R \quad (6.11)$$

- 8- Cross covariance

$$P_{xz} = \sum_{i=0}^{2n} W_i^c [\hat{X}_{k/k-1} - \hat{x}_{k/k-1}] [\hat{Z}_{k/k-1} - \hat{z}_{k/k-1}]^T \quad (6.12)$$

- 9- Update

$$K_k = P_{xz} P_{vv}^{-1} \quad (6.13)$$

$$\hat{x}_k = \hat{x}_{k/k-1} + K_k (z_k - \hat{z}_{k/k-1}) \quad (6.14)$$

$$P_k = P_{k/k-1} - K_k P_{vv} K_k^T \quad (6.15)$$

6.3 Mathematical Modeling of The Fixed-Wing

This model shows a two-dimension state: deflection and theta of the structure at the time step

$k - 1$ as $x(k - 1)$, $\dot{x}(k - 1)$, and $\ddot{x}(k - 1)$, respectively. The displacement during the time step k is obtained from the displacement, velocity, theta, and a system noise at time step $k - 1$ as

$$x(k) = x(k - 1) + \dot{x}(k - 1)\Delta t + 0.5\ddot{x}(k - 1)(\Delta t)^2 + w_d(k - 1) \quad (6.16)$$

where $w_d(k - 1)$ is the noise added to the displacement, and Δt is the time steps interval [108].

$$\dot{x}(k) = \dot{x}(k - 1) + \ddot{x}(k - 1)\Delta t + w_v(k - 1) \quad (6.18)$$

Where $w_v(k - 1)$ is the added noise to the system, and the main system noise assuming zero-mean Gaussian white [108] are $w_v(k - 1)$ and $w_d(k - 1)$. The real deflection and measurement noise are

$$x_m(k) = x(k) + v(k) \quad (6.19)$$

The measured deflection related to $x_{sd}(k)$ is $x_m(k)$ and $v(k)$ is the measured noise from the measurement error [108]. The state vector is required to get the state-space model.

$$\mathbf{X}(k) = [x(k) \ \dot{x}(k)]^T \quad (6.20)$$

Where the T is the vector transpose, and the state-space model of the deflection measurement from camera and strain gauge data fusion is written as

$$\mathbf{X}(k) = \mathbf{A}\mathbf{X}(k-1) + \mathbf{B}\ddot{x}(k-1) + \mathbf{w}(k-1) \quad (6.21)$$

$$x_{sd}(k) = \mathbf{H}\mathbf{X}(k) + v(k) \quad (6.22)$$

The matrices \mathbf{A} , \mathbf{B} , \mathbf{H} , and $\mathbf{w}(k-1)$ are represented as

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} (\Delta t)^2/2 \\ \Delta t \end{bmatrix}, \quad \mathbf{H} = [1 \quad 0]$$

$$\mathbf{w}(k-1) = \begin{bmatrix} w_d(k-1) \\ w_v(k-1) \end{bmatrix} \quad (6.23)$$

The model measurement noise $v(k)$ and noise vector $\mathbf{w}(k-1)$ are both Gaussian white noise with a covariance matrix, and q and r are the noise variables [108].

$$\mathbf{Q} = q \begin{bmatrix} \frac{(\Delta t)^3}{3} & \frac{(\Delta t)^2}{2} \\ \frac{(\Delta t)^2}{2} & (\Delta t) \end{bmatrix}, \quad \mathbf{R} = \frac{r}{\Delta t} \quad (6.24)$$

The DDVS and strain gauge starts to measure the deflection of all points in the wing at speed, starting from 10 km/h to 35 km/h, AOA of 0 degrees, and roll angle of 0 degrees. Then the deflection is measured at AOA 10, 20, and 30 degrees with a roll angle of 0 degrees, as shown in Figures 6.3 to 6.10. The maximum deflection was measured for the vision system and simulation result using a wind speed of 35km/h, and AOA 30 degrees, as shown in Figure 6.11. The test was repeated for different wind speeds ranging from 10 km/h to 35 km/h for 180 seconds. The maximum deflection occurs at a velocity of 35 km/h, AOA 30 degrees, and roll angle of 0 degrees. The experiment shows that, as the roll angle

increases, the deflection decreases, and as AOA increases, with 0 roll angle, deflection increases, as shown in Figure 6.5.

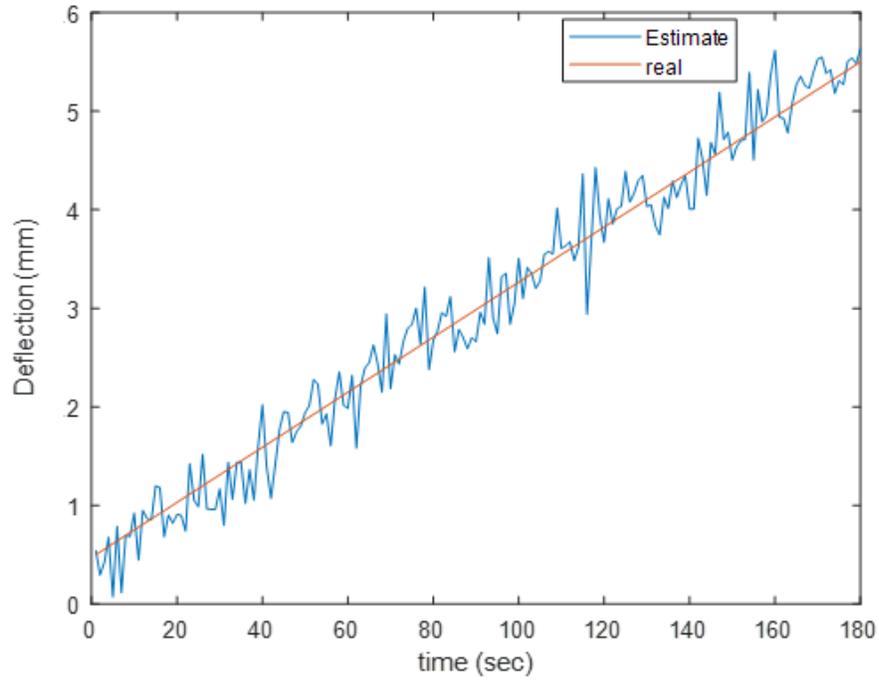


Figure 6.3: UKF fusion of camera and strain gauge at AOA 10 deg, Roll angle 0 deg.

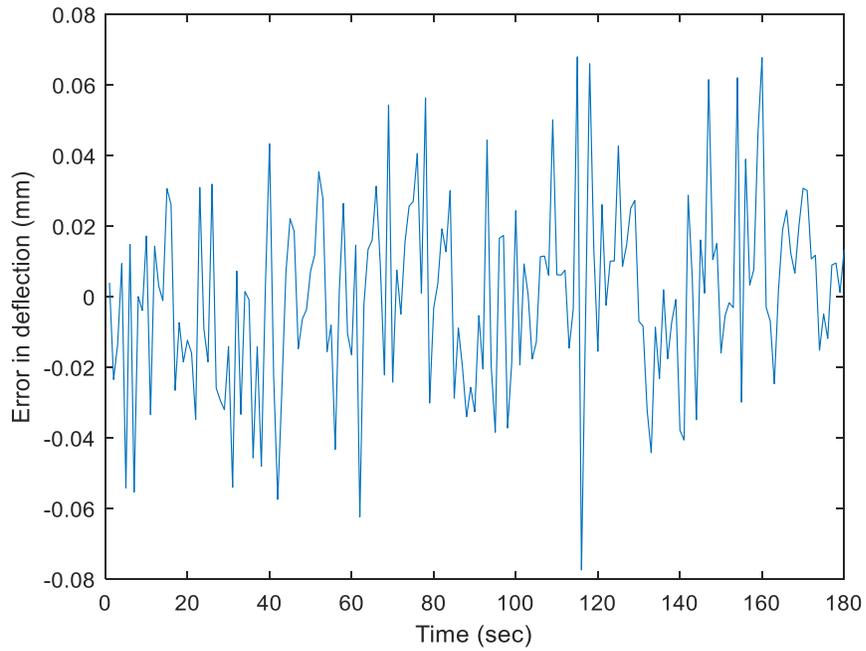


Figure 6.4: Measurement error, AOA 10 deg, Roll angle 0 deg.

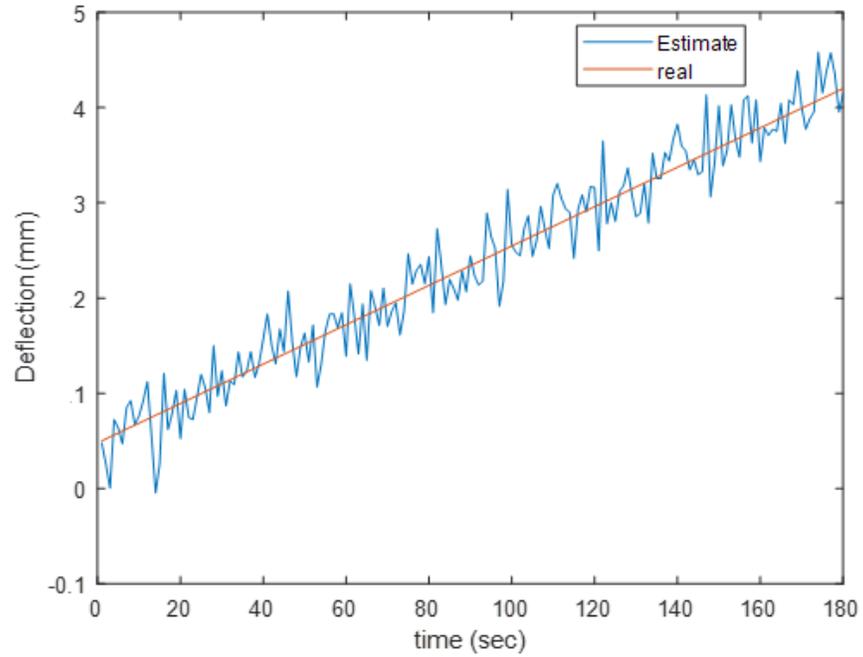


Figure 6.5: UKF fusion of camera and strain gauge at AOA 10 deg, Roll angle 10 deg.

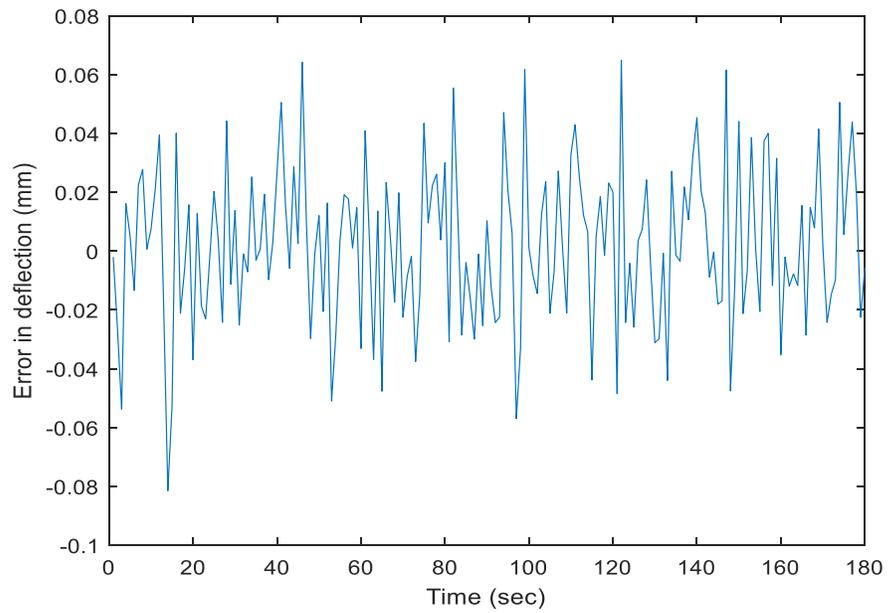


Figure 6.6: Measurement error, AOA 10 deg, Roll angle 10 deg.

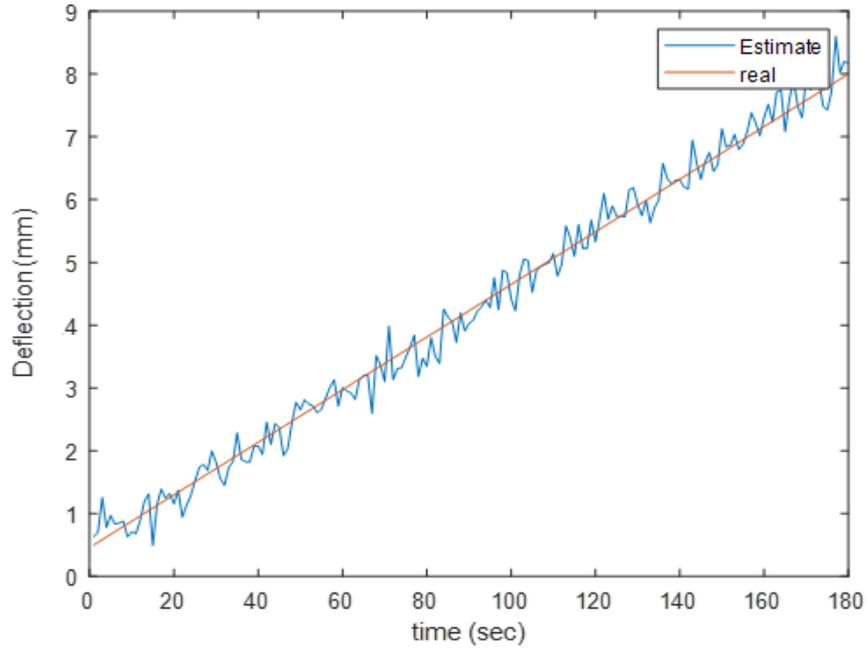


Figure 6.7: UKF fusion of camera and strain gauge at AOA 20 deg roll angle 0 deg.

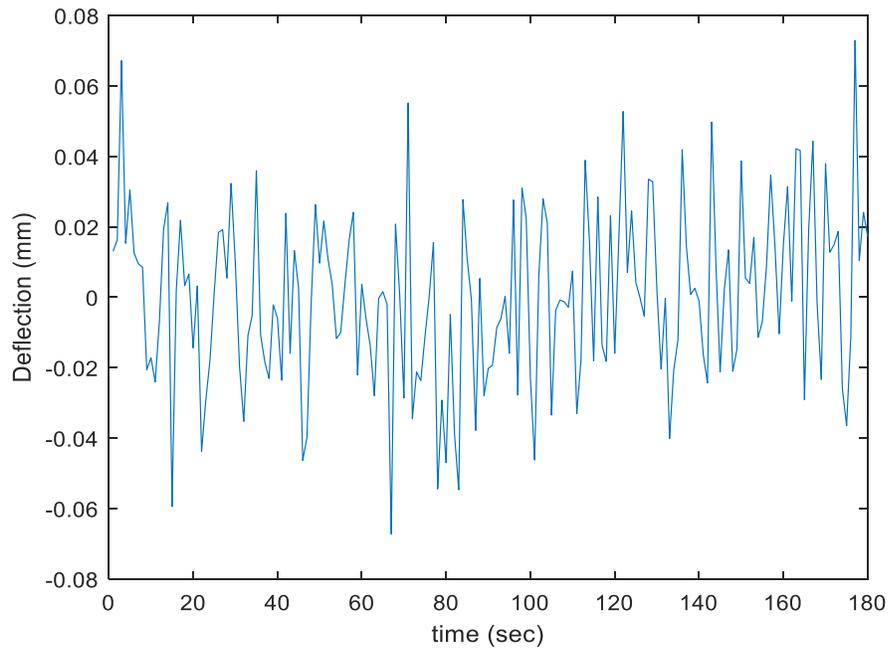


Figure 6.8: Measurement error, AOA 20 deg, Roll angle 0 deg.

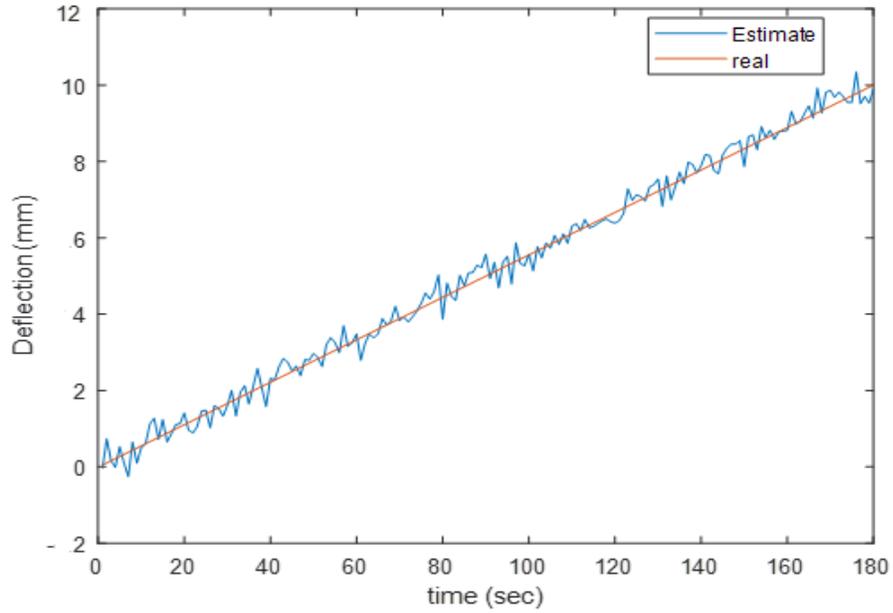


Figure 6.9: UKF fusion of camera and strain gauge at AOA 30 deg roll angle 0 deg.

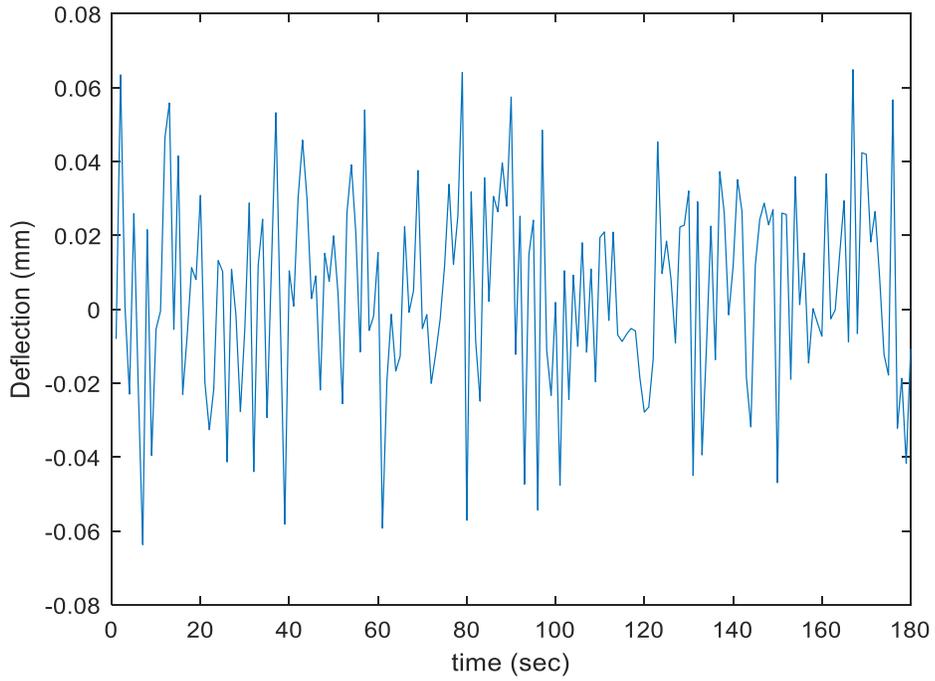


Figure 6.10: Measurement error, AOA 30 deg, Roll angle 0 deg.

The deflection increases with an increase in AOA. Also, the deflection at each node measured by the camera is very close to the strain gauge and simulation measurement at an AOA of 30 degrees, as shown in Figure 6.11.

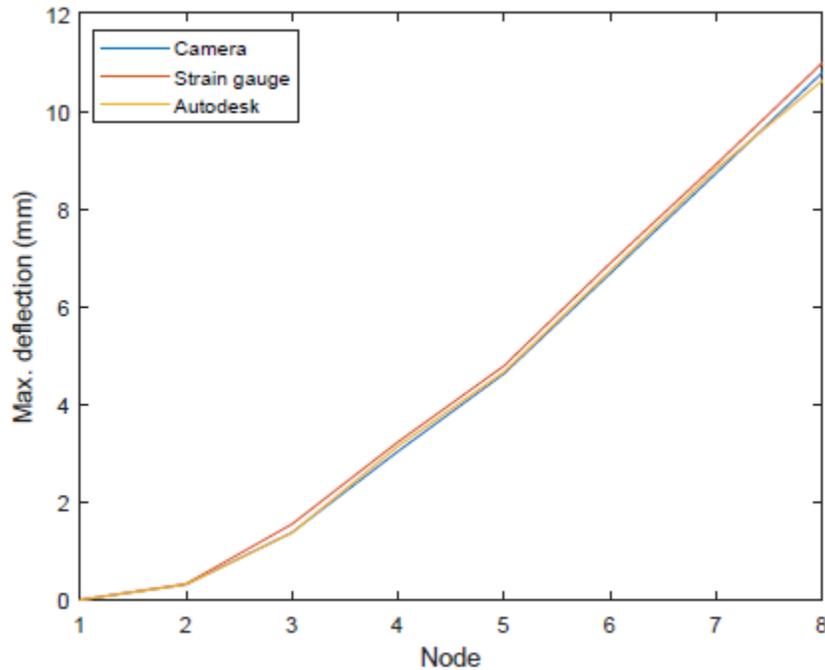


Figure 6.11: Camera vs. Simulation and Strain Gauge deflection at AOA= 30 deg Roll angle = 0 deg.

The wing controls the airplane, and during the flying condition, large numbers of aerodynamic forces act on it. The wing deflection plays a very crucial role because high deflection will affect the navigation of the UAV. The vision system is one that can measure the deflection of the wing by using a stereo camera calibration. The system compares the ideal results using the idealization process, by which the same conditions in the analyzing software are the same as in the real-time experiment.

6.4 The Aerodynamic Results

For the inflow analysis, there is an assumption that the tunnel (computational domain) walls are adiabatic and with zero roughness. The input of the inlet velocity of the wind in the wind tunnel is in the Z direction of the fixed wind setup. After that, set the goals to find the various required values at the fixed-wing surface (data are required at the wing surface, especially at the lower surface area). The flow distribution of the wind over the surface area of the fixed-wing is shown in Figure 22. Considering the fixed-wing UAV is essential to calculate the aerodynamic force required for the fixed-wing to lift the model's mass. These values were used in the live camera-vision measurement system, and the original dimension of the wing used for that calculation is;

Width = 0.168 m

Length = 1.160 m

Square size = 32 x 32 mm

The pressure difference between the upper and lower surface of the wing is needed to obtain the force is required to calculate the lift force-velocity because the lift is a result of pressure. Pitot tubes are assembled at the upper and lower surfaces to compare the calculated pressure, which was used to achieve the lift force-velocity, as shown in Figure 6.12. The lift coefficient can be calculated using the equation (6.26) if that lift force value is known. The values of thrust force and sine product of the weight of the model are required to calculate the drag force. The value of the drag force is then used to calculate the drag coefficient using equation (6.29).

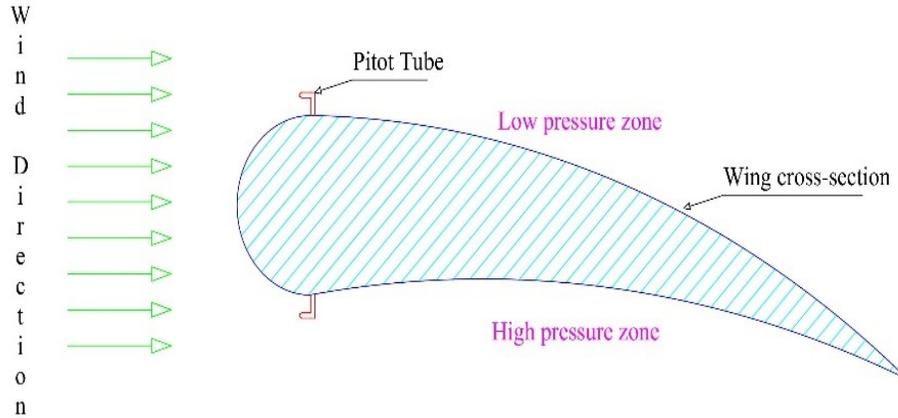


Figure 6.12: Wing setup for lift force calculation.

Equations used for analysis.

- Lift force (F_L) = $\frac{1}{2} * A\rho * (V_{upper}^2 - V_{lower}^2)$ (6.25)

- Lift co-efficient (C_L) = $\frac{2 * F_L}{\rho * A_L * V^2}$ (6.26)

- Drag force during climb condition (F_D) = Thrust - [W * sin (AOA)] (6.27)

- Drag force during descend condition (F_D) = Thrust + [W * sin (AOA)] (6.28)

- Drag coefficient (C_D) = $\frac{2 * F_D}{\rho * A_D * V^2}$ (6.29)

The velocity square difference will give the lift force magnitude from equation (6.25).

Equation (6.26) is used to calculate the lift force coefficient, and similarly, equation (6.29)

also gives the drag coefficient value. For drag force calculation, there are mainly three

cases first is during take-off (climb), which uses the sine product of the weight of the UAV

as subtraction. The second case during landing (decent) that sine product will be an addition

in thrust as equation (6.28). The third case is during the stable condition in the air when the thrust force is equal to the drag force.

6.4.1 The input parameters

When the UAV is climbing, the input character is classified as 1 or write 0 =1.

Weight of Wing (N) = 12.2625

Angle of attack (Degree) = 30

Relative velocity of Airplane (m/s) = 0.3

Exit velocity of air from engine (m/s) = 2.096

Density of air (kg/m^3) = 1.225

Area of the propulsive device (Engine) (m^2) = 1.44

Velocity difference at wing (m/s) = 4.612

6.4.2 The out parameters

First point of Y-axis = 1215.039673

Last point of Y-axis = 55.032108

First point of X-axis = 173.830368

Last point of X-axis = 41.390911

Actual Width (m) = 0.161679

Length (m) = 1.160008

Surface Area (m^2) = 0.187549

Mass flow rate enter the propulsive device = 0.529200 Kg/s

Mass flow rate exit the propulsive device = 3.697344 Kg/s

Thrust force = 7.590873 N

minimum Lift force require = 10.619637 N

Actual Lift force = 2.824850

Drag force during climb condition = 0.557392

Lift co-efficient = 0.260174

Drag co-efficient = 0.051337

The final output result is summarized in Table 6.3. The fixed-wing setup during flow analysis in the computational domain is shown in Figure 6.13. Figures 6.14, 6.15, and 6.16 show the different drag and lift forces generated by the DDVS when the AOA is 10 degrees and roll angles are 0, 10, 20, and 30. With increasing roll angle, the lift and drag force values decrease due to changes in the projected area of the fixed-wing. The drag force values are lesser than lift force values in all the cases, which shows the validity of the results. As the AOA increases, the values of the drag and lift forces become higher.

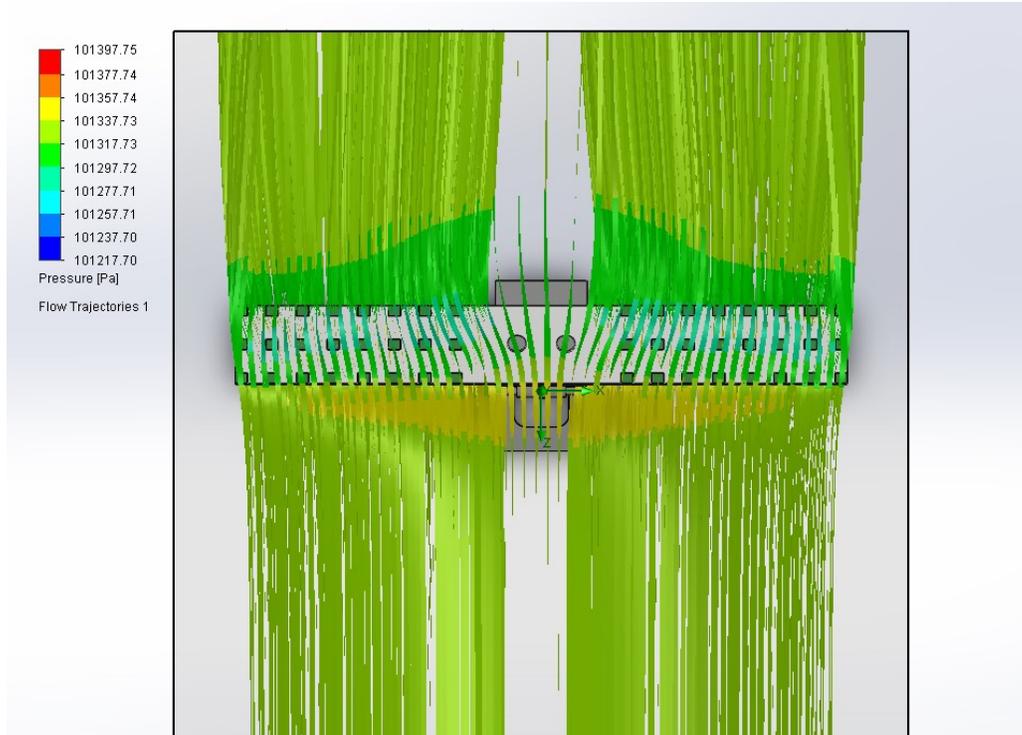


Figure 6.13: Top view of wing setup during flow analysis in the computational domain.

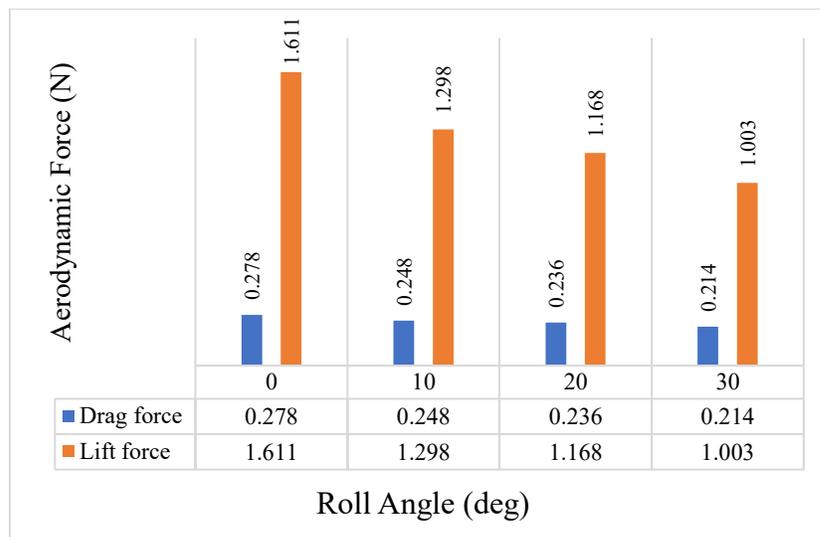


Figure 6.14: Roll Angle vs. Aerodynamic forces for AOA 10 deg.

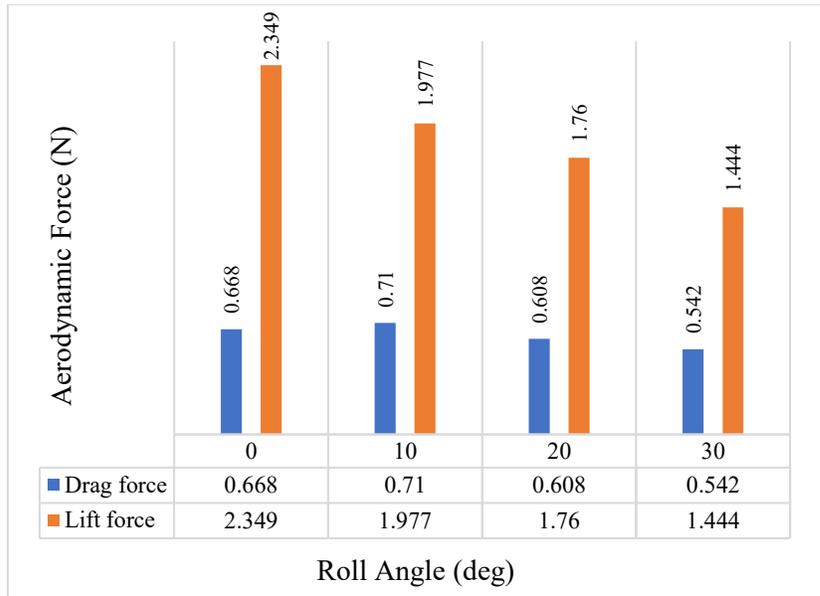


Figure 6.15: Roll angle vs. Aerodynamic forces for AOA 20 deg.

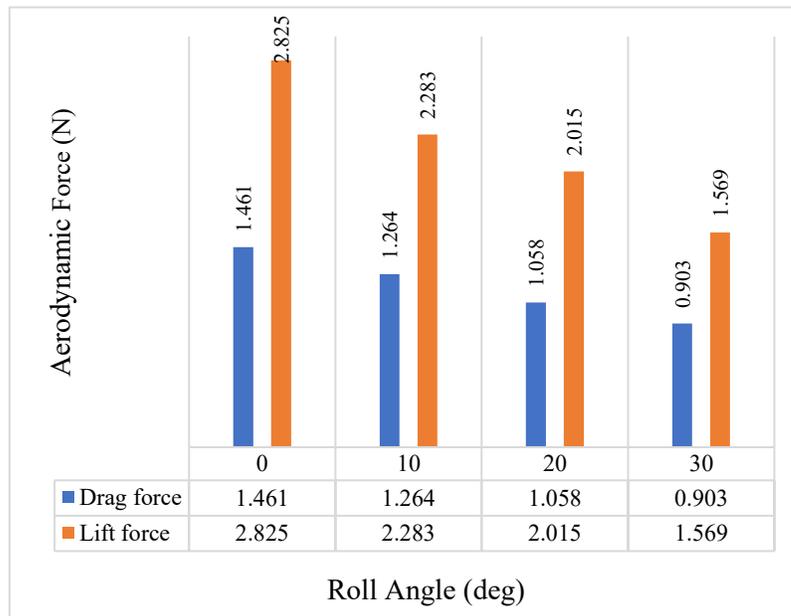


Figure 6.16: Roll angle vs. Aerodynamic forces for the AOA 30 deg.

Figures 6.17, 6.18, and 6.19 show the different deflection for roll angles 0, 10, 20, and 30 for the constant angle of attack 10 degrees, 20 degrees, and 30 degrees, respectively. With increasing roll angle, the deflection values decrease due to changes in forces, and the deflection increases as the AOA increases. Moreover, the simulation and vision system values are compared in that graph, and the differences between both are significantly less.

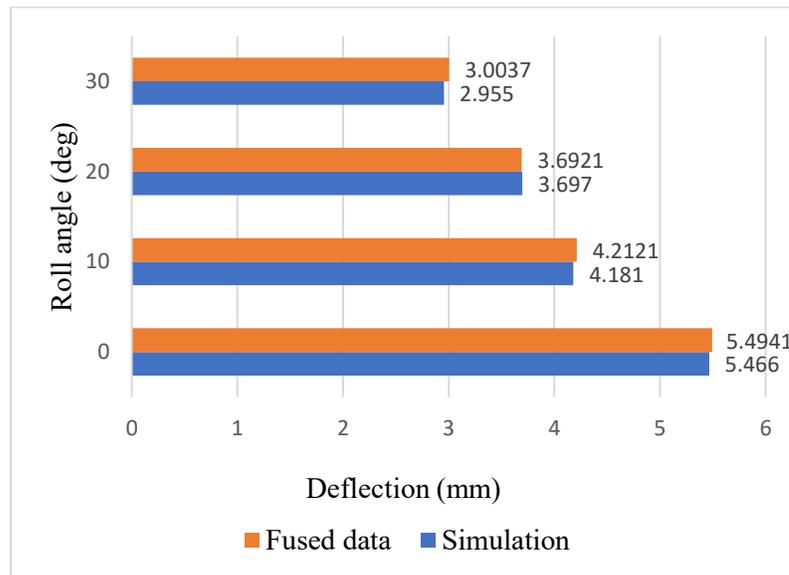


Figure 6.17: Roll angle vs. Deflection for AOA 10 deg.

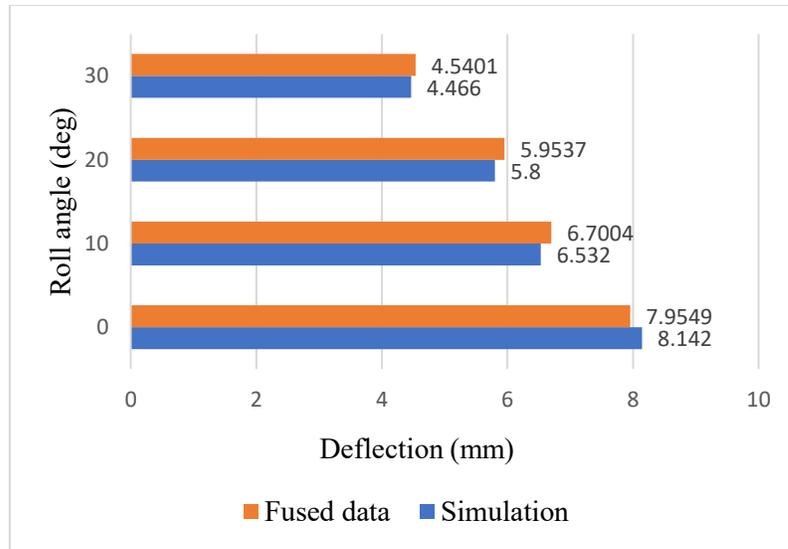


Figure 6.18: Roll angle vs. Deflection for AOA 20 deg.

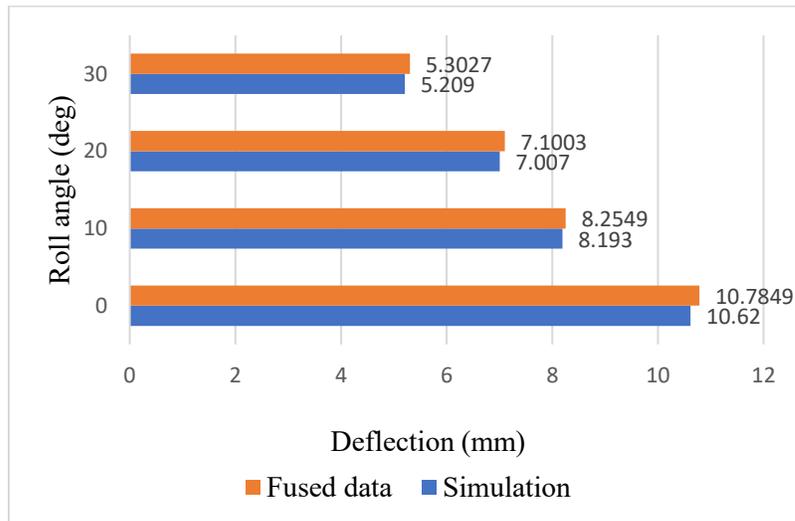


Figure 6.19: Roll angle vs. Deflection for AOA 30 deg.

As the AOA increases, the area will increase due to the increases in induce force, consequently increasing deflection. The experiment was conducted using AOA 10 degrees, 20 degrees, and 30 degrees, respectively. Moreover, the software analysis output values

and camera system analysis values are compared in that graph, and the difference between them is very little, as shown in Figure 6.20.

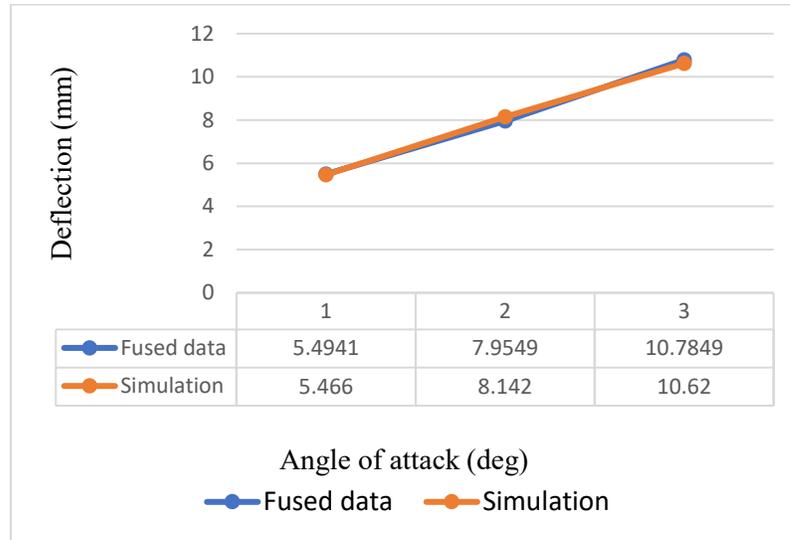


Figure 6.20: Angle of attack vs. Deflection.

The minimum velocity required to lift the fixed-wing under several conditions was calculated using both the surface and the projected area for an accurate result. The results are presented in Tables 6.4 and 6.5. The roll angle and minimum velocity to lift the fixed-wing UAV are represented in the graphs with AOA, 10 degrees, 20 degrees, and 30 degrees, as shown in Figures 6.21, 6.22, and 6.23. With increasing roll angles, the lift force decrease, and relative to the decrease in force, more velocity is required to lift the body. Therefore, more force and velocity are required during airplane take-off. In this experiment, at constant AOA of 10 degrees, 20 degrees, and 30 degrees and different roll angles ranging from 0 to 30 degrees at an interval of 10 were considered. Comparatively, the value of lift force increases with an increase of AOA, so the lift velocity at AOA of 10 degrees is higher compared to the lift velocity of 30 degrees under the same conditions.

Table 6.4 Minimum velocity required to lift the wing (Projected area).

AOA	Roll angle	Mass (kg)	Gravitational acceleration (m/s ²)	The density of air (kg/m ³)	Projected Area (m ²)	Lift force (N)	Co-efficient of lift	Minimum velocity for lift (m/sec)
10	0	1.25	9.81	1.225	0.026	1.611	1.0686	26.82238
10	10	1.25	9.81	1.225	0.026	1.298	0.8610	29.88187
10	20	1.25	9.81	1.225	0.026	1.168	0.7747	31.50095
10	30	1.25	9.81	1.225	0.026	1.003	0.6653	33.9934
20	0	1.25	9.81	1.225	0.0513	2.349	0.7909	22.21282
20	10	1.25	9.81	1.225	0.0513	1.977	0.6656	24.21262
20	20	1.25	9.81	1.225	0.0513	1.76	0.5926	25.6619
20	30	1.25	9.81	1.225	0.0513	1.444	0.4862	28.33098
30	0	1.25	9.81	1.225	0.075	2.825	0.6506	20.25517
30	10	1.25	9.81	1.225	0.075	2.283	0.5258	22.53161
30	20	1.25	9.81	1.225	0.075	2.015	0.4640	23.98323
30	30	1.25	9.81	1.225	0.075	1.569	0.3613	27.17901

Table 6.5 Minimum velocity required to lift the wing (Surface area).

AOA	Roll angle	Mass (kg)	Gravitational acceleration (m/s ²)	The density of air (kg/m ³)	Surface Area (m ²)	Lift force (N)	Co-efficient of lift	Minimum velocity for lift (m/sec)
10	0	1.25	9.81	1.225	0.153	1.611	0.1818	26.8223
10	10	1.25	9.81	1.225	0.153	1.298	0.1465	29.8818
10	20	1.25	9.81	1.225	0.153	1.168	0.1318	31.5009
10	30	1.25	9.81	1.225	0.153	1.003	0.1132	33.9934
20	0	1.25	9.81	1.225	0.153	2.349	0.2652	22.2128
20	10	1.25	9.81	1.225	0.153	1.977	0.2232	24.2126
20	20	1.25	9.81	1.225	0.153	1.76	0.1987	25.6619
20	30	1.25	9.81	1.225	0.153	1.444	0.1630	28.3309
30	0	1.25	9.81	1.225	0.153	2.825	0.3189	20.2551
30	10	1.25	9.81	1.225	0.153	2.283	0.2577	22.5316
30	20	1.25	9.81	1.225	0.153	2.015	0.2274	23.9832
30	30	1.25	9.81	1.225	0.153	1.569	0.1771	27.1790

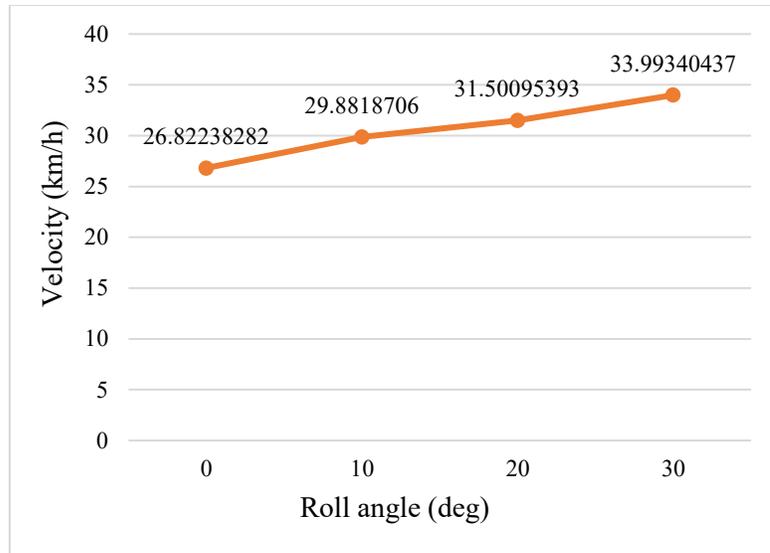


Figure 6.21: Roll angle vs. minimum lift velocity for AOA 10 deg.

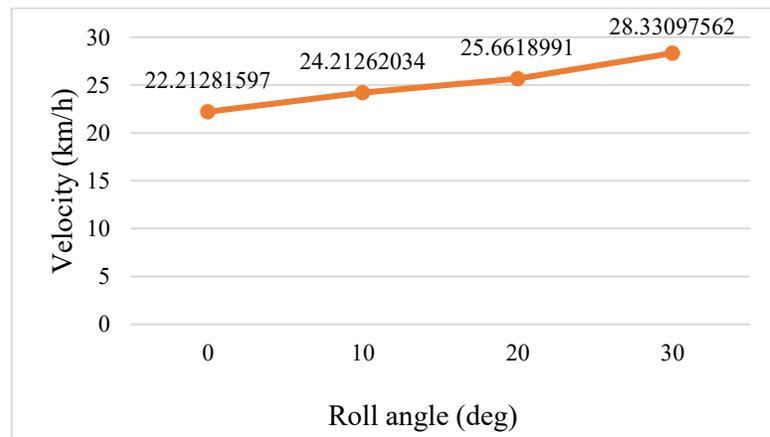


Figure 6.22: Roll angle vs. minimum lift velocity for AOA 20 deg.

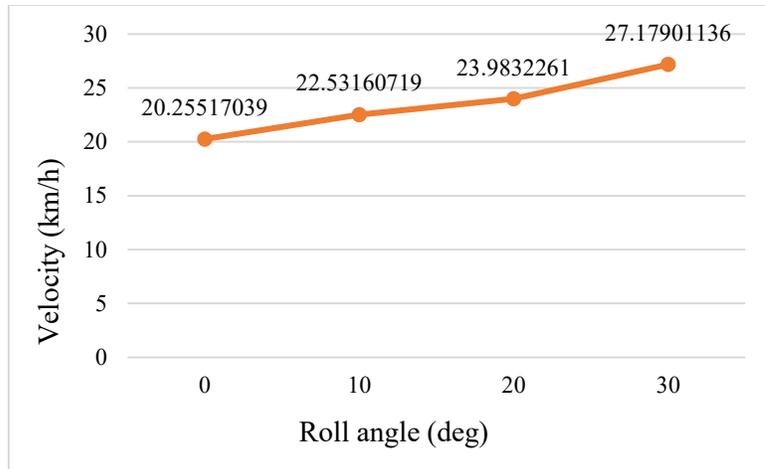


Figure 6.23: Roll angle vs. minimum lift velocity for AOA 30 deg.

The fixed-wing area exposed to the wind increases as the AOA increases from 10 degrees to 30 degrees. Due to this condition, the value of the force increase also increased. As a result, lesser velocity will be required for lift with higher AOA, as shown in Figure 6.24.

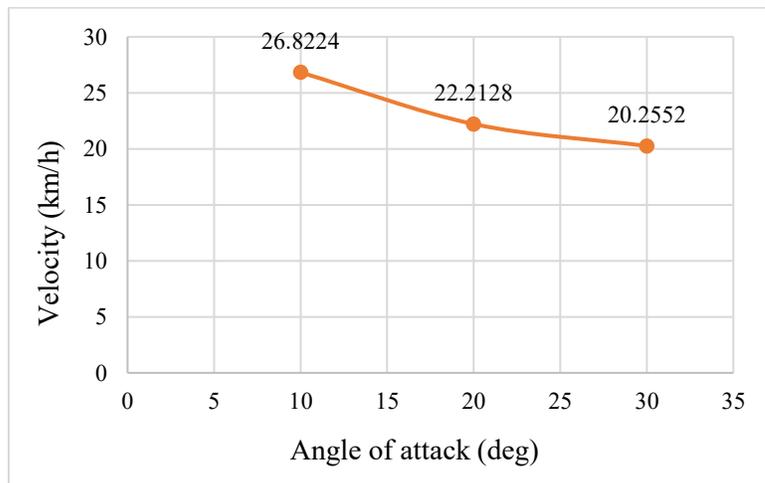


Figure 6.24: Angle of attack vs. minimum velocity required for lift.

Table 6.6. The final output result for the UAV control.

Wind velocity (km/h)	Angles		Camera/S.G		Simulation		Camera/S.G		Simulation	
	AOA	Roll angle	Drag force	Lift force	Drag force	Lift force	Co-efficient of drag	Co-efficient of lift	Co-efficient of drag	Co-efficient of lift
35	10	0	0.278	1.611	0.288	1.631	0.0432	0.2507	0.0464	0.2901
35	10	10	0.248	1.298	0.248	1.308	0.0385	0.2019	0.0389	0.2011
35	10	20	0.236	1.168	0.239	1.159	0.0367	0.1817	0.0369	0.1897
35	10	30	0.214	1.003	0.218	1.006	0.0333	0.1560	0.0335	0.1565
35	20	0	0.668	2.349	0.669	2.351	0.1039	0.3655	0.1031	0.3652
35	20	10	0.710	1.977	0.725	1.978	0.1104	0.3076	0.1117	0.3079
35	20	20	0.608	1.760	0.611	1.765	0.0946	0.2738	0.0948	0.2736
35	20	30	0.542	1.444	0.561	1.437	0.0843	0.2247	0.0853	0.2256
35	30	0	1.461	2.825	1.471	2.826	0.2273	0.4396	0.2288	0.4359
35	30	10	1.264	2.283	1.274	2.286	0.1967	0.3552	0.1945	0.3566
35	30	20	1.058	2.015	1.061	2.017	0.1646	0.3135	0.1648	0.3176
35	30	30	0.903	1.569	0.908	1.567	0.1405	0.2441	0.1401	0.2442

6.5 Future Work

Measure the dynamic behaviours and control of the fixed-wing UAV with a Mini ZED stereo camera attached to the tail of the UAV in real-time flight. The information will be transferred to the control station using the cloud system, which is a new added value to the new ZED stereo camera. The prepared model of the UAV is shown in Figures 6.25, 6.26, and 6.27.

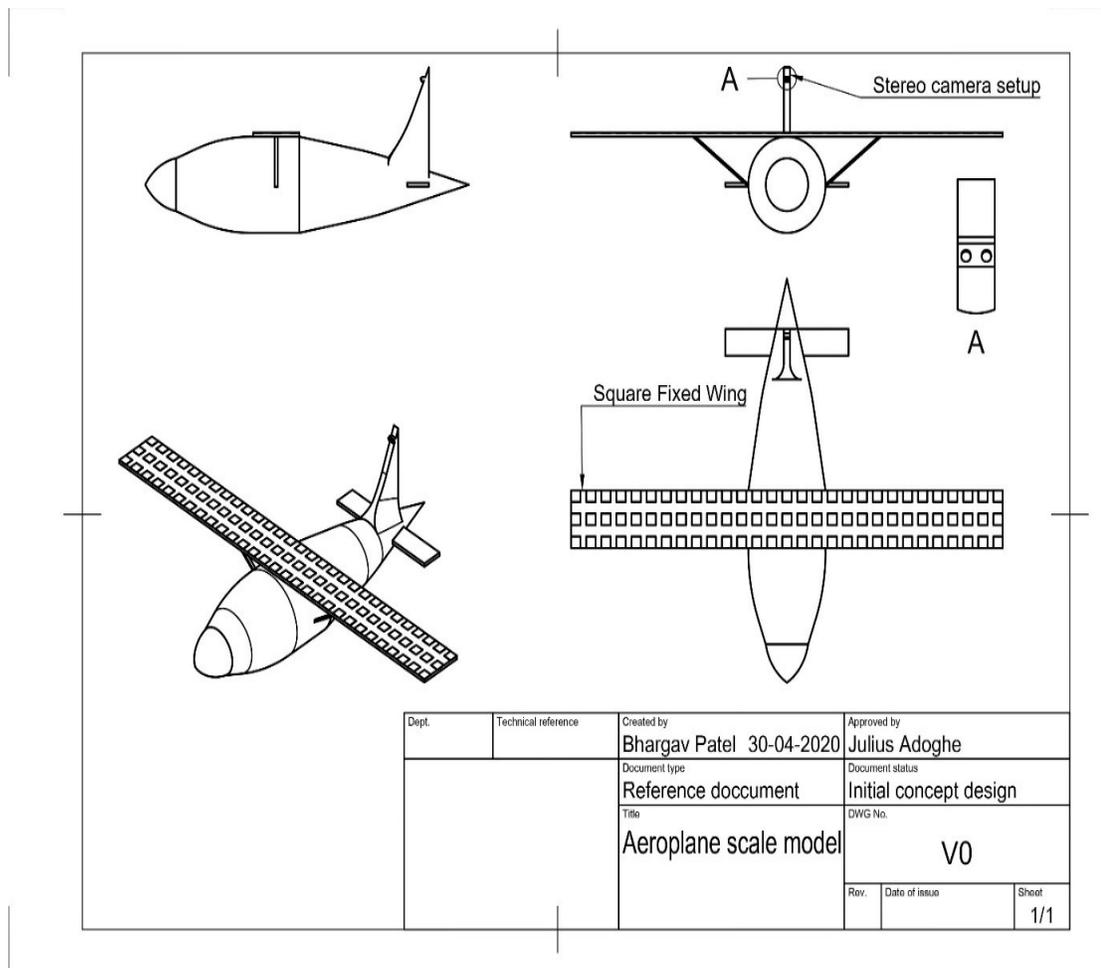


Figure 6.25: Technical drawing of the UAV.



Figure 6.26: Isometric view of the model UAV.

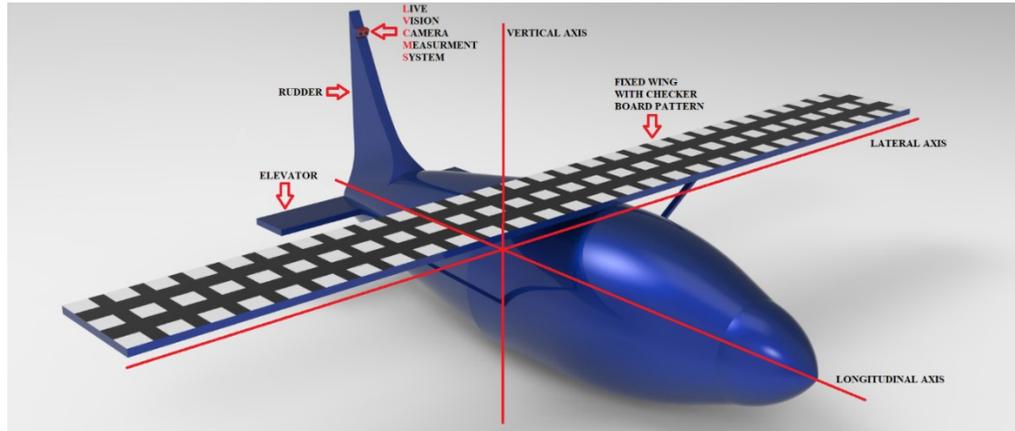


Figure 6.27: The prepared model of the UAV.

- Use the DDVS system in real-world tests, and keep in mind that this is the only method to validate the control system built in this thesis.
- Verify the simulation results using flight tests.
- Using sensor fusion, use multi-sensors to validate the results and estimate all UAV states.
- Increasing autonomous capabilities by adding trajectory planning.

Chapter 7

7 Conclusion

The Deflection-Detection-Vision-System (DDVS) is developed to calculate the deflection and identify aerodynamic coefficients of the fixed-wing during the flight. This system employed a stereo camera to estimate the deflection for a specific point on the wing and compared it with the strain gauge sensor reading. This visual detection system is based on three main algorithms: stereo camera calibration, visual points tracking, and deflection identification using a three-dimensional technique. The calibration for a stereo camera allows us to find intrinsic parameters. Eight selected points are used to identify the wing shape visually during the flight. Extensive experiments have been conducted in a wind tunnel for different flight conditions. The experiential tests confirmed computer simulation results and measurements obtained from the vision system and strain gauges placed on the wing. When the angle of attack increases, the deflection of the fixed-wing increases, but with the increase in rolling angle, the deflection decreases for the same angle of attack. The maximum wing deflections occurred when the speed was 30 km/h, the roll angle was 0 degrees, and the angle of attack (AOA) was greater than 30 degrees. The Unscented Kalman Filter sensor fusion method was used to improve attitude and other flight parameters, measurements accuracy, and efficiency, by integrating data from the Deflection-Detection Vision System (DDVS) with the strain gauge sensor data. The novel sensor fusion algorithm was designed and implemented also to estimate the attitude of a flexible wing UAV. A control algorithm was developed and experimentally verified for the model of the Cessna 172 aircraft model in the wind tunnel, and the control system is based

on Fuzzy-PID. The proposed Fuzzy-PID controller's performance and response are noted to be satisfactory. It shows that the overshoot, settling and rising times have all improved. In the next stage of system development, the full integration of measurements from the vision system and strain gauge sensors during flight and the design of an autopilot system that will control and navigate both longitudinal and lateral motion is planned.

References

- [1] W. Burner and L. Tianshu, "Videogrammetric model deformation measurement technique" at NASA Langley research center, Hampton, Virginia 23681-2199, *Journal of aircraft*, Vol. 38, No. 4, July–August 2001.
- [2] A. W. Burner, R. A. Wahls, and W. K. Goad, "Wing twist measurements at the national transonic facility," NASA NASA Langley research center, Hampton, Virginia, TM 110229, *Journal of aircraft*, Feb. 1996.
- [3] J. R. Hooker, A.W. Burner, and R. Valla, "Static aeroelastic analysis of transonic wind tunnel models using finite element methods," *15th Applied Aerodynamics Conference*, AIAA Paper 97-2243, June 1997.
- [4] S. Dongyue, et al. "Improved feature point extraction and mismatch eliminating algorithm" *Journal of Systems Science & Control Engineering*, volume 8, 2020
- [5] M. A. Al-Isawi, J. Z. Sasiadek, "Guidance and control of autonomous, flexible wing UAV with advanced vision system," *23rd IEEE International Conference on Methods & Models in Automation & Robotics (MMAR)*, DOI:10.1109/MMAR.2018.8486058 2018.
- [6] M. A. Al-Isawi, J. Z. Sasiadek, "Control of flexible wing UAV using stereo camera," *Aerospace Robotics III*, published by Springer publishers, 2019.
- [7] D. S. Varuna, R. Jamie, and K. Ahmet; "Robust Fusion of LiDAR and Wide-Angle Camera Data for Autonomous Mobile Robots." *Institute for Digital Technologies, Loughborough University*, London E15 2GZ, U.K., 15 August 2018.

- [8] J. O. Adoghe, J. Z. Sasiadek “Unmanned Aerial Vehicle Fixed Wing Shape, and Deflection Measurement Using Vision System,” *25th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2020.
- [9] Trilion Quality Systems; “Can Digital Image Correlation Replace Strain Gauges?” Trilion Headquarters, 651 Park Avenue King of Prussia, PA 19406, Online. Available: <https://trilion.com/replace-strain-gauges>, [Accessed 19th June 2019].
- [10] A. H. Mary, T. Kara, and A. H. Miry, “Inverse kinematics solution for robotic manipulators based on fuzzy logic and PD control.” *In 2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)* (pp. 1-6). IEEE. 2016.
- [11] D. Zhou, and Q. Geng; “Multi-model and fuzzy PID control for fixed-wing UAV.” *3rd International Conference on Mechatronics, Robotics and Automation (ICMRA 2015)*, pp 523-528, 2015.
- [12] D. F. Dementhon and L. S. Davis, “Model-based object pose in 25 lines of code,” *International Journal of Computer Vision*, vol. 15, pp. 123-141, 1995.
- [13] L. Quan and Z. Lan, “Linear n-point camera pose determination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 774-780, 1999.
- [14] E. Michaelsen, M. Kirchhof, U. Stilla, “Sensor pose inference from airborne videos by decomposing homography estimates,” *International Archives of Photogrammetry and Remote Sensing*, vol. 35, no. 3, p. 6, 2004.
- [15] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, “An invitation to 3D vision: from images to geometric models,” Springer Science and Business Media, vol. 26, 2003.

- [16] E. Montijano and C. Sagues, “Fast pose estimation for visual navigation using homographies,” *IEEE/RSJ International Conference in Intelligent Robots and Systems*, 2009.
- [17] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239-256, 1992.
- [18] M. J. Walker and J. Z. Sasiadek, “Accurate pose determination for autonomous vehicle navigation,” *IEEE 18th International Conference in Methods and Models in Automation and Robotics MMAR*, 2013.
- [19] B. K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *Journal of the Optical Society of America*, vol. 4, no. 4, pp. 629-642, 1987.
- [20] J. Z. Sasiadek and M. J. Walker, “Accurate image depth determination for autonomous vehicle navigation,” in *CARO3 3rd Conference on Aerospace Robotics*, 2015.
- [21] M. A. Al-Isawi, and J. Z. Sasiadek, “Guidance and Control of Autonomous, Flexible Wing UAV with Advanced Vision System,” *23rd IEEE International Conference on Methods & Models in Automation & Robotics (MMAR)* (pp. 441-448). August 2018
- [22] G. Cai, Y. Lum, M. Chen, and H. Lee, “A brief overview on miniature fixed-wing unmanned aerial vehicles,” *Proc. 8th IEEE International Conference on Control and Automation*, Xiamen, China, pp. 285–290, 2010.
- [23] B. L. Stevens, and F.L Lewis, “*Aircraft control and simulation*. John Wiley & Sons Inc.,” 2nd edition. 2003.

- [24] T. R. Teo, J. S. Jang, and C. J. Tomlin, "Automated multiple UAV flight" – the Stanford dragon Fly UAV program *43rd IEEE Conference on Decision and Control*. 2004
- [25] T. Kara, and A.H. Mary, "Robust trajectory tracking control of robotic manipulators based on model-free PID-SMC approach," *Journal of Engineering Research*, pp.170-188. 2018
- [26] T. Kara, and A.H. Mary, "Adaptive PD-SMC for Nonlinear Robotic Manipulator Tracking Control," *Studies in Informatics and Control*, 26(1), pp.49-58. 2017.
- [27] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and M. Smailzadeh, "Attitude control of a quadrotor with optimized PID controller," *Intelligent Control and Automation*, pp. 335-342, 2013.
- [28] A. Sarhan, and M. Ashry, "Self-tuned PID controller for the Aerosonde UAV autopilot," *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2 Issue 12, pp. 3329-3340, 2013.
- [29] M. A. Al-Isawi, J. Z. Sasiadek, "Guidance and Control of a Robot Capturing an Uncooperative Space Target," *Journal of Intelligent & Robotic Systems*, 93(3-4), pp.713-721, 2019.
- [30] M. A. Al-Isawi, J. Z. Sasiadek, "Control of flexible wing UAV using stereo camera," *In Aerospace Robotics III* (pp. 97-120). Springer, Cham. 2019.
- [31] T.C.T. Ng, F.H.F. Leung and P.K.S. Tam, "A simple gain scheduled PID controller with stability consideration based on a grid-point concept," *Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 1090-1094, 1997.

- [32] A. Brezoescu, T. Espinoza, P. Castillo, and R. Lozano, "Adaptive trajectory following for a fixed-wing UAV in the presence of crosswind," *Journal of Intelligent and Robotic Systems*, Vol 69, Issue1-4, pp. 257-271, 2013.
- [33] N. Wahid, and N. Hassan, "Self-tuning fuzzy PID controller design for aircraft pitch control," *IEEE Third International Conference on Intelligent Systems Modelling and Simulation*, pp. 19-24, 2012
- [34] H.B. Kazemian, "Developments of fuzzy PID controllers," *Expert Systems*, vol.22, no.5, pp. 254-264, 2005.
- [35] A.H. Mary, T. Kara and A.H. Miry, "Inverse kinematics solution for robotic manipulators based on fuzzy logic and PD control," *In 2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)* (pp. 1-6). IEEE, May 2016.
- [36] D. Zhou, and Q. Geng, "Multi-model and fuzzy PID control for fixed-wing UAV," *3rd International Conference on Mechatronics, Robotics and Automation (ICMRA 2015)*, pp 523-528, 2015.
- [37] L.M. Argentim, W.C. Rezende, P.E. Santos and R.A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," *IEEE International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1 – 6, 2013.
- [38] X.J. Xing, J.G. Yan and D.L. Yuan, "Augmented-stability controller design and its simulation or a UAV based on LQR theory," *Flight Dynamics*, vol. 29, No. 5, pp. 54-56, 2011.
- [39] Y. LI, C.Chen, and W. Chen, "Research on longitudinal control algorithm for flying wing UAV based on LQR technology," *International Journal on Smart Sensing and Intelligent Systems* Vol. 6, No. 5, pp. 2155-2181, 2013.

- [40] M. R. Rahimi, S. Hajjighasemi., D.Sanaei; “Designing and Simulation for Vertical Moving Control of UAV System using PID, LQR and Fuzzy logic,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 3, no. 5, pp. 651-659, 2013.
- [41] K. K. Yit, P. Rajendran and L.K. Wee, “Proportional-derivative linear quadratic regulator controller design for improved longitudinal motion control of unmanned aerial vehicles,” *International Journal of Micro Air Vehicles*, 2016.
- [42] S. M. Raafat, Z. Mahmoud, “Robust Multiple Model Adaptive Control for Dynamic Positioning of Quadrotor Helicopter System,” *Engineering and Technology Journal*, vol. 36, no.12, Part (A), p.p. 1249-1259, 2018
- [43] M. V. Cook; “Flight Dynamics Principles.” *Arnold, London*, 3rd edition. 2013
- [44] T.A Weisshaar, and H. Ashley, “Static Aeroelasticity and the Flying Wing,” *Journal of Aircraft*, Vol. 10, No. 10, 1973, pp. 586–594.
- [45] M. Wintzer, P. Sturdza, and I. Kroo; “Conceptual Design of Conventional and Oblique Wing Configurations for Small Supersonic Aircraft,” *44th AIAA Aerospace Sciences Meeting and Exhibit*, Paper 2006-930, 2006.
- [46] J. R. Banerjee; “Flutter Characteristics of High Aspect Ratio Tailless Aircraft,” *Journal of Aircraft*, Vol. 21, No. 9, 1984, pp. 733–736.
- [47] M. Love, et al.; “Body Freedom Flutter of High Aspect Ratio Flying Wings,” *47th AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 2005-1947, pp 345-367, 2005.

- [48] M. J. Patil, D. H. Hodges, and C. E. S. Cesnik; “Nonlinear Aeroelastic Analysis of Complete Aircraft in Subsonic Flow,” *Journal of Aircraft*, Vol. 37, No. 5, Sept.–Oct. 2000, pp. 753–760.
- [49] M. J. Patil, and D. H. Hodges; “On the Importance of Aerodynamic and Structural Geometrical Nonlinearities in Aeroelastic Behavior of High-Aspect-Ratio Wings,” *Journal of Fluids and Structures*, Vol. 19, No. 7, Aug. 2004, pp. 905–915.
- [50] M. Keshmiri and W. F. Xie, “Catching moving objects using a navigation guidance technique in a robotic visual servoing system,” *American Control Conference (ACC)*, pp. 6302-6307, 2013.
- [51] G. Rouleau, S. Verma, I. Sharf, and É. Martin, “Vision-based tracking and trajectory generation for robotic capture of objects in space,” Collection of technical papers-AIAA *Guidance, Navigation and Control Conference*, pp. 5937-5950, 2005.
- [52] Y. Shi, B. Liang, X. Wang, W. Xu, and H. Liu, “Study on intelligent visual serving of space robot for cooperative target capturing,” *International Conference on Information and Automation (ICIA)*, 2012, pp. 733-738, 2012.
- [53] W. Xu, B. Liang, C. Li, Y. Liu, and Y. Xu, “Autonomous target capturing of free-floating space robot: Theory and Experiments,” *Robotica*, vol. 27, pp. 425-445, 2009.
- [54] X. Du, B. Liang, W. Xu, and Y. Qiu, “Pose measurement of large non-cooperative satellite based on collaborative cameras,” *Acta Astronautica*, vol. 68, pp. 2047-2065, 2011.

- [55] R. Zhan and J. Wan, "Neural network-aided adaptive unscented Kalman filter for nonlinear state estimation," *IEEE Signal Processing Letters*, vol. 13, pp. 445-448, 2006.
- [56] Z. Jiang, Q. Song, Y. He, and J. Han, "A novel adaptive unscented Kalman filter for nonlinear estimation," *46th IEEE Conference on Decision and Control*, pp. 4293-4298, 2007.
- [57] C. Hajiyev and H. E. Soken, "Robust adaptive unscented Kalman filter for attitude estimation of picosatellites," *International Journal of Adaptive Control and Signal Processing*, vol. 28, pp. 107-120, 2014.
- [58] S. Pourtakdoust and H. Ghanbarpour, "An adaptive unscented Kalman filter for quaternion-based orientation estimation in low-cost AHRS," *Aircraft Engineering and Aerospace Technology*, pp. 485-493, 2007.
- [59] H. Ashley, "Static aeroelasticity and the flying wing," *Journal of Aircraft*, vol. 10, no. 10, pp. 586-594, 1973.
- [60] M. J. Patil and D. H. Hodges, "Flight dynamics of highly flexible flying wings," *Journal of Aircraft*, vol. 43, pp. 1790-1798, 2006.
- [61] R. Albertani, B. Stanford, J. P. Hubner, R. Lind, and P. Ifju, "Experimental analysis of deformation for flexible-wing micro air vehicles," *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, 2005.
- [62] N. A. Pitcher, J. T. Black, M. F. Reeder, and R. C. Maple, "Videogrammetry dynamics measurements of a lightweight, flexible wing in a wind tunnel," *Journal of Aircraft*, vol. 47, no. 1, pp. 172-180, 2010.

- [63] A. Burner and T. Liu, "Videogrammetric model deformation measurement technique," *Journal of Aircraft*, vol. 38, pp. 745-754, 2001.
- [64] A. W. Burner, R. A. Wahls, and W. K. Goad, "Wing twist measurements at the national transonic facility," *NASA Technical Memorandum* 110229, 1996.
- [65] J. Bakunowicz and R. Meyer, "In-flight wing deformation measurements on a glider," *The Aeronautical Journal*, vol. 120, pp. 1917-1931, 2016.
- [66] A. Burner, T. Liu, and R. DeLoach, "Uncertainty of videogrammetric techniques used for aerodynamic testing," *22nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, 2002.
- [67] S. Chiu, S. Chand, D. Moore, and A. Chaudhary, "Fuzzy logic for control of roll and moment for a flexible wing aircraft," *IEEE Control Systems*, vol. 11, pp. 42-48, 1991.
- [68] P. Wahi, R. Raina, and F. N. Chowdhury, "A survey of recent work in adaptive flight control," *Proceedings of the 33rd Southeastern Symposium on System Theory*, 2001.
- [69] P. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles," *IEEE Control Systems*, vol. 34, pp. 42-59, 2014.
- [70] H. Bolandi, M. Rezaei, R. Mohsenipour, H. Nemati, and S. M. Smailzadeh, "Attitude control of a quadrotor with optimized PID controller," *Intelligent Control and Automation*, vol. 4, 2013.

- [71] A. Sarhan and M. Ashry, "Self-tuned PID controller for the Aerosonde UAV autopilot," *International Journal of Engineering Research & Technology*, vol. 2, pp. 3329-3340, 2013.
- [72] P. Shahmaleki and M. Mahzoon, "GA modeling and ANFIS control design for a solar power plant," *American Control Conference (ACC)*, pp. 3530-3535, 2010.
- [73] L. S. Brian and L. L. Frank, "Aircraft control and simulation," New Jersey: John Wiley & Sons, Inc., 2003.
- [74] R. Teo, J. S. Jang, and C. J. Tomlin, "Automated multiple UAV flight-the Stanford Dragonfly UAV program," *43rd IEEE Conference on Decision and Control, (CDC)*. 2004.
- [75] T. Ng, F. Leung, and P. Tam, "A simple gain scheduled PID controller with stability consideration based on a grid-point concept," *ISIE'97, Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 1090-1094, 1997.
- [76] A. Brezoescu, T. Espinoza, P. Castillo, and R. Lozano, "Adaptive trajectory following for a fixed-wing UAV in presence of crosswind," *Journal of Intelligent & Robotic Systems*, pp. 1-15, 2013.
- [77] N. Wahid and N. Hassan, "Self-tuning fuzzy PID controller design for aircraft pitch control," *Third International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, 2012, pp. 19-24, 2012.
- [78] H. B. Kazemian, "Developments of fuzzy PID controllers," *Expert Systems*, pp. 254-264, 2005.

- [79] D. Zhou and Q. Geng, "Multi-model And Fuzzy PID Control for Fixed-wing UAV," in the *3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*, 2015.
- [80] M. M. Al-Isawi, "Navigation and Control of Flexible Wing UAV Using Vision System," Doctor of Philosophy in Mechanical Engineering, Ottawa-Carleton Institute for Mechanical and Aerospace Engineering (OCIMAE), *Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, Ontario*, June, 2018.
- [81] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp 123-133, 1999.
- [82] J. Shi, "Good features to track," *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, 1994.
- [83] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.
- [84] R. Lakemond, S. Sridharan, and C. Fookes, "Hessian-based affine adaptation of salient local image features," *Journal of Mathematical Imaging and Vision*, vol. 44, pp. 150-167, 2012.
- [85] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381-395, 1981.

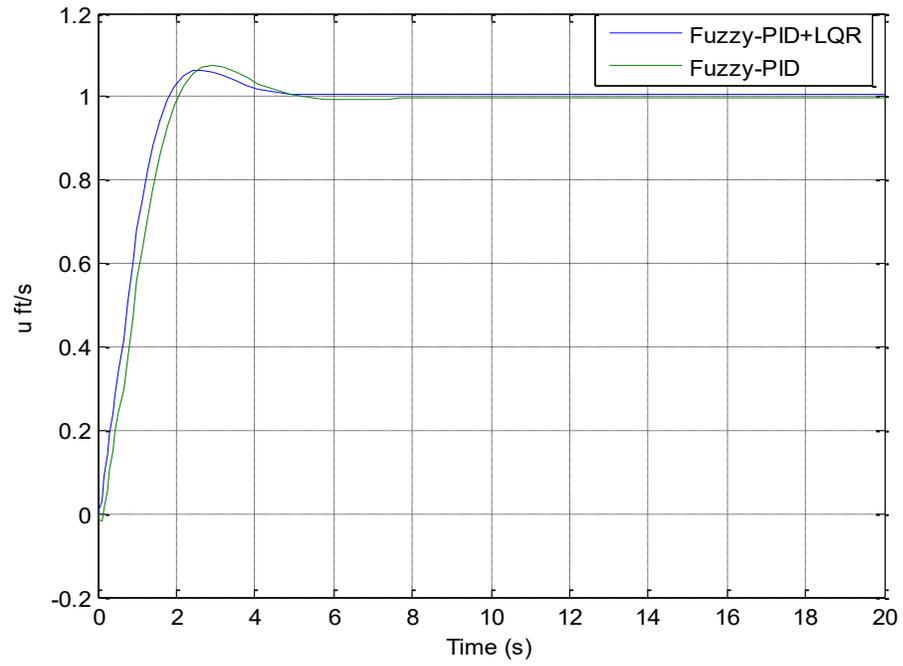
- [86] J. Jae Lee and G. Kim, "Robust estimation of camera homography using fuzzy RANSAC," *International Conference on Computational Science and Its Applications*, 2007.
- [87] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1472-1482, 2008.
- [88] O. D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, pp. 485-508, 1988.
- [89] J. Bouguet, "Camera calibration toolbox for Matlab," [Online]. Available: [www.vision.caltech.edu/bouguetj/calib - doc](http://www.vision.caltech.edu/bouguetj/calib_doc). [Accessed: 14-Feb-2019]
- [90] M. Chiaramonti, F. Giulietti, and G. Mengali, "Formation control laws for a formation of autonomous flight vehicles," *14th Mediterranean Conference Control and Automation*, 2006.
- [91] G. Hu, W. MacKunis, N. Gans, W. E. Dixon, J. Chen, A. Behal, et al., "Homography-based visual servo control with imperfect camera calibration," *IEEE Transactions on Automatic Control*, vol. 54, pp. 1318-1324, 2009.
- [92] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239-256, 1992. B.
- [93] K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A (JOSA A)*, vol. 4, pp. 629-642, 1987.

- [94] H. F. Viki, "NI Strain gauge tutorial, national instruments corporation," 1998, [Online]. Available: <https://www.scribd.com/document/12921011/ni-strain-gauge-tutorial>, [Accessed: 01-Sep-2019]
- [95] B. D. Patel, A. R. Srinivas, "Validation of experimental strain measurement technique and development of force transducer," *International Journal of Scientific & Engineering Research*, Volume 3, Issue 10, October-2012.
- [96] A. E. Ahmed, A. Hafez, A. Ouda, H. E. H. Ahmed, and H. M. Abd-Elkader, "Modeling of a small unmanned aerial vehicle," *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering* vol. 9, no. 3, 2015.
- [97] Y. Shengyi, L. Kunqin, and S. Jiao, "Design and simulation of the longitudinal autopilot of UAV based on self-adaptive fuzzy PID control," *International Conference on Computational Intelligence and Security*, pp 335, 2009.
- [98] J. O. Adoghe, M. M. A. Al-Isawi, A. J. Attiya, "UAV Control Based on Dual LQR and Fuzzy-PID Controller," *Al-Khwarizmi Engineering Journal on Computer Science*, volume 16, pp 43-53, September 2020.
- [99] L. Charles, L. Phillips, R.D. Harbor, Feedback control systems, Prentice Hall, 1991.
- [100] J. Ziegler, N. Nichols, "Optimum settings for automatic controllers," *Transactions of the American Society of Mechanical Engineers (ASME)*, vol.64, pp.759-768, 1942.
- [101] K. Ogata, "Modern control engineering," 3rd Ed., Prentice Hall, 1999.
- [102] H. Ozbay, "Introduction to feedback control theory," CRC Press, 2000.

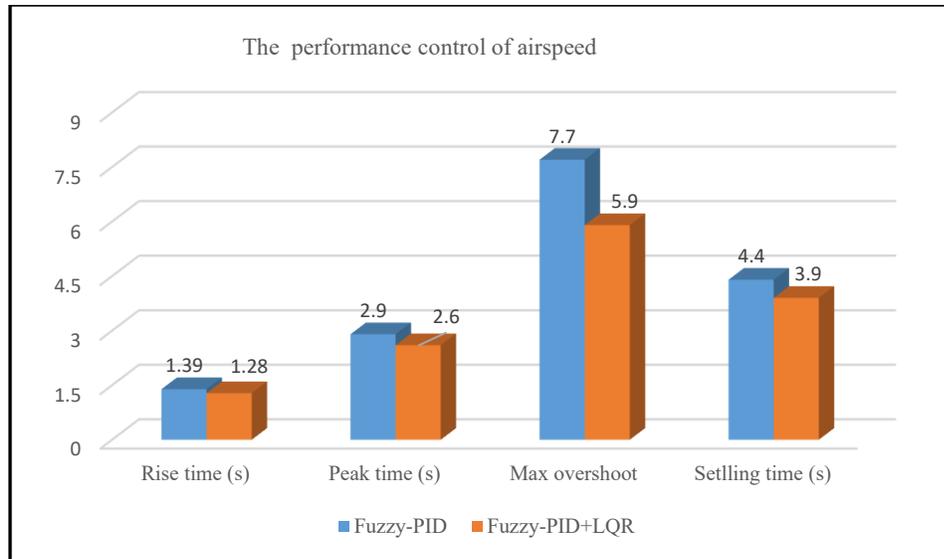
- [103] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, No. 4, pp. 555 -576, 2005.
- [104] W.F. Liu, Z. Jiang, and Z. Gong, "Online fuzzy self-adaptive PID attitude control of a sub-mini fixed-wing air vehicle," *IEEE International Conference on Mechatronics and Automation*, pp. 153-157, 2007.
- [105] I. Aniekan, O. Ikechukwu, and S. Paul, "Comparative analysis of a PID controller using Ziegler Nichols and auto turning method," *International Academic Journal of Science and Engineering*, Vol. 3, No. 10, pp. 1-16, 2016.
- [106] S. J. Simon, J. K. Jeffrey, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE Periodical, Internet Resource*, vol. 92, no. 3, pp. 401-422, 2004.
- [107] E.A. Wan, and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153-158, 2000.
- [108] J. Kim, H. Y. Kim, and H. Sohn, "Autonomous dynamic displacement estimation from data fusion of acceleration and intermittent displacement measurements." *Mech Syst Signal Pr 2014*; 42: 194–205.
- [109] L.A. Zadeh (1072) "A Rationale for Fuzzy Control Dynamic." *System Measurement Control*, vol.94, series G, pp.3 - 4.

Appendix

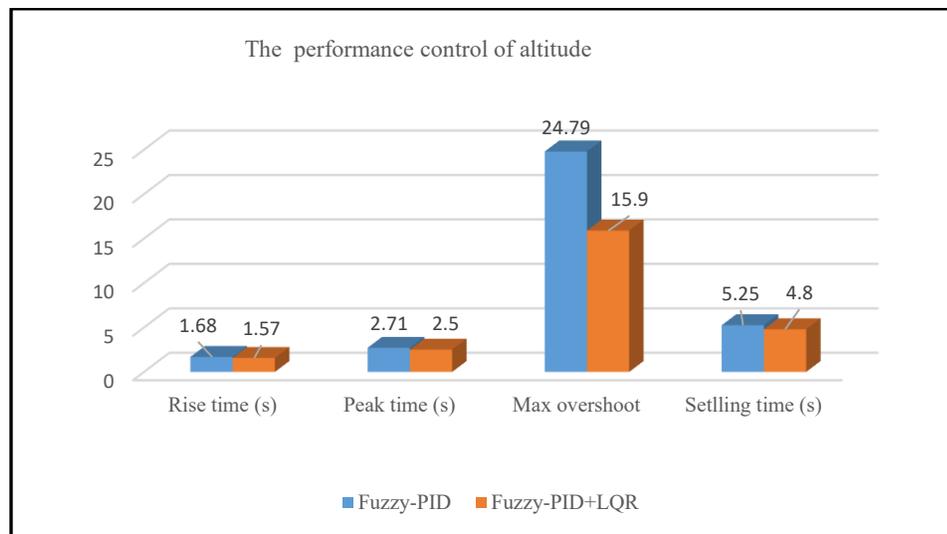
Appendix A



Appendix A1: Airspeed due to both controllers



Appendix A2: The airspeed performance control for both cases.



Appendix A3: The altitude performance control for both cases.

Appendix B

Equivalent Aerodynamic Stability Derivative

$$X_U = \rho u_0 S C_w \sin\theta + \frac{1}{2} \rho u_0 S C_{xu}$$

$$Z_U = -\rho u_0 S C_w \sin\theta + \frac{1}{2} \rho u_0 S C_{zu}$$

$$M_U = \frac{1}{2} \rho u_0 \bar{c} S C_{mu}$$

$$X_w = \frac{1}{2} \rho u_0 S C_{x\alpha}$$

$$Z_w = \frac{1}{2} \rho u_0 S C_{z\alpha}$$

$$M_w = \frac{1}{2} \rho u_0 \bar{c} S C_{m\alpha}$$

$$X_q = \frac{1}{4} \rho u_0 \bar{c} S C_{xq}$$

$$Z_q = \frac{1}{4} \rho u_0 \bar{c} S C_{zq}$$

$$M_q = \frac{1}{4} \rho u_0 \bar{c}^2 S C_{mq}$$

Appendix C

Live Loop Stereo Camera Deflection Calculation Code Using Matlab

In Matlab, the camera input has been defined by the function

```
cam = webcam;
```

```
preview(cam);
```

Firstly, capture the initial image and save that image in system

```
img = cam.snapshot;
```

```
    imshow(img); % display the image
```

```
    imwrite(img,sprintf('image%d.jpg',j));
```

Secondly, by putting the camera image capturing and saving functions in FOR loop where

it stands for the limit of 1: N where the N is for N^{th}

```
for i=1:N
```

```
    img = cam.snapshot;
```

```
    imshow(img); % display the image
```

```
    imwrite(img,sprintf('image%d.jpg',i));
```

```
    pause(2); % pause for one second
```

Also, putting the point matching algorithm in the same loop for matching points and co-

ordinate defined by using first and N^{th} image, by subtraction of co-ordinates of both

images that gives deflection value

```
I1=rgb2gray(imread('image0.jpg'));
```

```
I2=rgb2gray(imread(sprintf('image%d.jpg',i)));
```

```
points1 = detectMinEigenFeatures(I1);
```

```

points2 = detectMinEigenFeatures(I2);

[features1,valid_points1] = extractFeatures(I1,points1);

[features2,valid_points2] = extractFeatures(I2,points2);

indexPairs = matchFeatures(features1,features2);

matchedPoints1 = valid_points1(indexPairs(:,1),:);

matchedPoints2 = valid_points2(indexPairs(:,2),:); %figure;

showMatchedFeatures(I1,I2,matchedPoints1,matchedPoints2,'montage');

[tform, inlierBoxPoints, inlierScenePoints] = ...
    estimateGeometricTransform(matchedPoints1(), matchedPoints2(), 'affine');

figure;

showMatchedFeaturesI1, I2, inlierBoxPoints, ...
    inlierScenePoints, 'montage');

XL=matchedPoints1.Location(:,1);

XR=matchedPoints2.Location(:,1);

b=12;

f=1000;

Z=f*b/(XL-XR);

end

```

This code will run continuously for the set limit

Unscented Kalman Filter Matlab code for fusing Camera and strain gauge

- function [y,Y,P,Y1]=ut(f,X,Wm,Wc,n,R)
- %Unscented Transformation
- L=size(X,2);
- y=zeros(n,1);
- Y=zeros(n,L);
- for k=1:L
- Y(:,k)=f(X(:,k));
- y=y+Wm(k)*Y(:,k);
- end
- Y1=Y-y(:,ones(1,L));
- P=Y1*diag(Wc)*Y1'+R;

→ Sigma points around and is usually set to a small positive value. The sigma points are then propagated through the nonlinear functions, from which a new mean and covariance estimate is then formed.

- function X=sigmas(x,P,c)
- %Sigma points around reference point
- A = c*chol(P);%Cholesky??
- Y = x(:,ones(1,numel(x)));
- X = [x Y+A Y-A];

→ Unscented Kalman filter algorithm Matlab codes

- function [x,P]=ukf(fstate,x,P,hmeas,z,Q,R)

% UKF Unscented Kalman Filter for nonlinear dynamic systems

- $L = \text{numel}(x);$ %number of states
- $m = \text{numel}(z);$ %number of measurements
- $\alpha = 1e-2;$ %default, tunable
- $k_i = 0;$ %default, tunable
- $\beta = 2;$ %default, tunable
- $\lambda = \alpha^2 * (L + k_i) - L;$ %scaling factor
- $c = L + \lambda;$ %scaling factor
- $W_m = [\lambda/c \ 0.5/c + \text{zeros}(1, 2 * L)];$ %weights for means
- $W_c = W_m;$
- $W_c(1) = W_c(1) + (1 - \alpha^2 + \beta);$ %weights for covariance
- $c = \text{sqrt}(c);$
- $X = \text{sigmas}(x, P, c);$ %sigma points around x
- $[x_1, X_1, P_1, X_2] = \text{ut}(\text{fstate}, X, W_m, W_c, L, Q);$ %unscented transformation of process
- $[z_1, Z_1, P_2, Z_2] = \text{ut}(\text{hmeas}, X_1, W_m, W_c, m, R)$ %unscented transformation of measurements
- $P_{12} = X_2 * \text{diag}(W_c) * Z_2';$ %transformed cross-covariance
- $K = P_{12} * \text{inv}(P_2);$
- $x = x_1 + K * (z - z_1);$ %state update
- $P = P_1 - K * P_{12}';$ %covariance update

→ Matlab codes for fuse two sensors data

- `clear;clc;`
- `close all;`
- `n=2;t=.5;MC=1;`
- `Q=[.01 0 ; 0 0.02];`
- `R=1;`
- `ddt=1;`
- `f=@(x)[x(1)+x(2)*ddt;x(2)];`
- `h=@(x)[x(1);0];`
- `% measurement equation`
- `s=[.1 .1]'`;
- `x0=s+sqrtm(Q)*randn(n,1); % initial state with noise`
- `P0=[.1 0;0 .02];`
- `N=150; %50; % total dynamic steps`
- `ukV = zeros(n,N); %ukf estimate`
- `sV = zeros(n,N); %actual`
- `zV = zeros(2,N); % Of measured value`
- `ekx=zeros(MC,N);eky=zeros(MC,N);`
- `eux=zeros(MC,N);euy=zeros(MC,N);`
- `prompt = 'ENTER 4X1 MATRIX FOR CAMERA DEFLECTION
VALUES';`

- `sCamera = input(prompt);`
- `yc=linspace(min(sCamera),max(sCamera),N);`
- `xc=interp1(sCamera,1:4,yc,'spline','extrap')`
- `prompt = 'ENTER 4X1 MATRIX FOR STRAIN GAUGE DEFLECTION
VALUES';`
- `sstrain = input(prompt);`
- `ys=linspace(min(ssrain),max(ssrain),N);`
- `xS=interp1(ssrain,1:4,ys,'spline','extrap');`
- `sV= ys-yc;`
- `uP=P0;eP=P0;`
- `ux=x0;`
- `for k=1:N`
- `z = h(sV(k)) + sqrtm(R)*randn(2,1)/50; % measurments value`
- `zV(:,k) = z; % save measurment`
- `[ux, uP] = ukf(f,ux,uP,h,z,Q,R); % ukf`
- `ukV(:,k) = ux;`
- `end`
- `Ecamera=yc+ukV(1,:);`
- `plot(1:N,Ecamera,1:N,ys)`
- `legend('Estimate value by UKF','real values by CAMERA')`

- `xlabel('time (sec)')`
- `ylabel('Deflection (mm)')`